



FUNDAÇÃO EDSON QUEIROZ
UNIVERSIDADE DE FORTALEZA - UNIFOR

KARINA MARINHO DE SOUZA

**PROCESSAMENTO DE CONSULTAS EM REDES DE
SENSORES SEM FIO: UMA ABORDAGEM DE DETECÇÃO
DE NOVIDADES PARA O CONTROLE DA QUALIDADE DOS
SERVIÇOS DAS CONSULTAS**

FORTALEZA
CEARÁ - BRASIL
2009

Livros Grátis

<http://www.livrosgratis.com.br>

Milhares de livros grátis para download.

KARINA MARINHO DE SOUZA

**PROCESSAMENTO DE CONSULTAS EM REDES DE
SENSORES SEM FIO: UMA ABORDAGEM DE DETECÇÃO
DE NOVIDADES PARA O CONTROLE DA QUALIDADE DOS
SERVIÇOS DAS CONSULTAS**

Dissertação apresentada ao Curso de Mestrado em Informática Aplicada da Universidade de Fortaleza como parte das exigências para a obtenção do Título de Mestre em Ciências da Computação.

Orientador: Prof. Dr. Angelo Roncalli Alencar Brayner.

Co-orientador: Prof. Dr. André Luís Vasconcelos Coelho.

FORTALEZA
CEARÁ - BRASIL
2009

S729p Souza, Karina Marinho de.

Processamento de consultas em redes de sensores sem fio : uma abordagem de detecção de novidades para o controle da qualidade dos serviços das consultas / Karina Marinho de Souza. - 2009.

104 f.

Dissertação (mestrado) – Universidade de Fortaleza, 2009.

“Orientação: Prof. Dr. Angelo Roncalli Alencar Brayner.”

“Co-orientação: Prof. Dr. André Luís Vasconcelos Coelho.”

1. Banco de dados. 2. Sistema de comunicação sem fio. 3. Redes de sensores sem fio. I. Título.

CDU 681.3.06:061.68

KARINA MARINHO DE SOUZA

PROCESSAMENTO DE CONSULTAS EM REDES DE SENSORES SEM FIO: UMA ABORDAGEM DE DETECÇÃO DE NOVIDADES PARA O CONTROLE DA QUALIDADE DOS SERVIÇOS DAS CONSULTAS

Dissertação apresentada ao Curso de Mestrado em Informática Aplicada da Universidade de Fortaleza como parte das exigências para a obtenção do Título de Mestre em Ciências da Computação.

APROVADA em ____ de _____ de _____.

BANCA EXAMINADORA:

Prof. Angelo Roncalli Alencar Brayner, Dr.-ing.
(Orientador - Universidade de Fortaleza - UNIFOR)

Prof. André Luís Vasconcelos Coelho, Dr.
(Co-orientador - Universidade de Fortaleza - UNIFOR)

Prof. Eduardo Freire Nakamura, Dr.
(Fundação Centro de Análise, Pesquisa e Inovação Tecnológica - FUCAPI)

Prof. Raimir Holanda Filho, Dr.
(Universidade de Fortaleza - UNIFOR)

*Ao meu esposo e aos meus pais,
com muito amor.*

AGRADECIMENTOS

Após um longo caminho percorrido, cheio de desafios e incertezas, chega o momento de registrar meus sinceros agradecimentos a pessoas muito queridas e especiais que, de alguma maneira, contribuíram para a realização deste trabalho.

Agradeço primeiramente ao meu esposo, José Luiz, pelo seu enorme incentivo principalmente durante os últimos anos decorridos. Sua paciência, amor, carinho, cuidado, companheirismo e compreensão em todos os momentos que passamos juntos são uma bênção em minha vida.

Aos professores Angelo Brayner e André Coelho pelas suas orientações, motivações e valiosos ensinamentos que contribuíram muito para o meu aprendizado e sucesso na concretização deste trabalho.

Ao professor Plácido, por todo apoio fornecido durante o nosso convívio.

Aos meus pais que, além das infindáveis palavras de ternura, sempre estimularam as minhas carreiras profissional e acadêmica.

Ao meu irmão Arnaldo e ao amigo Ricardo Vasconcelos pelas importantíssimas contribuições efetuadas durante a implementação da proposta contida nesta dissertação.

Ao amigo Diorgens Miguel Meira, pela experiência de mestrando transmitida.

Ao Banco do Nordeste do Brasil S.A., que possibilitou a conclusão deste curso; em especial, ao José Dourivan F. Braga, José Sérgio F. da Costa, José Aurélio B. Lopes, Cláudio Reginaldo Alexandre, Jocélia Maria de C. Alcântara e Anadete Apoliano A. Torres.

Por fim, a todos os meus queridos amigos, agradeço pelo apoio e afeto dispensados durante esta caminhada.

Resumo da dissertação apresentada ao MIA/UNIFOR como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciência da Computação (M.Sc.).

PROCESSAMENTO DE CONSULTAS EM REDES DE SENSORES SEM FIO: UMA ABORDAGEM DE DETECÇÃO DE NOVIDADES PARA O CONTROLE DA QUALIDADE DOS SERVIÇOS DAS CONSULTAS

RESUMO

Redes de Sensores Sem Fio (RSSFs) são redes de comunicação compostas por dispositivos de sensoriamento distribuídos geograficamente, nós-sensores, que podem ser empregadas para monitorar fenômenos físicos em tempo real. Devido à natureza física desses dispositivos, as RSSFs são usualmente propensas a falhas e possuem restrições computacionais para armazenar e processar dados. Apesar disso, o ponto crítico dessas redes está no modo como a energia disponível nos nós-sensores é consumida; se isso for feito apropriadamente, a vida operacional da RSSF poderá ser prolongada. Este trabalho apresenta uma abordagem para adaptar o processamento de consultas em RSSFs baseada em noções de qualidade de serviço das consultas e detecção de novidades. Enquanto o primeiro conceito captura a idéia de se possivelmente ter diferentes consultas servidas de diferentes maneiras em uma mesma RSSF, o segundo refere-se a técnicas de aprendizado de máquina embutidas nos seus componentes para possibilitá-los alterar dinamicamente seus comportamentos durante o processamento de consultas. Essa abordagem foca o consumo inteligente dos limitados recursos dessas redes ao passo que tenta manter a qualidade dos dados esperada pelos seus usuários. Nesse contexto, quatro classes de qualidade de serviço de consultas foram definidas tendo-se em conta níveis distintos de requisitos relativos à acurácia e ao comportamento temporal dos dados coletados. Além disso, um novo algoritmo baseado em detecção de novidades, nomeado *AdaQuali* (*ADaptive QUALity control for query processing in wireless sensor networks*), é introduzido em detalhe como uma forma de controlar as atividades dos nós-sensores, através da adaptação de suas frequências de coletas e transmissões de dados. Para propósitos de validação, experimentos foram conduzidos com um protótipo sobre dados reais, e os resultados apontam para ganhos de conservação de energia que variam de 1,73% a 42,99% para diferentes classes de qualidade de serviço no experimento que varia o número de nós-sensores da rede.

PALAVRAS-CHAVE: Redes de sensores sem fio. Processamento de consulta. Detecção de novidades. Qualidade de serviço da consulta.

Abstract of thesis presented to MIA/UNIFOR as a partial fulfilment of the requirements for the degree of Master of Science (M.Sc.).

QUERY PROCESSING IN WIRELESS SENSOR NETWORKS: A NOVELTY DETECTION APPROACH FOR CONTROL OF QUALITY OF QUERY SERVICES

ABSTRACT

Wireless Sensor Networks (WSNs) are communication networks composed of geographically distributed devices (sensor nodes) which can be used to monitor a variety of physical phenomena in real time. Due to the nature of these devices, WSNs have computational restrictions in terms of data processing and storage, and they are usually prone to failures. The main critical point of these networks, however, is in the way the levels of energy available in the sensor nodes are consumed; if this is done properly, the operational lifetime of the WSN can be prolonged. This work presents an approach for adapting query processing in WSNs based on notions of quality of query service and novelty detection (ND). While the former concept captures the idea of possibly having different queries serviced in different ways by the same WSN, the latter relates to a machine learning technique embedded in the WSN components that allows them to modify their query processing behaviours in a dynamic fashion. This approach aims at the intelligent consumption of the limited resources available in these networks while still trying to deliver the data quality as expected by their users. In this context, four classes of quality of query service have been specified having in mind distinct levels of requirements in terms of accuracy and temporal behaviour of the sensed data. Moreover, a new ND-based algorithm, named AdaQuali (ADAPtive QUALity control for query processing in wireless sensor networks), is introduced in detail as a way to control the activities of sensor nodes through the adjustment of their rates of data collection and transmission. For validation purposes, experiments with a prototype have been conducted over real data, and the results achieved point to gains in terms of energy consumption reduction that vary from 1.73% to 42.99% for different classes of quality of query service in the experiment that varies the number of sensor nodes of the network.

KEY WORDS: *Wireless Sensor Networks. Query processing. Novelty detection. Quality of query services.*

LISTA DE ILUSTRAÇÕES

ILUSTRAÇÃO	Página
Figura 2.1 - Componentes de um nó-sensor. Figura adaptada de Akyildiz <i>et al.</i> (2002, p. 103).....	18
Figura 2.2 - MICA2 (a) e MICAz (b) da Crossbow. Fonte: Crossbow Technology Inc. (2008).....	21
Figura 2.3 - Cricket da Crossbow. Fonte: Crossbow Technology Inc. (2008).....	21
Figura 2.4 - IRIS da Crossbow. Fonte: Crossbow Technology Inc. (2008).....	22
Figura 2.5 - (a) Modelo de dados genérico para aplicações em RSSFs e (b) exemplo de uma instância desse modelo. Figuras adaptadas de Brayner <i>et al.</i> (2007, pp. 2914-2915).....	33
Quadro 2.1 - Cláusulas da SNQL e suas especificações. Quadro adaptado de Brayner <i>et al.</i> (2007, p. 2916).	33
Figura 2.6 - Exemplo de consulta especificada em SNQL. Figura adaptada de Brayner <i>et al.</i> (2007, p. 2916).	35
Figura 2.7 - Diagrama esquemático para processamento de consultas em RSSFs. Figura adaptada de Brayner <i>et al.</i> (2007, p. 2917).....	35
Quadro 2.2 - Requisitos de aplicação. Quadro adaptado de Chen e Varshney (2004)...	39
Figura 3.1 - Ilustração de um conceito emergente percebido como uma novidade.....	46
Figura 3.2 - Exemplo da curva de uma distribuição normal.....	48
Figura 3.3 - Exemplo de uma mistura envolvendo duas distribuições normais.....	49
Quadro 4.1 - Ações que possibilitam a conservação de energia global da rede.	61
Quadro 4.2 - Características gerais das classes de qualidade de serviço da consulta....	62
Quadro 4.3 - Extensão da SNQL (BRAYNER <i>et al.</i> 2007), incorporando a cláusula <i>QUALITY CLASS</i>	64
Figura 4.1 - Exemplo de consulta especificada em SNQL com a cláusula <i>QUALITY CLASS</i>	64
Figura 4.2 - Diagrama esquemático do mecanismo de processamento de consultas em RSSFs com a introdução do <i>AdaQuali</i> . Figura adaptada de Brayner <i>et al.</i> (2007, p. 2917).....	65
Figura 4.3 - Função para adaptação da cláusula <i>SEND INTERVAL</i> . Figura adaptada de Brayner <i>et al.</i> (2008, p. 334).....	68
Figura 4.4 - Função para adaptação da cláusula <i>SENSE INTERVAL</i> . Figura adaptada de Brayner <i>et al.</i> (2008, p. 335).....	69
Figura 4.5 - Diagrama esquemático dos estágios do processamento de consultas com classes de qualidade de serviço executado nos nós-sensores.	70
Figura 4.6 - Pseudocódigo do estágio Processamento de subconsultas.....	75

Figura 4.7 - Pseudocódigo do estágio Processamento de políticas de qualidade.	76
Figura 4.8 - Pseudocódigo do estágio Aprendizagem do conceito.	77
Figura 4.9 - Pseudocódigo do estágio Gerenciamento de recursos.	77
Figura 5.1 - Histogramas gerados para alguns nós-sensores empregados nos experimentos.	81
Figura 5.2 - Consulta SNQL aplicada para avaliar o consumo de energia por classe de qualidade.	82
Figura 5.3 - Energia total consumida por quantidade de nós-sensores para cada classe de qualidade.	83
Figura 5.4 - Percentual médio de coletas efetivamente realizadas por número de nós-sensores para as classes de qualidade <i>Fine-grained Accuracy</i> e <i>Coarse-grained Accuracy</i> com respectivo desvio-padrão.	84
Figura 5.5 - Percentual médio de coletas não enviadas à rede por número de nós-sensores para cada classe de qualidade com respectivo desvio-padrão.	85
Figura 5.6 - Quantidade de mensagens injetadas na rede por número de nós-sensores para cada classe de qualidade.	86
Figura 5.7 - Percentual médio de coletas não enviadas à rede por intervalo de tempo entre coletas para cada classe de qualidade com respectivo desvio-padrão.	88
Figura 5.8 - Percentual de energia total preservada por intervalo de tempo entre coletas para cada classe de qualidade.	88
Figura 5.9 - Percentual médio de coletas efetivamente realizadas por intervalo de tempo entre coletas para as classes de qualidade <i>Fine-grained Accuracy</i> e <i>Coarse-grained Accuracy</i> com respectivo desvio-padrão.	89
Figura 5.10 - Energia total consumida conforme variação do parâmetro β para cada classe de qualidade.	90
Figura 5.11 - Percentual médio de coletas não enviadas à rede por variação do parâmetro β para cada classe de qualidade com respectivo desvio-padrão.	90
Figura 5.12 - Percentual de energia preservada por variação da cláusula <i>SEND INTERVAL</i> para cada classe de qualidade.	91
Figura 5.13 - Percentual médio de coletas não enviadas à rede por variação da cláusula <i>SEND INTERVAL</i> para cada classe de qualidade com respectivo desvio-padrão. .	92
Quadro 5.1 - Quantitativo de coletas para o nó-sensor 54 por cláusula <i>SEND INTERVAL</i> para a classe de qualidade <i>Coarse-grained Accuracy</i>	93
Quadro 5.2 - Cláusulas <i>TIME WINDOW</i> , <i>SEND INTERVAL</i> e <i>SENSE INTERVAL</i> empregadas nas consultas para avaliação do tempo médio de execução do <i>AdaQuali</i>	94
Figura 5.14 - Tempo médio de execução do <i>AdaQuali</i> em milissegundos por tamanho da amostra para a classe <i>Without Outliers</i> com respectivo desvio-padrão.	94

SUMÁRIO

1 INTRODUÇÃO.....	12
1.1 Motivação	12
1.2 Objetivos.....	13
1.3 Organização da Dissertação	15
2 REDES DE SENSORES SEM FIO (RSSFs).....	16
2.1 Conceitos.....	16
2.2 Arquitetura Física do Nó-sensor	18
2.3 Particularidades das Redes de Sensores Sem Fio	22
2.3.1 Comunicação	22
2.3.2 Tolerância a Falhas.....	25
2.3.3 Topologia da Rede	25
2.3.4 Consumo de Energia	27
2.4 Processamento de Consultas em Redes de Sensores Sem Fio.....	28
2.5 Qualidade de Serviço em Redes de Sensores Sem Fio.....	36
2.6 Considerações Finais	40
3 DETECÇÃO DE <i>OUTLIERS</i> E NOVIDADES	41
3.1 Aprendizado de Máquina	41
3.2 Detecção de <i>Outliers</i> e Novidades	44
3.3 Aplicações de Detecção de <i>Outliers</i> e Novidades em Redes de Sensores Sem Fio.....	50
3.4 Considerações Finais	55
4 PROCESSAMENTO DE CONSULTAS EM RSSFs COM CLASSES DE QUALIDADE DE SERVIÇO	56
4.1 Processamento de Consultas em RSSFs via Classes de Qualidade de Serviço de Consultas	57
4.2 AdaQuali	62
4.3 Execução de Consultas nos Nós-sensores com Classes de Qualidade de Serviço	70
4.4 Discussão	77
4.5 Considerações Finais	79
5 EXPERIMENTOS EFETUADOS E SEUS RESULTADOS	80
5.1 Experimentos, Análise e Discussão dos Resultados.....	80
5.2 Considerações Finais	95
6 CONCLUSÃO E TRABALHOS FUTUROS	96
REFERÊNCIAS.....	99
GLOSSÁRIO	105

1 INTRODUÇÃO

1.1 Motivação

Redes de Sensores Sem Fio (RSSFs) são redes de comunicação compostas por dispositivos de sensoriamento, denominados nós-sensores, que são capazes de detectar, coletar e disseminar cooperativamente os dados relacionados a fenômenos físicos (e.g., umidade e pressão) através de uma mídia de transmissão sem fio. Devido à natureza dos seus dispositivos, as RSSFs são propensas a falhas e possuem limitações computacionais para processar e armazenar dados (AKYILDIZ *et al.*, 2002). Contudo, o ponto crítico dessas redes está na pouca disponibilidade de energia nos nós-sensores para que essas permaneçam operacionais. Dessa forma, o consumo racional da energia da bateria dos nós-sensores é fundamental para maximizar a vida útil dessas redes.

O maior consumo de energia em uma RSSF ocorre durante a transmissão e a recepção de dados; por conseguinte, o custo de comunicação é mais oneroso que o custo de processamento (POTTIE; KAISER, 2000 *apud* TILAK *et al.*, 2002, p. 29). Assim sendo, uma estratégia eficaz para preservar a energia é reduzir o número de mensagens transmitidas na rede, o que pode ser alcançado introduzindo técnicas de processamento de consultas de banco de dados e de agregação em rede nas RSSFs. Os mecanismos de processamento de consultas propostos em Brayner *et al.* (2007) e Madden *et al.* (2005) são exemplos da primeira classe de técnicas, enquanto ADAGA (BRAYNER *et al.*, 2006, 2008) e TAG (MADDEN *et al.*, 2002) são exemplos da segunda. Outros exemplos da segunda classe de técnicas podem ser vistos em Nakamura *et al.* (2007).

Outra abordagem, que está mais voltada à aplicação de técnicas de aprendizado de máquina (MARKOU; SINGH, 2003), segue a idéia de induzir um modelo diretamente dos dados coletados para caracterizar o comportamento normal dos dados associados ao fenômeno monitorado (BRANCH *et al.*, 2006; DESHPANDE *et al.*, 2005; ELNAHRAWY; NATH,

2003; MA *et al.*, 2004; PALPANAS *et al.*, 2003; ZHUANG; CHEN, 2006). Como resultado, por exemplo, e conforme características das aplicações e necessidades dos usuários, apenas dados considerados normais (ou anormais) precisariam ser efetivamente enviados pelos nós-sensores. Isso também permite melhorar a qualidade da informação fornecida, já que possíveis irregularidades nos dados referentes ao fenômeno monitorado poderiam ser removidas do resultado da consulta ou mesmo eventos interessantes poderiam ser observados pelos usuários através dos dados anormais.

Embora eficazes em termos de preservação de energia, as abordagens mencionadas de processamento de consultas e de agregação em rede não consideram as especificidades associadas a diferentes aplicações que podem ter acesso a distintos serviços providos por uma mesma RSSF. Isso se deve ao modo como o processamento de consultas é usualmente realizado nessas redes, não considerando diferentes requisitos que podem ser necessários a diferentes usuários com respeito à qualidade dos dados coletados.

1.2 Objetivos

Com o intuito de abordar a questão mencionada na Seção 1.1, esta dissertação apresenta uma nova abordagem de processamento adaptativo de consultas baseada em noções de qualidade de serviço da consulta (levando-se em conta, em especial, a acurácia e o comportamento temporal dos dados coletados) e em detecção de novidades, um campo de pesquisa da área de aprendizado de máquina. Técnicas de detecção de novidades podem ser utilizadas para adequar a consulta nos nós-sensores, mediante a análise comparativa entre o modelo de comportamento normal do fenômeno monitorado e os dados coletados.

O objetivo geral desta dissertação é introduzir uma abordagem para adaptar dinamicamente o processamento de consultas em RSSFs, conforme critérios de qualidade predefinidos, com o intuito de maximizar o período de atividade da rede, por meio do consumo inteligente dos seus escassos recursos durante as atividades de coleta, processamento

e entrega de dados à rede, ao mesmo tempo em que tenta fornecer a informação com o nível de qualidade esperado pelos usuários.

Os objetivos específicos são:

- (i) fornecer um mecanismo que possibilite uma melhoria no processamento de consultas em RSSFs através da utilização de técnicas de detecção de novidades;
- (ii) fundamentar as respostas das consultas em classes de qualidade de serviço, predeterminadas em alto nível nas próprias consultas submetidas à rede, por meio de uma extensão da linguagem de consulta declarativa SNQL (*Sensor Network Query Language*) e da utilização do mecanismo de processamento de consultas que a suporta, ambos propostos por Brayner *et al.* (2007); e
- (iii) adaptar dinamicamente as atividades de coleta e envio de dados à rede, a fim de reduzir os gastos de energia que essas atividades requerem, através do balanceamento entre o consumo de recursos (energia e memória) e os níveis de qualidade esperados.

Para o propósito de tratar diferentes tipos de qualidade de serviço de consultas, quatro classes de qualidade de serviço foram concebidas: (i) *Without Outliers* (sem *outliers*, isto é, sem discrepâncias); (ii) *Fine-grained Accuracy* (acurácia com granulosidade fina); (iii) *Coarse-grained Accuracy* (acurácia com granulosidade grossa); e (iv) *Outlier-Aware* (ciente de *outliers*). Essas classes representam conjuntos distintos de requisitos dos usuários, com respeito à qualidade dos dados sensoriados, processados e entregues à rede, as quais são detalhadas no Capítulo 4.

Para todas as classes de qualidade de serviço de consultas, o consumo de energia da rede poderá ser reduzido já que menos dados trafegarão por ela, seja pela supressão de dados normais (ou anormais) ou pela redução do volume de dados efetivamente sensoriados e difundidos na rede. Pela incorporação de diferentes classes de qualidade de serviço durante o processamento distribuído de consultas, espera-se alcançar ganhos significativos em termos de consumo de energia ao longo da rede.

1.3 Organização da Dissertação

Este trabalho está organizado conforme a seguir. O Capítulo 2 apresenta uma introdução sobre RSSFs e suas características arquiteturais, além dos principais conceitos sobre processamento de consultas e qualidade de serviço nessas redes. O Capítulo 3 traz os aspectos essenciais sobre detecção de *outliers* e novidades, abrangendo propostas elaboradas para RSSFs. O Capítulo 4 descreve como o processamento de consultas nessas redes pode ser adaptado por meio da adoção de classes de qualidade de serviço e de técnicas de detecção de novidades. Nesse capítulo, ainda são detalhadas as quatro classes de qualidade de serviço estabelecidas e o algoritmo concebido para adaptar o comportamento dos nós-sensores conforme a classe. Os experimentos efetuados com um protótipo sobre dados reais são expostos no Capítulo 5. Por fim, o último capítulo conclui este trabalho.

2 REDES DE SENSORES SEM FIO (RSSFs)

Neste capítulo, são apresentados os principais conceitos referentes às RSSFs, objeto de estudo primário desta dissertação. Os fundamentos expostos abordam os diversos aspectos de pesquisa da área, com ênfase no processamento de consultas e na respectiva qualidade de serviço, particularmente, em relação aos dados e ao consumo de energia.

A estrutura deste capítulo é descrita a seguir. A Seção 2.1 apresenta os conceitos gerais relativos às RSSFs. A Seção 2.2 exhibe os aspectos físicos dos dispositivos de sensoriamento. Na Seção 2.3, são detalhadas as particularidades mais relevantes das RSSFs, incluindo: comunicação, tolerância a falhas, topologia de rede e consumo de energia. As Seções 2.4 e 2.5 consideram, respectivamente, o processamento de consultas nessas redes e os conceitos gerais relativos ao tema qualidade de serviço. As considerações finais são descritas na Seção 2.6.

2.1 Conceitos

Nos últimos anos, muitas pesquisas têm sido direcionadas às RSSFs em virtude de possuírem um grande potencial de virem a ser amplamente utilizadas em diversas áreas, tais como: militar, ambiental, médica, doméstica e comercial (AKYILDIZ *et al.*, 2002, p. 102). Essas redes possibilitam o monitoramento do meio onde seus dispositivos de sensoriamento estão inseridos, permitindo a tomada de decisões tão logo os dados sejam disponibilizados aos usuários e aplicações. Em geral, devido a suas características físicas, os nós-sensores podem passar despercebidos e tendem a não interferir no meio onde estão implantados. Algumas possíveis aplicações para RSSFs incluem (AKYILDIZ *et al.*, 2002, p. 102): reconhecimento de forças e de terrenos inimigos, detecção de ataques químicos, detecção de movimento de animais, identificação de queimadas em florestas, automação residencial e monitoração da qualidade das águas, do habitat, de equipamentos, da qualidade de produtos e do tráfego em rodovias.

O sensoriamento, pelo qual os fenômenos físicos podem ser observados, é o principal intuito de qualquer aplicação em uma RSSF, e a entrega dos dados coletados a essas aplicações é a função primária dessas redes. A interface entre a aplicação e a rede ocorre por intermédio de um ou mais pontos da rede, conhecidos como estações-base e igualmente denominados nós-sorvedouros ou *sinks*, que, em geral, são mais robustos do que os nós-sensores (AKYILDIZ *et al.*, 2002; TILAK *et al.*, 2002).

Uma RSSF pode ser composta por uma elevada quantidade de nós-sensores densamente dispostos, podendo essa quantidade ser tão elevada quanto 20 nós/m². Tais nós-sensores são usualmente implantados muito próximos aos fenômenos a serem observados, em ambientes hostis ou não. A localização física exata de cada nó-sensor não precisa ser predeterminada durante a sua implantação; assim, é possível que os nós-sensores sejam dispostos aleatoriamente e até mesmo arremessados em locais de difícil acesso (AKYILDIZ *et al.*, 2002, pp. 102-104; SHIH *et al.*, 2001, p. 272). Em uma RSSF, podem existir múltiplos usuários (ou observadores) que monitoram os fenômenos de interesse por meio da submissão de consultas, e múltiplos fenômenos podem ser observados concorrentemente em uma mesma rede.

Alguns outros aspectos das RSSFs abrangem (AKYILDIZ *et al.*, 2002, p. 102; HILL *et al.*, 2000, p. 94):

- (i) os nós-sensores efetuam esforços cooperativos para o funcionamento da rede;
- (ii) os protocolos de rede devem ter capacidade de auto-organização;
- (iii) os nós-sensores são suscetíveis a falhas;
- (iv) a topologia da rede é frequentemente modificada devido à mudança de localização de um nó-sensor ou mesmo em virtude da sua inclusão, falha ou extinção;
- (v) os nós-sensores utilizam tipicamente o paradigma de comunicação *broadcast*;
- (vi) os nós-sensores possuem restrições de energia, comunicação, processamento e armazenamento; e
- (vii) os nós-sensores, em geral, não sofrem manutenções físicas.

A seguir, são apresentadas as principais características físicas de um nó-sensor.

2.2 Arquitetura Física do Nó-sensor

Um nó-sensor é um dispositivo que realiza o sensoriamento físico de um ou mais fenômenos e os reporta a um usuário ou uma aplicação, permitindo a monitoração de eventos e o apoio na tomada de decisões. É composto por quatro componentes fundamentais e três opcionais como descrito por Akyildiz *et al.* (2002, pp. 103-104). Os quatro componentes essenciais são: (i) unidade de sensoriamento; (ii) unidade de processamento; (iii) transceptor; e (iv) fonte de energia. Já os três componentes opcionais são: (i) gerador de energia, baseado, por exemplo, em calor, luz ou vibrações ambientais; (ii) sistema de locomoção; e (iii) sistema de localização (e.g., um *Global Position System* – GPS). A Figura 2.1 expõe os componentes mencionados e as suas ligações.

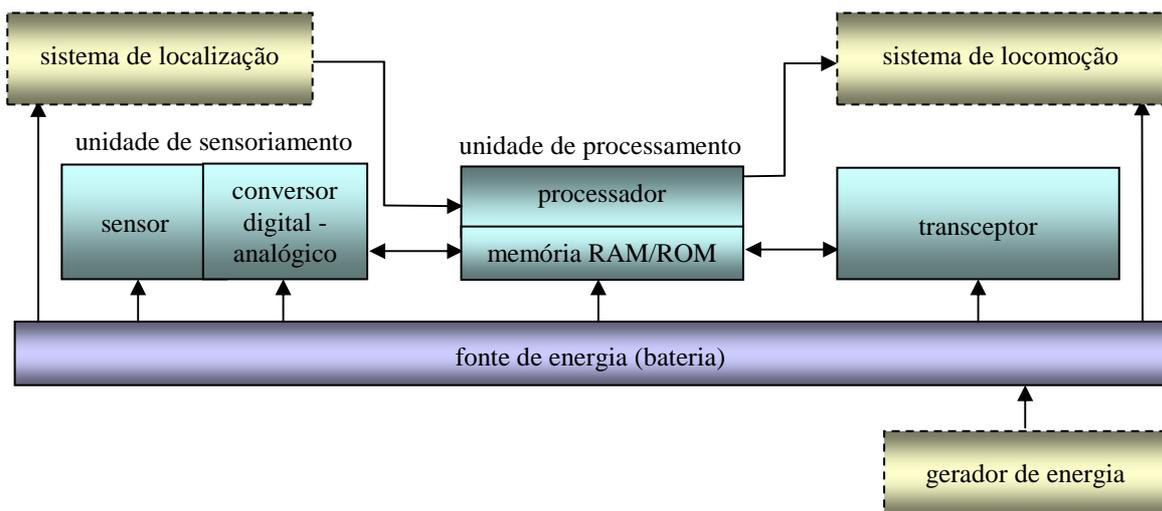


Figura 2.1 - Componentes de um nó-sensor. Figura adaptada de Akyildiz *et al.* (2002, p. 103).

A **unidade de sensoriamento** possui a função de coletar dados sobre os eventos observados e é normalmente formada por um sensor e um conversor digital-analógico. A tarefa desse conversor é transformar os sinais analógicos, originados pelo sensor, em sinais digitais, que serão remetidos à unidade de processamento. A capacidade de sensoriamento dos sensores pode ser afetada em virtude das propriedades físicas do ambiente onde os nós-sensores estão inseridos e da distância entre o sensor e o fenômeno monitorado.

A **unidade de processamento** é responsável por toda a computação ocorrida no nó-sensor, tais como: a execução do conjunto de *software* necessário para o funcionamento da rede, o controle dos sensores, a operação de algoritmos necessários para o processamento colaborativo e a execução de algoritmos para o processamento de dados. De mais a mais, essa unidade geralmente está associada a uma pequena área de armazenamento (memória RAM/ROM) para persistir códigos de programas e dados.

O **transceptor** possibilita a conexão do nó-sensor à rede e é através dele que os sinais são recebidos e transmitidos. Geralmente, o transceptor pode operar em três modos (recepção, transmissão e desligado) e cada um deles demanda uma determinada quantidade de energia.

Um dos componentes mais críticos de um nó-sensor é a **fonte de energia**, a qual é provida por uma bateria usualmente restrita e não reposta, cujo recurso é fornecido aos demais componentes para que possam realizar suas atividades.

Para repor a força consumida, os nós-sensores podem conter um **gerador de energia** que deve ser utilizado com parcimônia, de acordo com as necessidades dos usuários, por aumentar o custo financeiro da rede. Ainda assim, no geral, as fontes de energia são perecíveis e desenvolvidas com tecnologias variadas de fabricação, as quais começam a deteriorar logo após a sua fabricação, sendo os componentes mecânicos e químicos a causa do seu esgotamento. A taxa de desgaste é influenciada pela intensidade da descarga, pelas condições ambientais, pelos procedimentos de carga e pelos procedimentos de manutenção (MINI; LOUREIRO, 2008, p. 56). Até o presente, as fontes de energia dos nós-sensores usualmente não possuem um mecanismo para repor a energia; portanto, são consideradas finitas em geral.

Segundo o domínio da aplicação, os nós-sensores devem conter características específicas para prover informações sobre as suas localizações físicas ou mesmo para se locomoverem para a coleta dos dados. Essas características são fornecidas por intermédio dos componentes **sistema de localização** e **sistema de locomoção**, respectivamente (AKYILDIZ *et al.*, 2002, pp. 103-104).

Interessante notar que as restrições de *hardware* das RSSFs estão diretamente relacionadas às restritas capacidades computacionais dos nós-sensores – processador e memória – e ao limitado suprimento de energia, impostos pelas tecnologias existentes no momento, além das pequenas dimensões físicas desses dispositivos. Dessa forma, as atividades ocorridas na rede devem minimizar o consumo da energia, mantendo-a operacional o maior tempo possível. Outrossim, o nó-sensor deve ter um custo reduzido e ser dispensável, autônomo e adaptativo ao ambiente. Esses quesitos são muito importantes principalmente para as aplicações em que o acesso ao ambiente é restrito, dificultando ou mesmo impossibilitando a reposição dos nós-sensores.

A seguir, três modelos de nós-sensores para RSSFs comercializados pela Crossbow Technology Inc. (2008) são exibidos. A Crossbow oferece um amplo conjunto de produtos para essas redes.

MICA

Os sensores da geração MICA foram inicialmente projetados pela Universidade da Califórnia, Berkeley. O MICA2 é a terceira geração do modelo MICA. Suporta o sistema operacional TinyOS (LEVIS, 2006, TINYOS, 2004) que é um produto de código aberto específico para as RSSFs.

O MICA2 possui um processador Atmel Atmega128L (ATMEL CORPORATION, 2007) que consome 8mA em modo ativo e menos que 15 μ A em modo *sleep*. Nesse último modo, o tempo de vida da bateria é estimado em 1 ano. A unidade de armazenamento é composta por 128KB de memória *Flash* para armazenamento de programas, 512KB de memória *Flash* para persistência de dados, podendo comportar mais de 100.000 medições, e 4KB de SRAM. O rádio transceptor opera a 868/916MHz com taxa de transmissão de 38,4Kbps, corrente máxima de transmissão de 27mA com capacidade para atingir distâncias de 150m em ambientes externos, corrente de recepção de 10mA e corrente em modo *sleep* menor do que 1 μ A. A energia é suprida por duas baterias do tipo AA.

Da mesma forma que o MICA2, o MICAz é baseado em um processador Atmel Atmega128L (ATMEL CORPORATION, 2007), possuindo as mesmas características de armazenamento e de consumo e suprimento de energia. O MICAz pode alcançar distâncias de até 100m em ambientes externos. O rádio transceptor opera entre 2400 e 2483,5MHz com taxa de transmissão de 250Kbps, corrente de 19,7mA em modo de recepção, 20 μ A em modo ocioso (*idle*), 1 μ A em modo *sleep* e corrente máxima para transmissão de 17,4mA.

A Crossbow fornece uma variedade de placas de sensores compatíveis com os modelos MICA2 e MICAz para medir temperatura, luminosidade, pressão atmosférica, aceleração, campo magnético, entre outros. Ambos os modelos podem ser vistos na Figura 2.2.



Figura 2.2 - MICA2 (a) e MICAz (b) da Crossbow. Fonte: Crossbow Technology Inc. (2008).

Cricket

O Cricket (Figura 2.3) é uma versão do MICA2 ciente da localização produzido em conjunto com o MIT (*Massachusetts Institute of Technology*), com todas as características padrão de *hardware* do MICA2.



Figura 2.3 - Cricket da Crossbow. Fonte: Crossbow Technology Inc. (2008).

IRIS

Algumas melhorias foram adicionadas ao IRIS (Figura 2.4) em relação à geração MICAz: maior alcance de transmissão – até 500m em ambiente externo – e 8KB de SRAM.



Figura 2.4 - IRIS da Crossbow. Fonte: Crossbow Technology Inc. (2008).

Na próxima seção, são relatados os principais aspectos das RSSFs, muitos dos quais têm sido fortemente objetos de estudos na área.

2.3 Particularidades das Redes de Sensores Sem Fio

2.3.1 Comunicação

Conforme descrito na Seção 2.2, a função de uma RSSF é reportar a informação até o observador, o qual não está necessariamente consciente da infra-estrutura dessa rede, ou seja, dos nós-sensores e dos seus estados correntes de implantação (TILAK *et al.*, 2002, pp. 29-30). Uma RSSF forma um caminho entre o fenômeno observado e o observador, sendo que o objetivo do protocolo de rede é criar e manter esse caminho (ou múltiplos caminhos) sob condições dinâmicas, enquanto considera os requisitos de pouco consumo de energia, alta acurácia, tolerância a falhas e baixa latência. Latência pode ser definida como a diferença entre o tempo em que o primeiro pacote relacionado ao evento é gerado pelos nós-sensores e o tempo em que o último pacote relacionado a esse evento, ou o último pacote utilizado para

tomar uma decisão, alcança a estação-base (CHEN; VARSHNEY, 2004, TILAK *et al.*, 2002, p. 32).

Nas RSSFs, a comunicação entre os nós é limitada pela capacidade de transmissão da mídia. A comunicação pode ocorrer, por exemplo, mediante transmissão a rádio, a *laser* ou a infravermelho. Cada nó-sensor comunica-se diretamente com outros nós-sensores, localizados na sua vizinhança, cujas mensagens são transmitidas de nó-a-nó da estação-base aos destinos e vice-versa. Segundo Chen e Varshney (2004), essa comunicação é desbalanceada para a maioria das aplicações em RSSFs porque o fluxo segue principalmente de um grande número de nós-sensores a um pequeno conjunto de estações-base. Além de outros fatores, a comunicação entre dois nós-sensores pode ser rompida caso a distância entre eles ultrapasse o limite máximo que um dos nós possui para enviar um pacote de dados, o que modificaria a topologia da rede. Como exposto na Seção 1.1, as tarefas de comunicação – transmissão e recepção de dados – são as que consomem o maior volume de energia dos nós-sensores, e o consumo está diretamente relacionado à distância entre os nós receptor e transmissor, bem como o caminho a ser percorrido até a estação-base, interferindo no consumo de energia total da rede.

Uma RSSF deve considerar a si mesma como uma entidade única, onde os protocolos de comunicação colaborativa removem redundâncias de comunicação, preservando a energia residual (GARCÍA-HERNÁNDEZ *et al.*, 2007, p. 267). Essas redundâncias são peculiares a essas redes por conta da elevada densidade dos nós; conforme o número de nós-sensores aumenta, as rotas de comunicação também aumentam. Assim, os protocolos de rede devem estar aptos a operar com uma grande quantidade de nós-sensores sem que o desempenho da rede seja fortemente afetado, ou seja, a rede deve ser escalável permitindo absorver acréscimos em suas atividades de maneira uniforme.

Segundo Tilak *et al.* (2002, pp. 31-32), a comunicação em uma RSSF pode ser classificada em duas categorias, e um protocolo de rede deve ser capaz de suportá-las já que é responsável por toda a comunicação:

- (i) **Comunicação de aplicação:** refere-se à transferência do dado sensoriado, ou a informação obtida dele, com o objetivo de notificar o usuário. Esse tipo de comunicação pode ser cooperativo ou não-cooperativo. No modelo cooperativo, os nós-sensores se comunicam entre si para realizar um trabalho colaborativo a fim de atender ao interesse do usuário; no não-cooperativo, os nós-sensores disseminam a informação independentemente. A comunicação de aplicação pode ser otimizada por meio da divulgação das informações a uma taxa mínima que satisfará os requisitos de latência e acurácia considerando as habilidades dos nós-sensores e a qualidade dos caminhos entre os nós e o observador; e
- (ii) **Comunicação de infra-estrutura:** diz respeito à comunicação necessária para configurar, manter e otimizar as operações na rede, isto é, a comunicação entre os nós-sensores. Esse tipo de comunicação relaciona-se à capacidade da rede em descobrir caminhos e observar fenômenos eficientemente apesar das características de mobilidade e falha inerentes às RSSFs, mantendo-as funcionais, assegurando a sua robustez em ambientes dinâmicos e otimizando o desempenho operacional da rede. Essa categoria de comunicação representa a sobrecarga das atividades da rede, sendo fundamental minimizar o seu custo e ao mesmo tempo garantir que a rede possa suportar uma comunicação de aplicação eficaz.

A comunicação pode ser organizada em três maneiras distintas (GARCÍA-HERNÁNDEZ *et al.*, 2007, p. 265):

- (i) **Centrada no nó:** é a mais popular, sendo correntemente utilizada na Internet;
- (ii) **Centrada nos dados:** tende a prover uma solução em que os nós são endereçados pelos dados, por exemplo, o tipo do sensor e a área geográfica (TILAK *et al.*, 2002, p. 34), não necessitando de uma identidade, como o endereço da localização topológica na rede (e.g., “Onde estão os nós-sensores cuja temperatura atual excede a 30 graus?”). Entretanto, a habilidade para endereçar um determinado nó-sensor poderá ser necessária em algumas situações, como para averiguar os dados de um sensor em especial (ESTRIN *et al.*, 1999, pp. 264-265); e
- (iii) **Centrada na posição:** não requer nós específicos para a comunicação, utilizando qualquer um que possa prover a conectividade.

A comunicação centrada nos dados e a centrada na posição são mais escaláveis, melhores adaptáveis às aplicações e conceitualmente mais apropriadas em muitos casos (GARCÍA-HERNÁNDEZ *et al.*, 2007, p. 265).

2.3.2 Tolerância a Falhas

É possível que os nós-sensores sejam expostos a determinadas condições físicas que favoreçam a ocorrência de falhas. Esses nós podem ser inseridos em locais para monitorar batalhas, furacões, vulcões, abalos sísmicos, entre outras aplicações, propiciando interferências nos sensoriamentos dos fenômenos ou mesmo danificando definitivamente o dispositivo. Outro fator relevante que favorece a ocorrência de falhas refere-se à baixa disponibilidade de energia do nó-sensor (AKYILDIZ *et al.*, 2002, p. 103).

Apesar das suscetibilidades citadas, as RSSFs devem ser confiáveis e tolerantes a falhas fazendo com que as anomalias sucedidas em alguns nós-sensores não interfiram no comportamento geral da rede, garantindo a entrega dos dados reportados e mantendo a rede operacional. Uma maneira de gerenciar tolerâncias a falhas, com o objetivo dos dados alcançarem o destino final com uma maior probabilidade, é através da replicação dos dados; contudo, amplia-se o consumo de energia (TILAK *et al.*, 2002, p. 29). A alta densidade dos nós também permite o aumento de tolerância a falhas da rede, além de prover uma cobertura adequada (DELICATO *et al.*, 2005, p.38).

2.3.3 Topologia da Rede

A topologia das RSSFs pode mudar constantemente devido à mudança de localização dos nós-sensores (e.g., por possuírem mecanismos de mobilidade ou sofrerem ações externas naturais ou não), ao mau funcionamento (e.g, bateria fraca, ambiente agressivo, acidente, falha na rota de comunicação), à destruição ou adição de novos nós, ou mesmo em virtude do

desligamento intencional de um nó-sensor (AKYILDIZ *et al.*, 2002, p. 104, CHEN, VARSHNEY, 2004). Essas ocorrências induzem a reorganização da rede a fim de se adaptar às novas condições, proporcionando uma dinamicidade expressiva.

As RSSFs podem ser estáticas ou móveis (TILAK *et al.*, 2002, pp. 32-33), e muitas abordagens para criar e manter o caminho entre o observador e o fenômeno monitorado estão diretamente relacionadas a essas arquiteturas. Nas arquiteturas estáticas, tanto o observador quanto o fenômeno em observação e os nós-sensores não se locomovem (e.g., uma RSSF para medir a temperatura das salas em um prédio comercial). Nas arquiteturas móveis ou dinâmicas, o observador, o fenômeno monitorado e/ou os nós-sensores são móveis, o que pode gerar uma ruptura no caminho conhecido ao se locomoverem, e essas entidades podem se conectar ou deixar a rede a qualquer momento. Desse modo, na ocorrência de uma locomoção, as rotas de comunicação devem ser reconstruídas a fim de preservar os interesses do usuário na presença da mobilidade.

A topologia lógica das RSSFs influencia diretamente a estratégia com que os nós-sensores disseminam os dados, e essa topologia pode ser classificada em três tipos (PERRIG *et al.*, 1998 *apud* DELICATO, 2005, p. 23):

- (i) **Baseada em comunicação direta:** nos protocolos baseados em comunicação direta, cada nó-sensor envia seus dados diretamente à estação-base; logo, o consumo de energia não é otimizado. Outrossim, conforme o número de nós-sensores aumenta, as colisões também se ampliam, afetando a qualidade das transmissões dos dados. Em protocolos planos e hierárquicos, tais ocorrências são minimizadas;
- (ii) **Plana:** nos protocolos planos, cada nó-sensor que tenciona enviar seus dados deve encontrar uma rota, estabelecida em múltiplos saltos, até alcançar a estação-base. Uma questão crítica desses protocolos é que, habitualmente, os nós-sensores mais próximos à estação-base tendem a ter a sua energia exaurida mais rapidamente por, normalmente, serem eleitos a participar do processo de comunicação, o que pode inutilizar a rede; e

- (iii) **Hierárquica:** os protocolos hierárquicos têm como princípio a junção de nós-sensores constituindo grupos (*clusters*) que propiciam à rede operar apreciando uma hierarquia específica. Essa característica ocasiona a transmissão dos dados em vários níveis. O uso de grupos nas RSSFs permite que os nós-sensores de um agrupamento colaborem entre si na realização de tarefas; por exemplo, para revezarem nas coletas de dados ou mesmo para agregarem ou resumirem os dados coletados no agrupamento, por intermédio de um líder de grupo (*cluster head*), a fim de que o volume de dados transmitidos na rede seja reduzido (ULMER, 2005 *apud* DELICATO, 2005, p. 30). Como os líderes gastam mais recursos de energia, os protocolos de rede devem ter políticas de alternâncias desses nós a fim de manter a rede operacional por mais tempo. Além disso, por conta da dinamicidade da rede, a formação de grupos e a seleção do líder devem ser repetidas.

2.3.4 Consumo de Energia

Há três tarefas fundamentais que afetam a disponibilidade de energia dos nós-sensores: (i) o sensoriamento; (ii) o processamento; e (iii) a comunicação. Pesquisas têm mostrado que 3.000 instruções poderiam ser processadas com o mesmo custo de energia consumido para enviar 1 *bit* a 100m de distância via rádio (POTTIE; KAISER, 2000 *apud* TILAK *et al.*, 2002, p. 29). Assim, muitos trabalhos direcionados às RSSFs têm como foco a minimização das tarefas dos nós-sensores, principalmente as associadas à transmissão de dados, a fim de prolongar a vida útil dos nós-sensores e, conseqüentemente, da rede, a qual é delimitada por sua capacidade em conservar a energia utilizada para a realização dos serviços.

Destaca-se que, para determinadas aplicações, a reposição da energia gasta pelos nós-sensores ou mesmo a substituição destes poderá ser inexecutável, e a exaustão da energia implica a aniquilação do nó-sensor. A falta de energia ainda poderá vir a impactar a aplicação, caso esta seja significativa para grande parte dos nós-sensores que a compõem. Portanto, técnicas de gerenciamento adequadas para conservar a energia desempenham um papel expressivo no tocante à atividade da rede (AKYILDIZ *et al.*, 2002, p. 104).

Na próxima seção, as principais definições sobre processamento de consultas em RSSFs são expostas. Uma maior ênfase é dada na utilização de técnicas de processamento de consultas de banco de dados empregadas nessas redes.

2.4 Processamento de Consultas em Redes de Sensores Sem Fio

O processamento de consultas em RSSFs envolve o tratamento de seqüências de dados temporalmente ordenadas e não-estruturadas conhecidas como fluxos de dados (*data streams*). Essas seqüências contêm os dados sensoriados, ou seja, comportam os novos dados gerados continuamente pelos diversos nós-sensores distribuídos ao longo da rede por onde cada seqüência de dados transita.

Os fluxos de dados tendem a ser bastante volumosos e, mais cedo ou mais tarde, excederão a capacidade de armazenamento, independentemente do tipo de rede; portanto, são normalmente descartados após o seu uso. Segundo Deshpande *et al.* (2004, p. 588), nas RSSFs, o agrupamento de todos os dados importantes é impossível, dado que o mundo observável fisicamente consiste de um conjunto de fenômenos contínuos no tempo e no espaço, de modo que o conjunto de dados relevantes é, em princípio, infinito. Dessa forma, as tecnologias de sensoriamento coletam dados de fenômenos físicos em pontos discretos no tempo e no espaço. Por isso, qualquer sensoriamento em uma RSSF é aproximado por natureza, no sentido de que representa apenas o estado verdadeiro do ambiente no instante e local onde o dado relacionado ao fenômeno foi adquirido.

De modo geral, as consultas são direcionadas a uma determinada região cobrindo um grupo de sensores que nela se inserem, podendo, inclusive, abranger toda a rede. Algumas consultas também podem ser utilizadas para o gerenciamento e a reconfiguração dos nós-sensores; por exemplo, o usuário poderá examinar dados dos nós-sensores cuja bateria disponível esteja menor do que algum limite ou poderá desativar um nó-sensor segundo alguma necessidade da aplicação (MADDEN *et al.*, 2005, pp. 137-138).

Os dados sensoriados, que podem ser heterogêneos, são obtidos de acordo com as consultas especificadas e enviadas aos nós-sensores através da estação-base. Após o sensoriamento, os dados obtidos são enviados à estação-base através dos nós da rede por meio dos fluxos de dados.

De acordo com o exposto na Seção 1.1, em virtude da energia necessária ao processamento de dados nas RSSFs ser menor do que a energia requisitada para trafegá-los, uma alternativa proposta para o processamento de consultas nessas redes destina-se à redução do tráfego de mensagens por meio de técnicas computacionais aplicadas em rede, permitindo que os nós-sensores tenham um papel ativo no processamento e na disseminação da informação (TILAK *et al.*, 2002, pp. 29, 32). Algumas dessas técnicas incluem: (i) a agregação de dados em rede (BRAYNER *et al.*, 2006, 2008; MADDEN *et al.*, 2002); (ii) a detecção de novidades (BRANCH *et al.*, 2006; DESHPANDE *et al.*, 2005; ELNAHRAWY; NATH, 2003; MA *et al.*, 2004; PALPANAS *et al.*, 2003; ZHUNG; CHEN, 2006), permitindo que somente os dados normais ou anormais sejam reportados ao usuário; e (iii) os agrupamentos de nós-sensores, por exemplo, por proximidade ou similaridade dos dados sensoriados, concebendo que um nó-sensor específico realize tarefas pertinentes ao seu agrupamento enquanto os demais participantes do grupo são preservados (DELICATO, 2005; DELICATO *et al.*, 2005; LAM *et al.*, 2004).

Tecnologias de banco de dados têm sido estendidas com o intuito de serem aplicadas nas RSSFs, trazendo benefícios em termos de preservação de energia e permitindo reduzir o volume de mensagens transmitidas na rede (BRAYNER *et al.*, 2007; MADDEN *et al.*, 2005). Consultas declarativas similares à SQL (*Structured Query Language*) têm sido empregadas como um modelo para aquisição de dados nessas redes, como sugerido em Brayner *et al.* (2006, 2007, 2008), Madden *et al.* (2005) e Shen *et al.* (2001). O processamento de consultas nas RSSFs baseado nessas linguagens declarativas otimiza as consultas gerando planos de execução eficientes, direcionados ao processamento em rede, que possibilitam a redução significativa do consumo de energia e do uso da banda de comunicação por meio da substituição de operações de comunicação por operações de processamento nos próprios nós-sensores (YAO; GEHRKE, 2003, pp. 233-236). Diante disso, as consultas de dados nas RSSFs podem ser realizadas de modo similar às efetuadas em banco de dados relacionais. Os

registros gerados referentes a um mesmo tipo de fenômeno monitorado em diferentes nós-sensores, tal como temperatura ou luminosidade, possuem um único esquema comum e são formados por vários atributos (e.g., identificação do nó-sensor, dado sensoriado e data e hora da coleta), sendo que, assim, podem ser percebidos como um extenso sistema de banco de dados distribuído constituído de múltiplas tabelas (YAO; GEHRKE, 2003, p. 235).

Por intermédio do processamento distribuído de consultas baseado em técnicas de banco de dados, é possível enviar a outros nós-sensores somente as informações que foram efetivamente utilizadas e/ou geradas localmente pelo nó-sensor. É comum o uso de agregação de dados efetuada em rede para minimizar a quantidade de informações transmitidas e, conseqüentemente, otimizar o uso dos recursos computacionais providos. A junção de dados também é outro exemplo de técnica de banco de dados que pode ser empregada durante o processamento distribuído das consultas (YAO; GEHRKE, 2003, p. 240). Assim, tais atividades podem ser parcialmente efetuadas pelos nós-sensores e, posteriormente, concluídas de modo centralizado.

Muito embora alguns autores concebam uma RSSF como um extenso sistema de banco de dados distribuído em que cada sensor é visto como uma fonte de dados, a forma de processamento nas RSSFs difere do processamento tradicional em banco de dados relacionais. Para este, os dados são armazenados fisicamente e o número de acessos a disco é um dos critérios para se estimar o custo da consulta. Já nas RSSFs, apenas parte dos dados é obtida em um determinado espaço e tempo. Ademais, elas não possuem capacidade para armazenar todos os dados sensorizados, e o respectivo paradigma de custo da consulta passa a ser focado no consumo de energia para processá-la: quanto menor, mais eficiente será o processamento. Além do mais, devido às peculiaridades das RSSFs, quesitos referentes à confiabilidade e disponibilidade dos bancos de dados tradicionais não são aplicáveis às RSSFs. Yao e Gehrke (2003, pp. 234-235) destacam ainda que as restrições para processamento e comunicação nessas redes exigem que o processamento de consultas seja ciente dessas restrições e que o sistema deve estar fortemente integrado à camada de rede para o gerenciamento inteligente dos recursos de comunicação. Essas práticas podem ser introduzidas empregando-se algoritmos inteligentes e adaptativos nos próprios nós-sensores, e os algoritmos que possuem

ciência das restrições computacionais das RSSFs são denominados cientes de contexto (*context-aware*). Segundo Delicato (2005, pp. 52, 59), o contexto pode incluir:

- (i) localização do dispositivo;
- (ii) localização relativa a outros dispositivos;
- (iii) características do dispositivo, como capacidade de processamento e energia disponível;
- (iv) ambiente físico, como nível de ruído e largura de banda;
- (v) atividade do usuário, por exemplo, se está ou não em movimento; e
- (vi) valores dos dados monitorados.

Uma consulta é implementada em conformidade com os interesses do usuário, e a forma como as consultas podem ser realizadas governa a geração do tráfego de dados na rede, sendo que as consultas podem ser classificadas nos seguintes modelos de entrega de dados (TILAK *et al.*, 2002, p. 31):

- (i) **Contínuo**: os dados sensorizados são transmitidos constantemente à estação-base, onde o resultado da consulta será disponibilizado ao usuário em consonância com uma taxa de transmissão preestabelecida na consulta contínua. Neste caso, a consulta somente será abortada se o usuário desregistrá-la do sistema;
- (ii) **Dirigido a evento**: a informação somente será remetida à estação-base se um determinado evento de interesse ocorrer na região monitorada; por exemplo, ao se detectar a passagem de um animal;
- (iii) **Iniciado pelo observador**: os nós-sensores somente coletam e reportam seus resultados em resposta a uma requisição explícita do observador a qual contém um tempo delimitado. Assim, após o alcance desse limite, a consulta é finalizada; e
- (iv) **Híbrido**: ocorre quando os diferentes modelos de consulta citados puderem coexistir em uma mesma RSSF.

A seguir, serão descritos a SNQL (BRAYNER *et al.*, 2006, 2007, 2008), uma linguagem de consulta declarativa para RSSFs, e o respectivo mecanismo de processamento distribuído de consultas proposto em Brayner *et al.* (2007) que a suporta. Ambos serão

estendidos com o intuito de dar suporte à nova abordagem de processamento adaptativo de consultas em RSSFs baseada em classes de qualidade de serviço e em técnicas de detecção de novidades, conforme se discutirá no Capítulo 4.

A SNQL faz uso de um modelo de dados que provê uma visão lógica sobre os fluxos de dados injetados na rede. Essa visão possibilita que as aplicações possam perceber os dados como tuplas de relações virtuais, o que consente uma abstração aos usuários dos detalhes físicos da rede, tais como: identificar os nós-sensores relevantes para uma determinada consulta, determinar os dados a serem processados nos nós-sensores ou na estação-base, e aplicar regras de otimização para reduzir o volume de pacotes transmitidos na rede. As principais características da SNQL são:

- (i) permite representar consultas declarativas com base nas informações disponibilizadas pelos nós-sensores;
- (ii) possibilita controlar o volume de dados das consultas e quando estes serão obtidos do ambiente por meio de uma definição prévia;
- (iii) suporta consultas contínuas em que o tempo de processamento é indeterminado e seus resultados são remetidos ininterruptamente à estação-base e disponibilizados incrementalmente; e
- (iv) permite ajustar os valores de algumas cláusulas das consultas injetadas na rede em tempo de execução (*on-the-fly*).

O modelo de dados usado pela SNQL fornece uma visão lógica dos grupos de sensores da rede. Cada grupo de sensores relaciona os sensores que possuem o mesmo tipo de atividade de coleta, por exemplo, temperatura. Tem-se em conta que cada nó-sensor poderá efetuar mais de um tipo de atividade e, portanto, poderá estar associado a mais de um grupo de sensores. Cada grupo de sensores é representado como uma relação (GrupoSensor1, ..., GrupoSensorN) cujas tuplas contêm os atributos pertinentes ao fenômeno monitorado. O modelo também captura as informações relativas às áreas geográficas através da relação de regiões sensoriadas (RegiaoSensoriada). O identificador da região onde o fenômeno foi observado está contido em cada relação de grupo de sensores. A Figura 2.5 ilustra o modelo de dados genérico descrito e um exemplo de uma instância desse modelo cujas tabelas são virtuais e representadas sobre os

fluxos de dados da RSSF. A relação Temperatura contém o identificador da região onde a temperatura foi coletada, o valor da temperatura, o número de coletas realizadas para essa medida e a escala que identifica se a temperatura está em Celsius ou Fahrenheit, por exemplo.

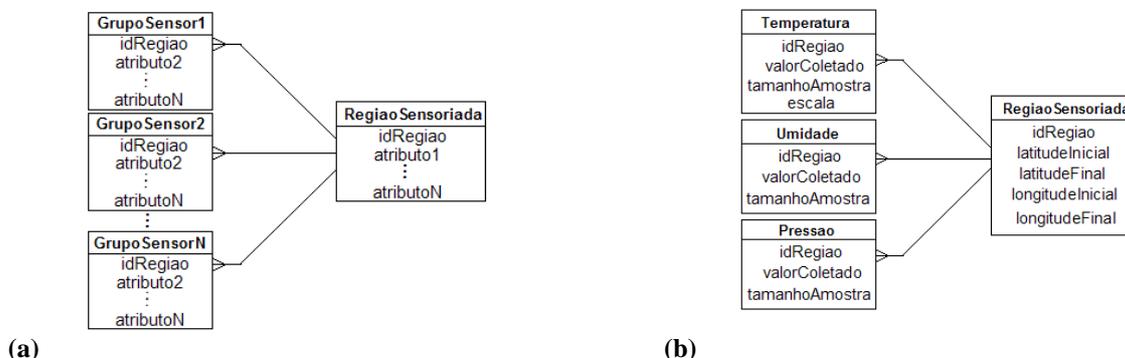


Figura 2.5 - (a) Modelo de dados genérico para aplicações em RSSFs e (b) exemplo de uma instância desse modelo. Figuras adaptadas de Brayner *et al.* (2007, pp. 2914-2915).

O Quadro 2.1 expõe as cláusulas da SNQL com as respectivas especificações.

Cláusulas ^a	Especificação
<i>SELECT</i> {<expr>}	<expr>: especifica um subconjunto de atributos das relações consideradas na consulta.
<i>FROM</i> {<sensor group>}	<sensor group>: define os grupos de sensores participantes da consulta.
[<i>WHERE</i> <pred>]]	<pred>: determina um conjunto de predicados para filtrar tuplas processadas.
[<i>GROUP BY</i> {<exprgroup>} [<i>HAVING</i> {<predhaving>}]]	<exprgroup>: define um conjunto de atributos sobre os quais funções de agrupamento devem ser aplicadas. <predhaving>: especifica predicados baseados nas funções de agrupamento para filtrar resultados agregados.
<i>TIME WINDOW</i> <twseconds> CONTINUOUS	<twseconds>: define o intervalo de tempo durante o qual a consulta é válida para o sistema (por exemplo, 3.600s). O valor predefinido CONTINUOUS especifica que a consulta tem um tempo de validade indeterminado.
[<i>DATA WINDOW</i> <dwnumrows>]	<dwnumrows>: demarca a quantidade máxima de dados a serem coletados por cada nó-sensor (por exemplo, 50.000 coletas).
<i>SEND INTERVAL</i> <sndseconds>	<sndseconds>: define o intervalo de tempo entre dois envios sucessivos de pacotes de dados (por exemplo, 120s).
<i>SENSE INTERVAL</i> <snsseconds>	<snsseconds>: especifica o intervalo de tempo entre duas coletas consecutivas de dados do ambiente pelos nós-sensores (por exemplo, 10s).
[<i>SCHEDULE</i> <numexecutions> [{datetime}] CONTINUOUS]	<numexecutions>: define o número de vezes em que uma consulta deve ser submetida à rede e quando cada submissão deve ocorrer. Se esse valor for maior do que 1, um conjunto de valores de data e hora deve ser informado para determinar quando as execuções deverão ocorrer, por exemplo, 2 ‘10-oct-09 14:00:00’, ‘15-oct-09 14:00:00’, que significa que a consulta deve ser executada duas vezes, uma instância, no dia 10 de outubro de 2009 às 14h, e a outra, no dia 15 de outubro de 2009 também às 14h. O valor predefinido CONTINUOUS especifica que a consulta tem que ser submetida à rede por um número indeterminado de vezes. A cada vez, uma nova instância da consulta é executada.

^a O símbolo “{ }” denota um conjunto, o “[]”, cláusulas opcionais, o “< >”, uma expressão, e o símbolo “|” exprime que um ou o outro deve aparecer, mas não ambos.

Quadro 2.1 - Cláusulas da SNQL e suas especificações. Quadro adaptado de Brayner *et al.* (2007, p. 2916).

As cláusulas *TIME WINDOW* e *DATA WINDOW* são usadas pelos nós-sensores e pela estação-base para que possam identificar quando a consulta deverá ser finalizada. Essas cláusulas determinam o tempo de execução da consulta; a primeira, por meio do intervalo de tempo, e a segunda, através da quantidade de coletas. O valor predefinido *CONTINUOUS* da cláusula *TIME WINDOW* especifica que a consulta será executada por um período indefinido. A cláusula *TIME WINDOW* tem prioridade sobre a cláusula *DATA WINDOW*.

A cláusula *SENSE INTERVAL* indica o intervalo entre coletas de dados consecutivas; assim sendo, quanto maior for o valor demarcado, menor será a precisão do resultado da consulta em virtude de que menos dados serão coletados no tempo e no espaço. Por outro lado, caso ocorra uma especificação de um valor muito baixo, isso poderá ocasionar um maior acúmulo de dados na memória dos nós-sensores e, por conseguinte, um volume maior de dados a serem processados e conduzidos à rede. Da mesma forma que a cláusula *SENSE INTERVAL*, a cláusula *SEND INTERVAL* também afeta o uso da memória em razão de que, quanto maior o seu valor, mais dados coletados terão que ser armazenados para posterior envio à estação-base, tendo em vista que esta cláusula define o intervalo de tempo entre dois envios sucessivos de pacotes de dados para outros nós.

Para o agendamento de uma consulta, a SNQL fornece a cláusula *SCHEDULE* que permite ao usuário definir o número de vezes e a periodicidade com que a consulta deve ser submetida.

As cláusulas que podem ser ajustadas durante a execução da consulta são: *TIME WINDOW*, *DATA WINDOW*, *SENSE INTERVAL* e *SEND INTERVAL*.

A Figura 2.6 mostra um exemplo de uma consulta SNQL com o seguinte propósito: obter o maior valor de temperatura e o menor valor de umidade, delimitados pela região geográfica 3°43'08''S-38°31'51''W e 3°43'16''S-38°31'14''W, sendo que o valor de temperatura seja maior do que 20°C e o valor de umidade menor do que 70%. A consulta será executada por um período de 3.600s, com intervalo entre coletas igual a 10s. Os dados serão enviados à estação-base a cada 120s.

```

SELECT          r.IdRegiao AS RegiaoGeografica, MAX(t.valorColetado),
                  MIN(u.valorColetado)
FROM           Umidade u, Temperatura t, RegiaoSensoriada r
WHERE          r.IdRegiao = t.IdRegiao AND
                  r.IdRegiao = u.IdRegiao AND
                  r.latitudeInicial > 034308 AND
                  r.latitudeFinal < 034316 AND
                  r.longitudeInicial > 383151 AND
                  r.longitudeFinal < 383114 AND
                  t.valorColetado > 20 AND
                  u.valorColetado < 0.7

GROUP BY      r.IdRegiao
TIME WINDOW   3600
SEND INTERVAL 120
SENSE INTERVAL 10
    
```

Figura 2.6 - Exemplo de consulta especificada em SNQL. Figura adaptada de Brayner *et al.* (2007, p. 2916).

Em síntese, o mecanismo de processamento de consultas proposto em Brayner *et al.* (2007) que suporta a SNQL contém quatro fases distintas para processar uma consulta (ver Figura 2.7):

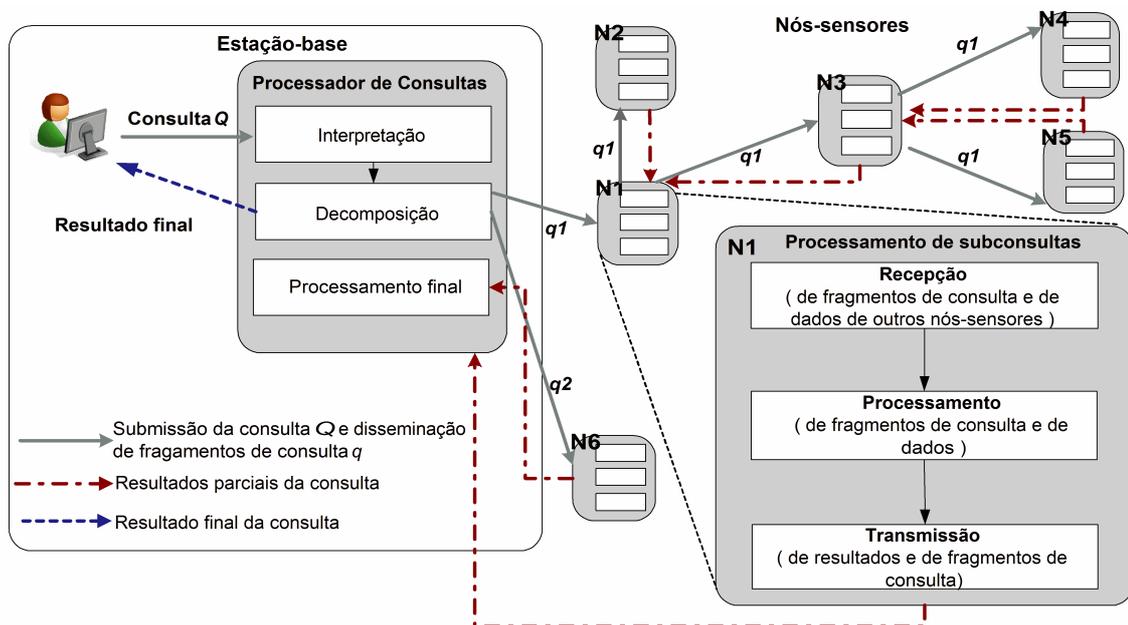


Figura 2.7 - Diagrama esquemático para processamento de consultas em RSSFs. Figura adaptada de Brayner *et al.* (2007, p. 2917).

- (i) **Interpretação:** executada na estação-base, esta fase é responsável por verificar se a consulta Q reproduz uma expressão válida para a SNQL e para o esquema de dados global definido. Em seguida, um plano de execução é gerado;

- (ii) **Decomposição:** também realizada na estação-base, esta fase é iniciada após a execução da etapa *Interpretação*, sendo a sua principal função decompor a consulta Q em subconsultas (fragmentos), a fim de serem enviadas aos respectivos destinos (nós-sensores), que terão a capacidade de avaliar se a subconsulta deverá ou não ser processada pelo nó-sensor;
- (iii) **Processamento de subconsultas:** esta etapa ocorre de modo distribuído nos nós-sensores, e o processamento da subconsulta somente é realizado quando esta for aceita pelo nó-sensor, o que depende dos parâmetros envolvidos. Após a aceitação da subconsulta, o nó-sensor efetua a coleta de acordo com a cláusula *SENSE INTERVAL* e com os parâmetros de projeção e seleção, e ainda poderá realizar agregações em rede. As transmissões dos resultados à estação-base acontecem conforme a cláusula *SEND INTERVAL*, e a consulta somente é concluída se a cláusula *TIME WINDOW* ou *DATA WINDOW* for atendida. Quando a subconsulta é rejeitada, o nó-sensor apenas a encaminha para outros nós-sensores; e
- (iv) **Processamento final:** tem como finalidade preparar e reportar o resultado da consulta. Esta etapa ocorre na estação-base.

2.5 Qualidade de Serviço em Redes de Sensores Sem Fio

No domínio das aplicações computacionais e das redes de comunicação, o termo qualidade de serviço tem sido vastamente utilizado, possuindo, portanto, diferentes perspectivas. No contexto das aplicações computacionais, a qualidade de serviço geralmente diz respeito à qualidade percebida pelos usuários durante o uso das aplicações. Já no âmbito das redes de comunicação, esta é aceita como uma medida da qualidade do serviço que a rede oferece a suas aplicações (CHEN; VARSHNEY, 2004).

No tocante às aplicações computacionais das RSSFs, García-Hernández *et al.* (2007, p. 269) mencionam que as particularidades dessas redes ditam a necessidade para que tanto as suas funcionalidades quanto os nós-sensores sejam controlados para melhor servirem aos

requisitos da aplicação. Tais requisitos, como acurácia e latência dos dados, podem diferir fortemente entre as aplicações (DELICATO *et al.*, 2005), sendo que a qualidade dos dados está associada à intenção de uso do usuário. Strong *et al.* (1997 *apud* WÜTHRICH *et al.*, 2000, p. 297), por exemplo, definem a alta qualidade dos dados como “dados que são apropriados para uso pelos seus consumidores”.

Sobre o aspecto da qualidade dos dados, os sistemas tradicionais de banco de dados, cujos dados são frequentemente alimentados pelos usuários através de sistemas legados, podem conter valores incompletos, errôneos ou faltantes (WÜTHRICH, 2000, p. 296). De maneira similar, a ocorrência de falhas nos nós-sensores, as interferências externas, entre outras situações, podem vir a impactar a qualidade dos dados obtidos via RSSFs, impossibilitando a realização de consultas baseadas em dados com alta qualidade. Assim, a identificação de métodos que possam ser utilizados para prevenir ou mesmo tratar inconsistências de dados pode trazer significativas contribuições ao processamento de consultas nas RSSFs.

Ante o exposto, observa-se que a noção de qualidade de serviço tem variado de aspectos objetivos, como a “habilidade em prover diferentes prioridades em termos de consumo de recursos computacionais para diferentes aplicações”, até aqueles subjetivos, como a “habilidade de atender a requisitos individuais impostos por diferentes usuários em contextos distintos”. Neste trabalho, adota-se o conceito de qualidade de serviço da consulta tendo-se em mente que os serviços de processamento de consultas providos por uma mesma RSSF podem ser customizados para atender mais efetivamente a diferentes necessidades requeridas por diferentes tipos de aplicação; em especial, trata-se a qualidade dos dados e o comportamento temporal das coletas e respectivos envios à estação-base.

No restante desta seção, alguns trabalhos que tratam da qualidade de serviço no âmbito das aplicações computacionais para RSSFs, foco de estudo desta dissertação, são descritos.

Em Delicato (2005) e Delicato *et al.* (2005), pode-se apreciar uma abordagem que emprega um *middleware* baseado em serviços, sem utilizar técnicas de banco de dados, para selecionar um conjunto de nós que tenta satisfazer a qualidade especificada para a aplicação e

ao mesmo tempo maximizar a vida útil da rede. Essa maximização se dá por meio do consumo uniforme de energia dos nós-sensores, sendo que o serviço prioriza os nós-sensores com maior energia residual e relevância para a aplicação, reduzindo o número de nós ativos que atendam à qualidade desejada, enquanto tenta garantir a cobertura e a conectividade da rede. Desse modo, o serviço provê uma topologia de rede virtual para a coleta de dados e o respectivo roteamento. Para a seleção dos nós-sensores, o serviço considera a relevância do nó-sensor em fornecer a acurácia solicitada, que depende da sua precisão física, do ruído ambiental de suas medições, do número de nós-sensores na vizinhança e da proximidade em relação à área-alvo. O algoritmo poderá ser executado nos nós-sensores líderes de agrupamentos ou na estação-base.

Deshpande *et al.* (2004, 2005) expõem um modelo probabilístico que possibilita ao usuário informar limites de tolerância de erros e de confiança por intermédio de consultas similares a bancos de dados relacionais tradicionais (SQL). O modelo probabilístico, cujo aprendizado ocorre de forma centralizada, é usado para estimar as coletas dos sensores no período de tempo corrente, o que formará a resposta à consulta. A aplicação avalia se esse modelo tem precisão suficiente para responder à consulta, com confiança aceitável, sem acessar a rede, permitindo conservar a energia disponível. Caso não tenha, um plano de observação será gerado e injetado na rede, que especifica como os dados da consulta serão obtidos. As coletas efetuadas são posteriormente usadas para atualizar o modelo. Se o usuário requisitar 100% de confiança e nenhuma tolerância a erros, a consulta será executada abrangendo todos os nós-sensores solicitados.

Em Lam *et al.* (2004), é apresentada uma abordagem probabilística para selecionar uma quantidade mínima de nós-sensores para a obtenção de dados que atendam aos requisitos probabilísticos das consultas com o intuito de proporcionar respostas confiáveis às mesmas e preservar o uso de nós-sensores (e.g., “Que salas possuem 60% de chance de ter uma temperatura maior do que 20°C?”). Os nós-sensores que supostamente não estão funcionando apropriadamente são descartados, em vez de se limpar ou calibrar individualmente os valores coletados. Em especial, o trabalho é concentrado em consultas de agregação – soma, média, mínimo, máximo e contagem – processadas parcialmente por nós-coordenadores (líderes de

agrupamentos) com alta capacidade computacional e responsáveis por gerenciar os nós-sensores requeridos pela consulta.

Em Chen e Varshney (2004), são expostos requisitos de qualidade de serviço de aplicações, efetuando-se um cruzamento entre quatro classes de aplicações e três modelos de consultas para RSSFs. As classes de aplicações consideradas são: interativa ou não-interativa, tolerante ou intolerante a atrasos, crítica ou não-crítica e fim-a-fim ou não-fim-a-fim. Já os modelos de consultas abrangidos são: dirigido a evento, iniciado pelo observador e contínuo. As aplicações para RSSFs são não-fim-a-fim, pois um dos seus fins é a estação-base e o outro usualmente não é um único nó-sensor, mas um grupo de nós-sensores. O Quadro 2.2 exibe o referido cruzamento.

Classe de aplicação	Modelo de consulta		
	Dirigido a evento	Iniciado pelo observador	Contínuo
Interativa	Sim	Sim	Não
Tolerante a atrasos	Não	Conforme consulta	Sim
Crítica	Sim	Sim	Sim
Fim-a-fim	Não	Não	Não

Quadro 2.2 - Requisitos de aplicação. Quadro adaptado de Chen e Varshney (2004).

A maioria das aplicações que aplicam o **modelo dirigido a evento** são interativas, intolerantes a atrasos, críticas e não-fim-a-fim. Isso significa que os eventos que os nós-sensores estão programados para monitorar são muito importantes e devem ser detectados pela aplicação a fim de possibilitar a tomada de ações apropriadas tão rápida e confiavelmente quanto possível. Com respeito às aplicações que utilizam o **modelo iniciado pelo observador**, essas são interativas, tolerantes a atrasos conforme a consulta, críticas e não-fim-a-fim, sendo ainda necessário receberem os dados desejados de forma fidedigna. Já as aplicações com **modelo contínuo** são não-interativas, tolerantes a atrasos, críticas e não-fim-a-fim (CHEN; VARSHNEY, 2004). Assim sendo, tanto o **modelo iniciado pelo observador** quanto o **modelo contínuo** permitem o emprego de algoritmos adaptativos que ajustem o comportamento temporal das consultas processadas.

2.6 Considerações Finais

O presente capítulo abordou as principais características das RSSFs: arquitetura física dos nós-sensores, comunicação, topologia de rede, consumo de energia e tolerância a falhas. Adicionalmente, alguns modelos de nós-sensores para essas redes foram apresentados. Neste capítulo, ainda se destacou o processamento de consultas e a qualidade de serviço nessas redes, sendo expostos alguns trabalhos referentes a esses dois temas.

No próximo capítulo, são apresentados conceitos gerais sobre detecção de *outliers* e novidades, além de trabalhos realizados para as RSSFs relativos a esse tema.

3 DETECÇÃO DE *OUTLIERS* E NOVIDADES

O presente capítulo apresenta, na Seção 3.1, os principais conceitos referentes ao aprendizado de máquina, que tem como um de seus campos de pesquisa a detecção de novidades. Em seguida, são expostos os fundamentos sobre detecção de *outliers* e novidades na Seção 3.2. Na Seção 3.3, alguns trabalhos que consideram essas técnicas no contexto das RSSFs também são mencionados, enquanto as considerações finais deste capítulo são relatadas na última seção.

3.1 Aprendizado de Máquina

Aprendizado de máquina é um domínio da inteligência artificial e tem por objetivo a elaboração de teorias computacionais sobre o aprendizado bem como a construção de sistemas capazes de adquirir conhecimento de forma automática. Um **sistema de aprendizado de máquina** é um programa de computador que toma decisões baseadas em experiências acumuladas por meio da solução bem-sucedida de problemas anteriores (MONARD; BARANAUSKAS, 2003, p. 89).

O aprendizado de máquina ocorre sobre a população de dados cujo conhecimento se deseja obter. Para isso, é preciso que um **programa de aprendizado de máquina**, também conhecido como **algoritmo de indução** ou **indutor**, seja executado. O objetivo de tais algoritmos é extrair o conhecimento referente aos conceitos inerentes a uma população a partir de um conjunto de observações de treinamento dessa população durante a fase de treinamento (MONARD; BARANAUSKAS, 2003, p. 95). Em geral, o conjunto de observações de treinamento é formado por uma parte da população, normalmente, em decorrência da impossibilidade ou inviabilidade de sua abrangência total.

Uma **observação**, também nomeada caso, instância, registro, exemplo ou mesmo dados, refere-se a um único item do conjunto de observações que o algoritmo de indução irá utilizar para efetuar o aprendizado. As observações freqüentemente são vetores com tamanho fixo e com características (atributos) predeterminadas e, portanto, a característica é uma propriedade de uma observação. Se relacionada aos aspectos de um banco de dados relacional, uma observação equivale a uma tupla de uma relação. Cada observação é um exemplo do conceito a ser aprendido (FRAWLEY *et al.*, 1992, p. 60; WITTEN; FRANK, 2005, p. 45).

No contexto de aprendizado de máquina, o termo **conceito** é importado da área de psicologia cognitiva e alude ao processo de categorização mental de classes de objetos por meio de suas características gerais. Em particular, a função de um algoritmo de aprendizado indutivo é a de aprender uma maneira de sumarizar os dados de treinamento de uma população de objetos, identificando as categorias (conceitos) subjacentes a estes. Essa sumarização, que é representada por uma linguagem de descrição de conceito, pode ser utilizada para categorizar novas observações (objetos) dessa mesma população. Um exemplo seria determinar o grau de risco financeiro de uma possível contratação a ser firmada por uma empresa. Isso poderia ser realizado através de uma sumarização extraída de um conjunto de dados de treinamento contendo negociações anteriores (que seriam, neste caso, os "objetos") dessa empresa. Esse conjunto de dados poderia ser composto pelo perfil dos clientes com respectivos contratos e seus pagamentos. Por meio das características da contratação a ser firmada, seria possível categorizar o seu grau de risco com base no conhecimento extraído pelo algoritmo de indução, enquadrando-a na categoria que melhor a representar (WITTEN; FRANK, 2005, p. 42).

Na **fase de treinamento**, o algoritmo de indução é executado sobre as observações de treinamento de uma população, produzindo o **modelo de conceito normal global**, doravante modelo de conceito normal. Esse modelo, também denominado descrição de conceito, é a hipótese gerada sobre esse conjunto, de modo que, dada uma nova observação dessa mesma população, o sistema possa classificá-la com a maior precisão possível (MONARD; BARANAUSKAS, 2003, p. 97). Além de o sistema de aprendizado de máquina poder inferir a classificação de uma nova observação recebida após a fase de treinamento, também poderá

continuar a aprender de forma incremental, aumentando o conhecimento inicialmente absorvido.

O aprendizado de máquina pode ser classificado basicamente em três modos:

- (i) **Aprendizagem supervisionada:** são fornecidas, ao algoritmo de indução, observações de treinamento para as quais a classe correta é conhecida. Uma **classe** descreve o conceito-meta que se deseja aprender e pertence tipicamente a um conjunto finito de classes. O objetivo é construir um classificador que determine corretamente a classe de novas observações, por exemplo: identificar se um produto é defeituoso ou não. Este tipo distingue-se pela existência de um supervisor para monitorar os resultados durante o treinamento (MONARD; BARANAUSKAS, 2003, p. 91, 96, BRAGA *et al.*, 2003, pp. 149-150);
- (ii) **Aprendizagem não-supervisionada:** o algoritmo de indução analisa as observações e tenta agrupá-las de alguma maneira, formando *clusters* (CHEESEMAN; STUTZ, 1999 *apud* MONARD; BARANAUSKAS, 2003, p. 91). Neste aprendizado, as saídas desejadas inexistem *a priori* (BRAGA *et al.*, 2003, p. 150); e
- (iii) **Aprendizagem por reforço:** este tipo de aprendizagem pode ser considerado um paradigma intermediário entre a aprendizagem supervisionada e a não-supervisionada. Caracteriza-se ainda pela inexistência de saídas desejadas, mas existe um crítico externo, em substituição ao supervisor da aprendizagem supervisionada, que retorna um sinal de penalidade ou reforço vinculado à última ação. Se houver degradação de desempenho, a ação será penalizada ocorrendo com menor chance em situações futuras; caso contrário, será reforçada (BRAGA *et al.*, 2003, p. 150).

3.2 Detecção de *Outliers* e Novidades

A detecção de novidades é um campo de pesquisa da área de aprendizado de máquina que está próximo ao campo de pesquisa de detecção de *outliers* da estatística (SPINOSA *et al.*, 2007, p. 448).

Técnicas envolvendo a identificação de observações cujas características não se enquadram na distribuição conhecida dos dados vêm sendo desenvolvidas em áreas de conhecimento como estatística e aprendizado de máquina (HODGE; AUSTIN, 2004; MARKOU; SINGH, 2003). Uma das abordagens para se detectar que uma observação não se enquadra na distribuição conhecida dos dados é o uso de classificadores para atuar como detectores, sendo que modelos de agrupamentos também podem ser usados para classificar observações. Basicamente, verifica-se se a nova observação condiz com alguma das classes conhecidas contidas no modelo de conceito normal; em caso negativo, a detecção ocorrerá (MARKOU; SINGH, 2003, p. 2481). Essa abordagem tem gerado uma gama de denominações distintas como: detecção de novidades, detecção de anomalias, detecção de ruídos, detecção de desvios, detecção de intrusos e detecção de *outliers* (pontos discrepantes) (ALMEIDA, 2006; BRANCH *et al.*, 2006; ELNAHRAWY; NATH, 2003; MA *et al.*, 2004; MARKOU; SINGH, 2003; MUKHERJEE; SEN, 2007; PALPANAS *et al.*, 2003). Desse conjunto de denominações, duas nomenclaturas são adotadas neste trabalho – detecção de *outliers* e detecção de novidades – as quais são definidas a seguir.

Tradicionalmente, a **detecção de *outliers*** tem sido utilizada para detectar observações que desviam da distribuição conhecida dos dados e removê-las quando apropriado, evitando o seu efeito indesejado sobre o conjunto de dados e, assim, purificando-o para a realização do aprendizado (HODGE; AUSTIN, 2004, p. 85). No entanto, a detecção de *outliers* pode ser usada para efetuar outras atividades além da remoção da observação quando conveniente. Dessa maneira, de modo mais abrangente, um *outlier* é uma observação que desvia notadamente dos outros membros da amostra a que pertence de acordo com o modelo de conceito normal para essa amostra (BARNETT; LEWIS, 1994 *apud* HODGE; AUSTIN, 2004, p. 86). Essa detecção pode vir a indicar, por exemplo, uma doença, e pode ser tratada segundo o domínio da aplicação e as necessidades dos usuários. É importante ressaltar a

existência de outras definições para o termo *outlier* que podem ser verificadas em Barnett e Lewis (1994 *apud* HODGE; AUSTIN, 2004, p. 86); todavia, neste trabalho, será apreciada a definição supracitada.

Por outro lado, de acordo com o exposto em Spinosa *et al.* (2007, p. 448), a **detecção de novidades** diz respeito à identificação de novos conceitos para um sistema de aprendizado de máquina, conceitos que não foram assimilados durante a fase de treinamento e que introduzem características que diferem significativamente daquelas relacionadas ao modelo de conceito normal. Essa aptidão é fundamental para um bom sistema de aprendizado de máquina, já que, geralmente, não é possível abranger todas as possibilidades durante a fase de treinamento. Ademais, a detecção de novidades possibilita a identificação de conceitos até então desconhecidos com base em medidas de distância e limites partindo do modelo de conceito normal (MARKOU; SINGH, 2003, p. 2481).

Conforme Spinosa *et al.* (2007, p. 448), além da detecção de novidades, os sistemas de aprendizado de máquina que operam sobre fluxos de dados ou consideram a aprendizagem contínua deveriam ser robustos o suficiente para detectar as situações de **mudança de conceito**, referenciadas na literatura por *concept drift*. Tais situações são interessantes de serem detectadas, em razão dos comportamentos dos dados poderem variar com o passar do tempo. Assim, mediante a adoção de técnicas de mudança de conceito e detecção de novidades, torna-se possível determinar se o conceito global conhecido (normal) está passando por uma mudança ou se um novo conceito está surgindo. Desse modo, a distinção entre *outliers*, novidades e mudanças de conceito efetuada neste trabalho segue uma abordagem equivalente à contida em Spinosa *et al.* (2007). Um conceito emergente será tratado como uma mudança de conceito quando este estiver próximo às fronteiras do modelo de conceito normal, pois aparenta ser uma ocorrência de mudança no conceito normal. Por outro lado, um conceito emergente que está distante dessas fronteiras pode representar um novo conceito. Ademais, tanto a novidade quanto a mudança de conceito, ou seja, a mudança comportamental dos dados, são fundamentadas em um conjunto de observações identificadas como *outliers* em vez de uma única observação.

Para ilustrar uma detecção de novidade, a Figura 3.1 exibe um modelo de conceito normal representado pelos círculos de cor amarela (1), laranja (2) e verde (3). O círculo de cor roxa (4) denota um conceito emergente, enquanto o círculo de cor cinza delimita as fronteiras do modelo de conceito normal. Como o conceito emergente não está contido na região do círculo representado pela cor cinza, este será percebido como uma novidade, caso contrário, seria considerado uma mudança de conceito.

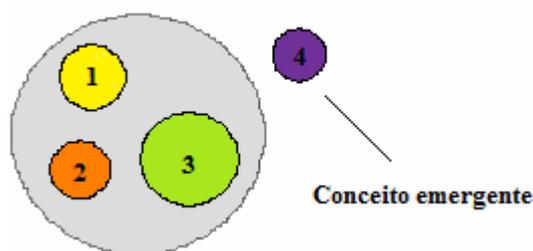


Figura 3.1 - Ilustração de um conceito emergente percebido como uma novidade.

A seguir, serão abordados os aspectos relacionados ao paradigma de aprendizado estatístico; em especial, dar-se-á ênfase sobre a distribuição gaussiana, abordagem da qual se fará uso na proposta apresentada no Capítulo 4 para se detectar *outliers*, novidades e mudanças de conceito.

A estatística moderna pode abranger dois tipos de enfoques para a inferência estatística: o paramétrico e o não-paramétrico. O enfoque **paramétrico** assume que os dados seguem uma certa forma funcional e depende de um ou mais parâmetros da população. Já o enfoque **não-paramétrico** permite esboçar conclusões por intermédio das observações sem formular suposições sobre a forma matemática da distribuição da população. Assim sendo, a forma geral da função de distribuição é derivada dos próprios dados, fornecendo uma maior flexibilidade (CHOU, 1969, pp. 439-440; MARKOU; SINGH, 2003, pp. 2483, 2491) e permitindo que a complexidade da conjectura cresça com o tamanho da amostra. Isto é, quanto maior for o número de dados (observações e características), mais complexa esta poderá ser (RUSSELL; NORVIG, 2004, p. 710).

Na abordagem paramétrica, assume-se que as distribuições dos dados são gaussianas (normais) por natureza e podem ser modeladas estatisticamente com base nas médias e na covariância dos dados (MARKOU; SINGH, 2003, p. 2483). A **distribuição gaussiana**, ou **distribuição normal**, é um dos modelos mais importantes de distribuição contínua. Através de estudos, constatou-se que as discrepâncias entre repetidas medidas da mesma grandeza física apresentavam um grau surpreendente de regularidade, sendo que a distribuição das discrepâncias podia ser satisfatoriamente aproximada por uma curva contínua, conhecida como “curva normal dos erros” e atribuída às leis do acaso. Nas distribuições contínuas, as probabilidades são representadas por áreas sob curvas contínuas cujos gráficos representam funções chamadas **densidades de probabilidade**. Essas funções permitem determinar e manusear probabilidades relacionadas a variáveis aleatórias contínuas. Na prática, essas variáveis são associadas a intervalos ou regiões e não a pontos individuais como se dá nas funções de probabilidades que determinam a distribuição de probabilidade de uma variável aleatória discreta (FONSECA; MARTINS, 1982, pp. 25-30; FREUND; SIMON, 2000, pp. 135-150, 163-165). A Figura 3.2 ilustra uma distribuição normal da variável x que é caracterizada pela forma de um sino com as seguintes propriedades (FONSECA; MARTINS, 1982, pp. 61-62):

- (i) $f(x)$ possui o ponto máximo se $x = \mu$ (média aritmética);
- (ii) $f(x)$ é simétrica em relação à origem $x = \mu$;
- (iii) $f(x)$ tende a zero se x tende a $\pm\infty$. As extremidades se prolongam indefinidamente cada vez mais próximas do eixo horizontal, mas sem jamais tocá-lo, contudo, quase nunca é preciso estendê-las além de quatro ou cinco desvios-padrão (σ) a contar da média, pois é desprezível para quase todos os fins práticos (FREUND; SIMON, 2000, pp. 164-165); e
- (iv) $f(x)$ possui dois pontos de inflexão.

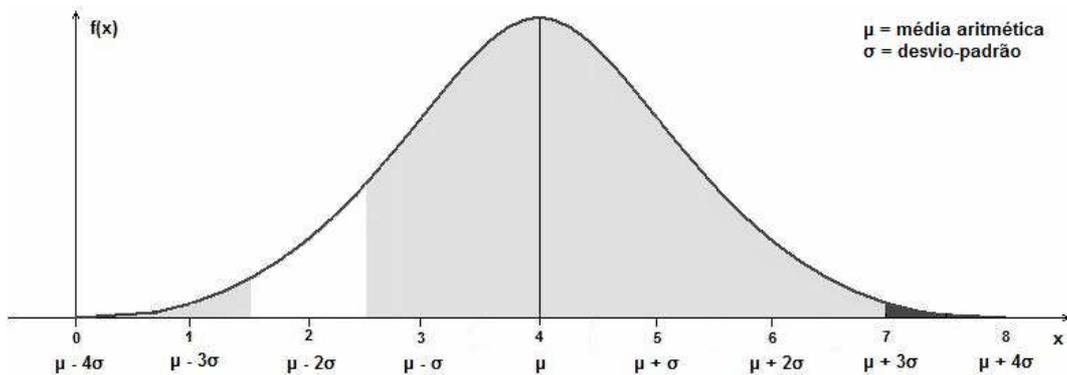


Figura 3.2 - Exemplo da curva de uma distribuição normal.

O diagrama da Figura 3.2 refere-se a uma variável aleatória contínua x que pode assumir qualquer valor no intervalo de 0 a 8. A probabilidade de a variável adotar um valor no intervalo de 1,5 a 2,5 é dada pela região branca sob a curva, e a probabilidade de ela assumir um valor não inferior a 7 é dada pela área em cinza escuro. Dessa maneira, a área sob a curva entre dois valores quaisquer a e b fornece a probabilidade de uma variável aleatória tomar um valor no intervalo de a a b . Quanto mais distantes os valores a e b estiverem do centro (μ), menor será a probabilidade de ocorrerem (FREUND; SIMON, 2000, pp. 61, 163-165; WITTEN; FRANK, 2005, pp. 262-268). A Equação 2.1 é a equação matemática da curva normal que representa a função de densidade de probabilidade de uma variável aleatória x (FREUND; SIMON, 2000, pp. 163-165).

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left\{\frac{x-\mu}{\sigma}\right\}^2} \quad (\text{Equação 2.1})$$

para $-\infty < x < \infty$, onde e é o número irracional 2,71828...

Conforme pode ser observado na Equação 2.1, a distribuição normal depende de dois parâmetros, μ e σ , que são, respectivamente, a média aritmética e o desvio-padrão; assim, há uma e somente uma distribuição normal com uma dada média aritmética μ e um determinado desvio-padrão σ . De forma intuitiva, a média de uma distribuição contínua é uma medida do seu centro, e o desvio-padrão é uma medida de variação que determina a sua dispersão, sendo que, quanto menor a dispersão, mais concentrados os dados estarão em torno da média. Para as situações em que existem múltiplos atributos numéricos por observação e os atributos têm correlação, uma matriz de covariância é utilizada em substituição ao desvio-padrão

empregado para cada atributo (FREUND; SIMON, 2000, pp. 163-165; WITTEN; FRANK, 2005, pp. 266-268).

Um **modelo de misturas gaussianas** (*Gaussian mixture model*) envolve a mescla de distribuições normais que podem representar k grupos para k distribuições normais, permitindo que cada distribuição proporcione a probabilidade de uma observação ser membro desse grupo. Conforme pode ser percebido na Figura 3.3, a função de densidade de probabilidade aparenta uma cadeia de montanhas, com um pico para cada componente, cujas médias e desvios-padrão são distintos, e a probabilidade da distribuição normal A (P_A) mais a probabilidade da distribuição normal B (P_B) é igual a 1. A situação de uma mistura mais simples acontece quando há somente um atributo numérico por observação que tem uma distribuição normal para cada grupo, mas com parâmetros diferentes (WITTEN; FRANK, 2005, pp. 262-263).

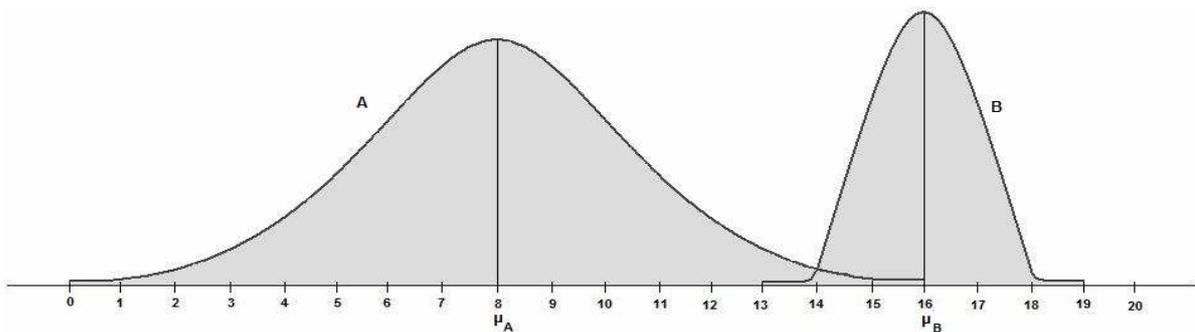


Figura 3.3 - Exemplo de uma mistura envolvendo duas distribuições normais.

A abordagem gaussiana possibilita que um modelo de conceito normal seja representado por médias e desvios-padrão de distribuições gaussianas que descrevem amostras aprendidas ao longo do tempo referentes a uma determinada população. Cada distribuição gaussiana contida no modelo de conceito normal pode ser percebida como um conceito primitivo pertinente a esse modelo. Assim sendo, uma maneira de se identificar *outliers* é verificar se uma nova observação dessa população está próxima a um dos conceitos primitivos que o modelo encapsula. Essa proximidade pode ser fundamentada em x desvios-padrão. Por exemplo, se um modelo encapsula dois conceitos primitivos, sendo o primeiro, com média igual a 10 e desvio-padrão igual a 2, e o segundo, com média igual a 20 e desvio-padrão igual a 1, e o limiar para se identificar um *outlier* for igual a 1 desvio-padrão, então uma nova observação que não estiver entre 8 e 12 e nem entre 19 e 21 será qualificada como *outlier*.

Na inferência estatística, a regra do vizinho mais próximo, a regra dos k -vizinhos mais próximos e a estimação de densidade da janela de Parzen são alguns exemplos de abordagens não-paramétricas.

Na próxima seção, são expostos alguns trabalhos cujo foco é a detecção de observações que não se enquadram na distribuição conhecida dos dados e que foram concebidos examinando as RSSFs.

3.3 Aplicações de Detecção de *Outliers* e Novidades em Redes de Sensores Sem Fio

Muitos trabalhos referentes à detecção de observações que não condizem com a distribuição conhecida dos dados podem ser encontrados (MARKOU; SINGH, 2003). Apesar disso, poucos envolvem aplicações no contexto de RSSFs, levando em conta as suas peculiaridades. Alguns desses trabalhos são mencionados nesta seção.

Mukherjee e Sen (2007) apresentam uma estratégia para detectar nós-sensores defeituosos ou maliciosos em uma RSSF, confrontando os dados coletados com os dados previstos conforme conceitos previamente aprendidos. Os autores admitem que os nós-sensores e a RSSF funcionarão sem erros por um período inicial de tempo, possibilitando a coleta de dados que serão utilizados para o aprendizado. Esse aprendizado ocorre através de uma rede neural artificial de forma centralizada; logo, é isento das limitações computacionais das RSSFs. Posteriormente, os conceitos aprendidos são utilizados pelos nós-sensores para predizerem os seus dados sensoriados. No experimento realizado pelos autores, a RSSF é do tipo hierárquica, sendo os nós-folha responsáveis pelo sensoriamento e os demais, pelo agrupamento dos dados e pela gerência da reputação dos seus nós-filho. A diferença entre o dado sensoriado e a respectiva predição no nó-folha é utilizada como uma medida de erro, possibilitando a qualificação do nó-sensor como defeituoso ou malicioso pelo seu nó-pai quando a reputação do nó-filho for menor do que um limiar. A reputação é avaliada de forma incremental pelo nó-pai, obtendo-se uma seqüência de erros entre os valores preditos e os

valores reportados pelo nó-filho, permitindo que as tomadas de decisões sobre as reputações sejam fundamentadas em múltiplas observações.

Palpanas *et al.* (2003) propõem um arcabouço para a detecção de desvios em fluxos de dados e discutem a sua aplicação nas RSSFs. O objetivo é reportar os desvios identificados e processar a maior quantidade possível de dados de modo descentralizado, evitando esforços de comunicação e computação desnecessários. A arquitetura apresentada considera a existência de nós-sensores mais sofisticados e poderosos para a detecção de desvios. Através de uma perspectiva mais global, todos os nós-sensores de uma área particular têm seus modelos de distribuição de dados combinados e armazenados nesses nós-sensores de maior capacidade, compondo um novo modelo. Os nós-sensores mais robustos também possuem uma banda larga de comunicação específica entre si de tal modo a não interferir na comunicação dos nós-sensores mais simples, sendo que estes últimos são implantados em um número bem maior e podem ser usados para identificar desvios locais. Diferentes partes da área coberta são designadas a diferentes grupos e, conseqüentemente, a diferentes nós-sensores de alta capacidade, os quais também são organizados em grupos e se reportam à estação-base. A detecção de um desvio é realizada da seguinte maneira: dado um valor sensoriado (v_0) e uma função de distribuição de densidade ($f(x)$), estima-se o número de valores que estão na vizinhança de v_0 ; em especial, checa-se o número de valores que se enquadram em uma esfera de raio r em torno de v_0 ($N(v_0, r)$) e, se o número for menor do que um limiar p da aplicação, v_0 será marcado como um desvio. Cada nó-sensor mantém um modelo de distribuição de dados dos seus valores sensorizados, fundamentado nos dados obtidos em uma janela de execução de tamanho N .

Branch *et al.* (2006) expõem uma forma distribuída para se detectar *outliers* em RSSFs por meio de métodos não-paramétricos e não-supervisionados, divulgando os resultados aos nós-sensores da vizinhança. O algoritmo encontra um número máximo de *outliers*, conforme parâmetro, sobre um conjunto de dados global com respeito à vizinhança. O algoritmo, que é executado em cada nó-sensor (p_i), tem como base o armazenamento dos conjuntos dos pontos de dados recebidos ($S_{j,i}$) dos nós-sensores que são vizinhos imediatos ($p_j \in N_i$) e dos conjuntos dos pontos de dados enviados a cada nó-vizinho ($S_{i,j}$). Apenas os dados significativos para a identificação dos *outliers* sob essa perspectiva são trocados entre os nós-sensores. O algoritmo

é baseado em evento e emprega a mesma lógica tanto na iniciação quanto na ocorrência de mudança do conjunto de dados global, seja devido ao recebimento de uma mensagem de um nó-sensor vizinho ou mesmo por conta da modificação do conjunto de dados locais. A proposta também considera que a adição de nós-sensores durante a operação do sistema é factível. No entanto, a contribuição de um nó-sensor removido somente é anulada com o passar do tempo. Experimentos foram conduzidos, através da simulação de uma RSSF com medições reais, comparando-se quatro versões do algoritmo. Essas versões utilizam as abordagens de inferência estatística não-paramétrica 1-NN e k -NN para detectar *outliers*, tanto de forma centralizada quanto descentralizada.

Outra proposta é feita por Pless *et al.* (2004) e Reznik *et al.* (2005). A técnica utiliza um modelo de rede neural artificial denominado *Modified Time-Based Multi-Layer Perceptron* (MTBMLP). Um limiar é configurado e especifica a diferença mínima entre a predição e o valor atual. Se a diferença for maior do que o limiar especificado, a saída do sensor é considerada um *outlier*. Em Pless *et al.* (2004), é mencionado que a rede neural artificial é adaptada gradualmente ao sensor quando a mudança no comportamento do mesmo deixa de ser pontual. A aplicação desenvolvida para avaliar a proposta simula sensores através de funções matemáticas, provê saídas desses sensores a uma rede MTBMLP e, então, compara as saídas dessa rede com as medidas obtidas dos sensores.

Ma *et al.* (2004) expõem uma proposta que permite identificar valores que desviam significativamente dos valores considerados normais, possibilitando a identificação de dados anormais, eventos interessantes ou sensores defeituosos, os quais são reportados à estação-base. A detecção de irregularidades relativas a um atributo examina as características temporais e espaciais de um nó-sensor e revela qualquer desvio em comparação com os dados prévios do próprio nó-sensor ou com os dados dos nós-sensores da sua circunvizinhança. Para a irregularidade temporal dos dados no nó-sensor, é efetuada a aproximação da distribuição dos dados em vez de se manter todo o seu histórico. Assim, são considerados apenas os N valores mais recentes do nó-sensor em uma janela de tempo. O modelo de conceito normal é construído conforme a chegada das coletas do sensor e, quando algumas coletas afetam substancialmente os coeficientes do modelo, a irregularidade é identificada. Em relação às irregularidades espaciais, é exposto um modelo estatístico sobre as coletas dos nós-sensores

adjacentes e, se algumas leituras de um nó-sensor forem distintas daquelas previstas pelo modelo, baseadas nas leituras dos nós-sensores das suas imediações, uma irregularidade será detectada. Caso um nó-sensor seja movido, os parâmetros do seu modelo serão ajustados incrementalmente. Nesse trabalho, não foram efetuados experimentos.

Xiao *et al.* (2007) apresentam um estudo sobre o problema de se determinar falhas na leitura de sensores sem comprometer a detecção de eventos relevantes. Para tal, uma rede lógica de correlações entre nós-sensores é construída no topo da rede, apoiada na similaridade entre as leituras de pares de nós-sensores. Posterior à criação dessa rede lógica e por meio da exploração de uma cadeia de Markov, um mecanismo (*SensorRank*) é desenvolvido. Esse mecanismo permite julgar os nós-sensores, em termos de correlação, por intermédio de um algoritmo de votação (*TrustVoting*), que possibilita determinar falhas nas coletas nos próprios nós-sensores. O *TrustVoting* considera os nós-sensores cujos dados sensorizados estão correlacionados ao nó-sensor em voto. Cada nó-sensor está associado a um valor de confiança (peso), o seu *SensorRank*, que será utilizado durante a votação pelo *TrustVoting* e, quanto menor o valor do *SensorRank* de um nó participante da votação, menor será o seu impacto sobre o resultado final da votação, ou seja, sobre o aceite ou a rejeição do dado sensorizado por um nó-sensor suspeito. O *SensorRank* é um fator obtido conforme o grau de correlação do nó-sensor com os seus vizinhos. Ao passo que o número de nós-sensores correlacionados ao nó-sensor se amplia, o valor do *SensorRank* também aumenta. Implicitamente, o *SensorRank* representa o número de nós-sensores próximos que são similares ao nó-sensor em avaliação. Os autores argumentam a importância do uso da correlação entre os dados sensorizados, além da própria proximidade física entre os nós-sensores, em virtude de ser possível a existência de leituras distintas a despeito dessa proximidade. Estudos práticos foram conduzidos através de simulações.

Zhuang e Chen (2006) propõem uma abordagem para corrigir ou mesmo remover *outliers* em séries sensorizadas, de modo descentralizado, até dois *hops* (saltos) da sua transmissão. Para a correção de *outliers*, cada nó-sensor tenta retificar a série de dados sensorizados substituindo os *outliers* identificados por um valor real aproximado. Para isso, é empregada uma abordagem baseada em *wavelets* cuja função capacita a transformação de outras funções para o domínio da frequência e do tempo. Nesse trabalho, a série temporal

original é transformada em uma escala de frequência e tempo, a qual é suavizada enquanto mantém uma aproximação equivalente à série temporal original. A correção somente acontece quando o *outlier* ocorre súbita, pontual e dissimilarmente do restante da série sensoriada e é feita por meio do confronto entre a série temporal original e a *wavelet* suavizada. Se a diferença entre o valor original e o valor restaurado for maior que um limiar predefinido pelo usuário, o valor será marcado como um *outlier* e substituído pelo valor restaurado contido na *wavelet*. A remoção de *outliers* somente é empregada nas séries onde há uma seqüência de *outliers*, formando um segmento, e ocorre quando a série contém uma elevada quantidade de erros se comparada às séries dos nós-sensores da redondeza. A verificação da dissimilaridade é executada ao longo do percurso até a estação-base. O nó-sensor recebe as séries sensoriadas dos nós-filho e decide o que será encaminhado para a estação-base e o que será eliminado. Neste caso, o nó-sensor reconstrói as séries sensoriadas pelos nós-filho e computa a similaridade tanto dos nós-filho quanto dele mesmo. Se a série desse nó-sensor verificador for dissimilar de todas as séries dos seus nós-filho, então ela será classificada como desconhecida, uma vez que outros nós-sensores na sua vizinhança precisam ser comparados antes de considerá-la um *outlier*, o que será tratado pelo seu nó-pai (próximo salto) mantendo-se a mesma idéia. Porém, o nó-pai realizará a comparação dessa série rotulada como desconhecida primeiramente com seus nós-filho e, posteriormente, com ele mesmo, removendo-a do fluxo caso a dissimilaridade seja confirmada. Experimentos de validação foram realizados de modo simulado.

Elnahrawy e Nath (2003) introduzem uma proposta para limpar ruídos identificados nos nós-sensores. A remoção dos ruídos tem como objetivo reduzir a incerteza associada às leituras de cada sensor que surge devido a divergências casuais. A proposta baseia-se na abordagem bayesiana e pode ser empregada tanto na estação-base quanto nos nós-sensores. As consultas são submetidas com um nível de confiança especificado pelo usuário, tal como: “retorne o valor máximo dos sensores que possuem ao menos 80% de chance de sua temperatura ser maior ou igual a 30°C”.

Conforme mencionado na Seção 2.5, Deshpande *et al.* (2005) apresentam um modelo probabilístico que permite ao usuário informar limites de tolerância de erros e de confiança. Esse modelo, cujo aprendizado ocorre de forma centralizada, é usado para estimar as leituras

dos sensores no período de tempo corrente, o que formará a resposta à consulta. Nesse trabalho, é introduzido um software local em cada nó-sensor que, em geral, facilita a execução da consulta e remove alguns *outliers* com base na versão de um modelo local.

Um arcabouço para computar uma aproximação de distribuições de dados multidimensionais em RSSFs de modo distribuído foi proposto em Subramanian *et al.* (2006). Nesse trabalho, é relatada uma demonstração de como esse arcabouço pode ser estendido para se detectar *outliers*. O foco de interesse foi a identificação de *outliers* em uma janela de tempo, que mantém os últimos dados sensoriados, considerando as leituras oriundas de múltiplos nós-sensores. Experimentos para validação da abordagem foram conduzidos por meio de um simulador.

Além da aplicação de técnicas de detecção de *outliers* e novidades para o tratamento dos dados sensoriados permitindo disponibilizá-los qualitativamente às aplicações, há publicações cujo enfoque está na própria infra-estrutura computacional da RSSF. Essas aplicações podem ser percebidas, por exemplo, em Song *et al.* (2005), que empregam tais técnicas para fornecer maior segurança à rede.

3.4 Considerações Finais

Este capítulo foi destinado às definições formais sobre o tema detecção de *outliers* e novidades, destacando-se outros conceitos e termos inter-relacionados. De modo complementar, uma exposição relativa ao paradigma de aprendizado de máquina estatístico foi efetuada. Por fim, ainda foram destacados alguns trabalhos envolvendo técnicas de detecção de *outliers* e novidades no contexto das RSSFs.

O capítulo subsequente introduz a nova abordagem para adaptar dinamicamente o processamento de consultas em RSSFs baseada em noções de qualidade de serviço da consulta e em técnicas de detecção de novidades.

4 PROCESSAMENTO DE CONSULTAS EM RSSFs COM CLASSES DE QUALIDADE DE SERVIÇO

Após a exposição dos conceitos fundamentais concernentes às RSSFs e à detecção de novidades nos Capítulos 2 e 3, respectivamente, neste capítulo, apresenta-se a nova abordagem para o processamento adaptativo de consultas em RSSFs, baseada em noções de qualidade de serviço da consulta e em técnicas de detecção de novidades. A proposta tem como objetivo diminuir o número de mensagens injetadas na rede a fim de permitir uma extensão da sua vida útil, enquanto tenta fornecer os resultados das consultas com o nível de qualidade esperado pelos seus usuários.

A estrutura deste capítulo é descrita a seguir. A Seção 4.1 descreve a abordagem de processamento de consultas em RSSFs com base em classes de qualidade de serviço. Ademais, as quatro classes de qualidade de serviço definidas neste trabalho são descritas com suas respectivas políticas. Na Seção 4.2, um algoritmo baseado em detecção de novidades, denominado *AdaQuali* (*ADaptive QUALity control for query processing in wireless sensor networks*), é introduzido para controlar os requisitos demandados pelos usuários e as atividades dos nós-sensores, através da adaptação das frequências tanto das coletas quanto das respectivas transmissões. Já o pseudocódigo da abordagem é apresentado na Seção 4.3. Na Seção 4.4, é efetuada uma comparação da abordagem introduzida neste capítulo em relação a trabalhos relacionados que foram expostos nas Seções 2.5 e 3.3. Por fim, as considerações finais deste capítulo são mencionadas na Seção 4.5.

4.1 Processamento de Consultas em RSSFs via Classes de Qualidade de Serviço de Consultas

Conforme exposto na Seção 2.5, as características das RSSFs ditam a necessidade para que a funcionalidade da rede e os nós-sensores sejam controlados para melhor servirem aos requisitos da aplicação, e tais requisitos, como acurácia e latência dos dados, podem diferir fortemente entre as aplicações (GARCÍA-HERNÁNDEZ *et al.*, 2007, p. 269; DELICATO *et al.*, 2005). Tal fato incita que as consultas processadas nessas redes devem ir ao encontro das exigências dos usuários. Na Seção 2.5, também se destacou que o conceito de qualidade de serviço tem assumido significados distintos em diferentes cenários, desde os objetivos até os subjetivos. Destacou-se ainda que o conceito de qualidade de serviço para esta dissertação diz respeito exclusivamente aos serviços de processamento de consultas providos por uma mesma RSSF, os quais podem ser customizados para atender mais efetivamente a diferentes necessidades requeridas por diferentes tipos de aplicação, em particular, à qualidade dos dados e ao comportamento temporal dos dados processados.

Para o propósito de tratar diferentes tipos de qualidade de serviço de consultas, quatro classes genéricas de qualidade de serviço foram concebidas. Cada classe de qualidade representa um tipo de consulta a ser executada, a qual poderá conceder uma maior flexibilidade aos usuários e ser indicada de acordo com as características e criticidades das aplicações. Outrossim, cada classe de qualidade conduz a um comportamento diferenciado no processamento distribuído da consulta no nó-sensor, o que permite que algumas propriedades de execução da consulta sejam controladas pelos usuários e aplicações, e é empregada de acordo com as características dos dados sensorizados e das disponibilidades de memória e energia. Além disso, as classes de qualidade estabelecidas possuem políticas de qualidade distintas que indicam, por meio de procedimentos e critérios de qualidade firmados por indicadores, como as consultas pertencentes a essas classes deverão ser servidas pelos nós-sensores. A qualidade dos dados e o comportamento temporal das coletas e respectivas remessas à estação-base são os principais ingredientes de qualidade de serviço a serem requeridos pelo usuário. A esse respeito, recorreu-se à teoria de detecção de novidades (MARKOU; SINGH, 2003; SPINOSA *et al.*, 2007) para caracterizar as políticas de qualidade e conduzir as operações de processamento de consultas relacionadas a cada classe.

A seguir, as políticas predefinidas para cada classe de qualidade de serviço serão descritas.

1. *Without Outliers (sem outliers)*:

As coletas que desviam significativamente do modelo de conceito normal dos dados monitorados são identificadas e removidas do resultado da consulta em cada nó-sensor antes de serem transmitidas a outros nós. Tal propriedade pode preservar a energia residual da rede, caso algum *outlier* seja detectado durante o processamento da consulta. Os *outliers* podem representar, por exemplo, uma falha na leitura do nó-sensor ou uma interferência ambiental, logo, podem ser desnecessários aos usuários em determinados contextos, além de poderem afetar consideravelmente os resultados das consultas como nas agregações de dados (e.g., mínimo e máximo). Esta classe respeita a latência dos dados a serem entregues e o volume de dados a serem coletados durante o processamento da consulta no nó-sensor, pois não adapta os intervalos de tempo entre coletas e entre envios de dados à estação-base estabelecidos para a consulta. Assim, esta classe pode ser empregada em aplicações para as quais os tempos de resposta e de coleta predeterminados na consulta são críticos. Por exemplo, monitorar a temperatura de um paciente em um hospital em determinados intervalos de tempo. Nesse caso, tem-se em mente que a precisão temporal da coleta e o volume de dados a serem coletados são mais relevantes do que a preservação da energia residual da rede. Mesmo que a temperatura do paciente aumente, essa alteração comportamental poderá ser identificada, pois tende a manter um crescimento contínuo e não uma ocorrência pontual. Já possíveis falhas nas leituras dos nós-sensores ou interferências poderiam ser identificadas como *outliers*. Por outro lado, se a remoção de qualquer *outlier* não for interessante para o usuário, esta classe não poderá ser empregada, tampouco as classes *Fine-grained Accuracy* e *Coarse-grained Accuracy*. Por exemplo, se o usuário deseja medir o nível de ruído sonoro em um ambiente por um determinado período de tempo, a ocorrência de um *outlier* seria importante, então este não poderia ser removido do resultado da consulta.

2. *Fine-grained Accuracy* (acurácia com granulosidade fina):

Assim como na classe *Without Outliers*, as coletas que desviam significativamente do conceito normal são descartadas. Ademais, esta classe permite um acréscimo moderado do intervalo de tempo entre coletas previsto inicialmente para a consulta conforme a disponibilidade de memória do nó-sensor diminui. O intervalo de tempo entre remessas parciais dos resultados da consulta à rede também pode aumentar, mas isso é gerenciado por meio da disponibilidade de energia do nó-sensor. Dessa forma, a energia da rede poderá ser preservada por meio da redução tanto do uso dos recursos de comunicação quanto da quantidade de coletas realizadas. Essas duas formas de adaptação ocorrem conforme em Brayner *et al.* (2006, 2008). Em razão de adequar esses intervalos de tempo estabelecidos na consulta, esta classe pode ser usada por aplicações para as quais o tempo de resposta e o volume previsto de dados a serem coletados podem ser ajustados (aproximados), tais como: monitoração da temperatura ambiente de salas comerciais, mapeamento da bio-complexidade do ambiente, monitoração do comportamento de animais e estudos sobre poluição. Por exemplo, monitorar a qualidade das águas de um determinado rio ao longo do ano. Neste caso, a precisão temporal da coleta e o volume de dados a serem coletados poderiam ser ajustados sem prejudicar a análise final dos dados monitorados. Para aplicações que podem fazer uso desta classe, apesar do fornecimento de dados aproximados, tem-se em conta que a informação fornecida mantém sua relevância. Isso também se aplica para a classe *Coarse-grained Accuracy*.

3. *Coarse-grained Accuracy* (acurácia com granulosidade grossa):

Esta classe implica um comportamento dos nós-sensores similar à classe *Fine-grained Accuracy*. A principal diferença é que ela permite um incremento mais arrojado no espaçamento entre coletas através de um balanceamento dinâmico entre os recursos disponíveis de memória e a aderência dos dados coletados ao modelo de conceito normal. Quanto maior for a aderência, mais intensa será essa adaptação. Esta classe ainda permite que um valor aproximado seja remetido em substituição aos dados coletados nas situações em que o comportamento dos dados não tenha mudado. Neste

caso, tem-se em conta que os dados coletados estão em conformidade com o que já é conhecido (isto é, condizentes com o modelo de conceito normal) a respeito do fenômeno monitorado e, portanto, já esperado pelos usuários. Por meio desta classe, a energia residual disponível nos nós-sensores poderá ser fortemente preservada.

4. *Outlier-Aware* (ciente de outliers):

Da mesma forma que a classe *Without Outliers*, esta classe não modifica os intervalos de tempo entre coletas e nem entre as transmissões parciais de dados predeterminados na consulta. Contrariamente à classe *Without Outliers*, esta classe estipula que as coletas aderentes ao modelo de conceito normal devem ser descartadas, permitindo que somente os *outliers* sejam realmente reportados a outros nós. A classe *Outlier-Aware* permite que as monitorações sejam utilizadas com o intuito de minimizar riscos; por exemplo, financeiros, identificando defeitos em equipamentos o mais breve possível e evitando maiores prejuízos. A partir do conhecimento do comportamento normal desses equipamentos, consultas periódicas poderiam ser submetidas para monitorá-los e alguma ação poderia ser disparada no caso em que *outliers* fossem identificados. Outra aplicação seria a obtenção de valores de temperaturas de ambientes que não se enquadram nos seus níveis considerados normais ou mesmo a identificação de tremores de terra atípicos nas regiões monitoradas. Esse comportamento pode trazer ganhos significativos de conservação de energia na rede uma vez que os *outliers* são usualmente muito menos freqüentes do que os dados associados ao conceito normal global.

Ao passo que a classe de qualidade *Outlier-Aware* somente demanda o reconhecimento apropriado de *outliers*, as demais classes requerem que um esquema eficiente de detecção de novidade ou mudança de conceito seja aplicado. Enquanto uma classificação de qualquer nova observação em normal ou *outlier* é sempre realizada para todas as classes, somente as três primeiras classes estão aptas a verificar se os *outliers* podem estar relacionados a uma novidade ou a uma mudança de conceito. Essas distinções permitem separar discrepâncias pontuais (e.g., uma falha na leitura do sensor) de uma mudança comportamental dos dados sensorizados pelos nós-sensores (e.g., aumento da temperatura em uma área geográfica).

Para a implementação das funcionalidades das quatro classes de qualidade concebidas, assume-se que o conceito normal segue a distribuição gaussiana (ver Seção 3.2) e, portanto, os *outliers*, as novidades e as mudanças de conceito podem ser identificados fazendo-se uso de determinadas estatísticas relativas ao fenômeno observado, como desvio-padrão e média. Além de a distribuição gaussiana ser simples, abordagens estatísticas paramétricas como essa têm sido aplicadas com sucesso em uma variedade de cenários envolvendo técnicas de detecção de novidades, como pode ser apreciado em Markou e Singh (2003).

Após o detalhamento das características concernentes ao processamento de consultas com classes de qualidade de serviço, dois quadros comparativos entre as classes de qualidade de serviço da consulta são expostos. O primeiro mostra quais ações efetuadas por cada classe de qualidade poderão resguardar a energia residual da rede, possibilitando uma sobrevida à mesma (Quadro 4.1), enquanto o segundo quadro (Quadro 4.2) compara algumas características entre as classes de qualidade.

Classe de qualidade de serviço da consulta	Ações que possibilitam a conservação da energia residual da rede
<i>Without Outliers</i>	<ul style="list-style-type: none"> • Suprimir <i>outliers</i> identificados no próprio nó-sensor, o que evita o seu tráfego pela rede.
<i>Fine-grained Accuracy</i>	<ul style="list-style-type: none"> • Suprimir <i>outliers</i>. Similar à classe anterior; • Reduzir a quantidade de dados coletados devido ao incremento do espaçamento entre coletas. Isso permite preservar a energia residual da rede pela diminuição do volume de dados disseminados na rede e pela redução do custo total das coletas no próprio nó-sensor; e • Aumentar o intervalo entre os envios de dados à estação-base, o que possibilita reduzir o número de mensagens transmitidas na rede ao mesmo tempo em que também conserva a energia do nó-sensor que tenciona enviar a mensagem.
<i>Coarse-grained Accuracy</i>	Esta classe realiza as mesmas ações da classe <i>Fine-grained Accuracy</i> , porém a redução da quantidade de dados sensorizados poderá ser mais intensa conforme a aderência das coletas realizadas ao modelo de conceito normal. Além disso, adiciona o fato de substituir as coletas contidas em uma amostra ^a por um valor aproximado (média das coletas consideradas normais) se nenhuma mudança no comportamento dos dados for identificada, permitindo assim, uma maior redução de dados transmitidos pela rede.
<i>Outlier-Aware</i>	<ul style="list-style-type: none"> • Suprimir coletas consideradas normais no próprio nó-sensor, o que permite poupar a energia através do envio exclusivo de <i>outliers</i> pela rede, que tendem a ser menos frequentes do que os dados associados ao modelo de conceito normal.

^a Conjunto de dados obtidos conforme cláusulas *SENSE INTERVAL* e *SEND INTERVAL*.

Quadro 4.1 - Ações que possibilitam a conservação de energia global da rede.

Característica	Classe de qualidade de serviço da consulta			
	<i>Without Outliers</i>	<i>Fine-grained Accuracy</i>	<i>Coarse-grained Accuracy</i>	<i>Outlier-Aware</i>
Pode efetuar a aprendizagem da amostra ^a aplicando um esquema de distinção entre novidade e mudança de conceito.	Sim	Sim	Sim	Não
Adapta as cláusulas <i>SENSE INTERVAL</i> e <i>SEND INTERVAL</i> com base na disponibilidade de memória e energia do nó-sensor, respectivamente.	Não	Sim	Sim	Não
Considera a aderência dos dados coletados ao modelo de conceito normal para adaptação da cláusula <i>SENSE INTERVAL</i> .	Não se aplica	Não	Sim	Não se aplica
Envia exclusivamente dados normais à estação-base.	Sim	Sim	Sim	Não
Envia restritamente <i>outliers</i> à estação-base.	Não	Não	Não	Sim
Envia valor aproximado (média das observações normais contidas na amostra ^a) se não ocorrer nenhuma mudança comportamental dos dados em relação ao modelo de conceito normal, isto é, se uma novidade ou mudança de conceito não for detectada.	Não	Não	Sim	Não
Adequa-se a aplicações intolerantes a atrasos.	Sim	Não	Não	Sim

^a Conjunto de dados obtidos conforme cláusulas *SENSE INTERVAL* e *SEND INTERVAL*.

Quadro 4.2 - Características gerais das classes de qualidade de serviço da consulta.

4.2 AdaQuali

Para o tratamento de cada classe de qualidade, o algoritmo *AdaQuali* (*ADaptive QUALity control for query processing in wireless sensor networks*) (BRAYNER *et al.*, 2010) foi desenvolvido. *AdaQuali* é um algoritmo adaptativo executado nos nós-sensores, que faz uso de técnicas de aprendizado de máquina para adequar o comportamento dos nós-sensores de acordo com as especificações das diferentes classes de qualidade de serviço. O papel do *AdaQuali* é ajustar as atividades de sensoriamento e transmissão de dados dos nós-sensores dinamicamente mediante balanceamento entre o consumo dos recursos computacionais (memória e energia) necessários para realização dessas atividades e a qualidade dos dados esperada pelos usuários, o que também poderá proporcionar um acréscimo na vida útil da rede.

AdaQuali pode ser executado tanto para consultas contínuas quanto para aquelas iniciadas pelo observador (agendadas ou não) e é considerado um algoritmo ciente de

contexto (ver Seção 2.4). Isso porque faz uso de informações correntes de contexto que dão subsídios para a execução de ações focando, por exemplo, o uso inteligente dos recursos da rede ou mesmo a alteração do modo como a consulta está sendo executada. Ademais, *AdaQuali* também pertence a uma classe de algoritmos concernentes às RSSFs denominados “algoritmos de fidelidade adaptativos” pelos quais a qualidade da resposta pode ser tratada em conformidade com o tempo de vida da bateria, a banda de comunicação ou o número de nós-sensores ativos (ESTRIN *et al.*, 1999, p. 267).

Dentre as propriedades do *AdaQuali*, destacam-se as seguintes:

- P1. processamento descentralizado: o processamento ocorre nos próprios nós-sensores;
- P2. generalização: a abordagem permite que outros modelos de aprendizado de máquina, além da distribuição gaussiana, sejam postos em prática; assim, pode ser generalizada para outros tipos de aplicações e necessidades;
- P3. ciência de contexto: *AdaQuali* faz uso de informações correntes de contexto para a tomada de decisões, em especial, dados, memória e energia disponíveis;
- P4. adaptação: *AdaQuali* possibilita uma adequação ao contexto corrente do nó-sensor;
- P5. orientação a regras: as regras são predefinidas e empregadas segundo a classe de qualidade de serviço especificada para a consulta; e
- P6. aprendizagem não-supervisionada, incremental e *on-line*: o aprendizado ocorre de forma *on-line* nos próprios nós-sensores sem a existência de um supervisor para monitorar os resultados durante a aprendizagem. Além disso, *AdaQuali* possui uma abordagem incremental; portanto, o aprendizado pode ser realizado continuamente, o que possibilita a atualização do modelo de conceito normal dos nós-sensores ao longo do tempo.

Para dar suporte à nova abordagem de processamento de consultas em RSSFs baseada em classes de qualidade de serviço da consulta e em técnicas de detecção de novidades, foi realizada uma extensão da SNQL (*Sensor Network Query Language*) (BRAYNER *et al.*, 2007), apresentada na Seção 2.4, para a introdução da cláusula *QUALITY CLASS*, cuja

definição é mostrada no Quadro 4.3. Essa nova cláusula especifica a classe de qualidade de serviço associada a uma determinada consulta. Um exemplo que diz respeito a essa extensão é exibido na Figura 4.1. Nesse exemplo, a consulta retornará os valores máximo de temperatura e mínimo de umidade coletados na região coberta, sendo que os *outliers* identificados serão previamente descartados nos próprios nós-sensores durante o processamento da consulta.

Cláusula ^a	Definição
[<i>QUALITY CLASS</i> <class>]	<class>: estabelece a classe de qualidade de serviço da consulta a ser empregada durante o respectivo processamento. As possíveis expressões para essa cláusula são: <i>Without Outliers</i> , <i>Fine-grained Accuracy</i> , <i>Coarse-grained Accuracy</i> e <i>Outlier-Aware</i> .

^a O símbolo “[]” denota cláusula opcional e o “<>”, uma expressão.

Quadro 4.3 - Extensão da SNQL (BRAYNER *et al.* 2007), incorporando a cláusula *QUALITY CLASS*.

```

SELECT          r.IdRegiao AS RegiaoGeografica, MAX(t.valorColetado),
                  MIN(u.valorColetado)
FROM           Umidade u, Temperatura t, RegiaoSensoriada r
WHERE          r.IdRegiao = t.IdRegiao AND
                  r.IdRegiao = u.IdRegiao AND
                  r.latitudeInicial > 034308 AND
                  r.latitudeFinal < 034316 AND
                  r.longitudeInicial > 383151 AND
                  r.longitudeFinal < 383114

GROUP BY       r.IdRegiao
TIME WINDOW   3600
SEND INTERVAL 120
SENSE INTERVAL 10
QUALITY CLASS  Without Outliers

```

Figura 4.1 - Exemplo de consulta especificada em SNQL com a cláusula *QUALITY CLASS*.

Uma consulta SNQL é processada em quatro fases: **Interpretação**, **Decomposição**, **Processamento de subconsultas** e **Processamento final** (ver Seção 2.4). A Figura 4.2 exibe uma extensão da representação inicial do processador de consultas para RSSFs proposto em Brayner *et al.* (2007), onde o componente *AdaQuali* foi introduzido. Conforme pode ser observado, esse componente está apto a ser executado nos próprios nós-sensores quando a subconsulta está sendo processada durante a fase **Processamento de subconsultas**. *AdaQuali* manipula amostras de dados, que são conjuntos de observações coletadas do ambiente, as quais são tratadas – classificadas, descartadas e possivelmente aprendidas – de acordo com a classe de qualidade de serviço estabelecida para a consulta. O tamanho de cada amostra é

delimitado pelas cláusulas *SENSE INTERVAL* e *SEND INTERVAL* da SNQL (BRAYNER *et al.*, 2006, 2007, 2008). Assume-se que o tamanho da amostra seja suficientemente grande para um aprendizado efetivo do conceito normal dos dados que descrevem o fenômeno monitorado.

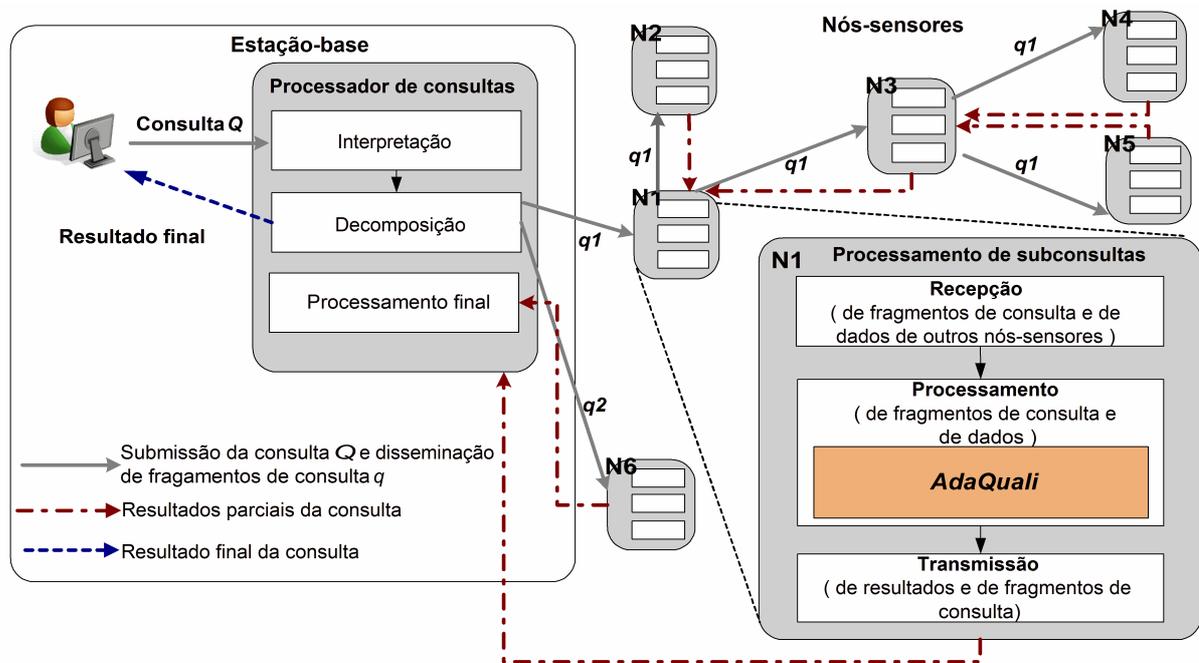


Figura 4.2 - Diagrama esquemático do mecanismo de processamento de consultas em RSSFs com a introdução do *AdaQuali*. Figura adaptada de Brayner *et al.* (2007, p. 2917).

A seguir, são detalhadas as regras específicas aplicadas pelo *AdaQuali* para a classificação e a possível aprendizagem das observações coletadas do ambiente, bem como o modo como o espaçamento entre coletas e entre remessas de dados à estação-base é ajustado.

Regras para Identificação de *Outliers*, Novidades e Mudanças de Conceito

A regra para *AdaQuali* classificar uma coleta do sensor como normal ou *outlier* é confrontar a proximidade desse dado em relação ao modelo de conceito normal. Em virtude do modelo de conceito normal para esta dissertação ser baseado na distribuição gaussiana, esse modelo encapsula conceitos primitivos baseados em médias e desvios-padrão de amostras (gaussianas). Cada conceito primitivo se refere a uma amostra aprendida, ou seja, a um conjunto de observações adquiridas do ambiente em um determinado intervalo de tempo

ΔT , de acordo com a cláusula *SEND INTERVAL*, e em uma determinada taxa corrente de aquisição de leituras, de acordo com a cláusula *SENSE INTERVAL*. Assim, se a coleta estiver próxima a algum conceito primitivo contido no modelo de conceito normal, de acordo com um limiar γ , essa coleta será considerada normal; caso contrário, será temporariamente armazenada para ser reclassificada futuramente, isto é, ser confirmada ou não como um *outlier*. A regra para se distinguir uma coleta normal de um *outlier* está em conformidade com o exemplo exposto na Seção 3.2, p. 49.

Dado que uma mudança comportamental dos dados só é percebida quando um determinado percentual de *outliers* β em relação ao tamanho da amostra estiver na memória temporária, todas as coletas inicialmente identificadas como *outlier* são confirmadas como tal caso o percentual de *outliers* da amostra seja menor do que β . Se esse percentual for maior ou igual a β , *AdaQuali* determinará se a amostra evidencia mais uma novidade ou uma mudança de conceito.

A fim de discriminar se uma amostra evidencia uma novidade ou uma mudança de conceito, *AdaQuali* primeiramente computa a maior distância d_{max} entre o valor médio do modelo de conceito normal (isto é, a média de todos os conceitos primitivos encapsulados no modelo) e a média de qualquer conceito primitivo. Para detectar uma novidade ou uma mudança de conceito, *AdaQuali* avalia quão próximo o novo conceito primitivo adquirido da amostra (isto é, a média) está do valor médio do modelo de conceito normal. Se essa distância exceder d_{max} , a amostra será rotulada como uma novidade, ou seja, um conceito emergente; caso contrário, como uma mudança de conceito. No primeiro caso, o modelo de conceito normal será posteriormente atualizado para também representar a novidade. Já no segundo caso, o conceito primitivo que estiver mais próximo do valor médio da amostra aprendida será atualizado para que a mudança de conceito também seja refletida no modelo de conceito normal. Supondo-se que um modelo de conceito normal encapsula três conceitos primitivos com médias iguais a 10, 14 e 17, e o novo conceito primitivo extraído de uma amostra seja igual a 21; uma novidade ou uma mudança de conceito seria identificada da seguinte forma:

1. Calcula-se o valor médio do modelo de conceito normal $medMod$:
 - a. $medMod = 10 + 14 + 17 / 3 = 13,67$;
2. Obtém-se a maior distância d_{max} entre o valor médio do modelo de conceito normal $medMod$ e a média de qualquer conceito primitivo:
 - a. distância entre o conceito primitivo 10 e $medMod = | 13,67 - 10 | = 3,67$;
 - b. distância entre o conceito primitivo 14 e $medMod = | 13,67 - 14 | = 0,33$;
 - c. distância entre o conceito primitivo 17 e $medMod = | 13,67 - 17 | = 3,33$;
 - d. $d_{max} = 3,67$;
3. Calcula-se a distância d_{conc} entre o valor médio do modelo de conceito normal $medMod$ e o novo conceito primitivo:
 - a. $d_{conc} = | 13,67 - 21 | = 7,33$; e
4. Determina-se se o novo conceito primitivo é uma novidade ou uma mudança de conceito, comparando-se d_{conc} com d_{max} :
 - a. $d_{conc} > d_{max}$, então a amostra será considerada uma novidade. Caso contrário, seria considerada uma mudança de conceito.

Neste trabalho, a qualidade é mensurada por meio de parâmetros, sendo garantida através do percentual de *outliers* β , que indica o início de um novo aprendizado, e por meio do limiar γ , que permite qualificar se uma observação é conhecida. Esses parâmetros dependem das necessidades dos usuários e características das aplicações.

Regras para adaptação das cláusulas *SENSE INTERVAL* e *SEND INTERVAL*

Como dito anteriormente neste capítulo, as adaptações dos espaçamentos entre aquisições de dados e remessas à estação-base somente ocorrem para as classes de qualidade de serviço *Fine-grained Accuracy* e *Coarse-grained Accuracy*. Para isso, faz-se uso das estratégias de adaptação das cláusulas *SEND INTERVAL* e *SENSE INTERVAL* definidas para o algoritmo de agregação ADAGA (BRAYNER *et al.*, 2006, 2008) e representadas respectivamente pelas Figuras 4.3 e 4.4. Essas funções são utilizadas neste trabalho para ajustar as consultas tendo-se em conta a respectiva qualidade de serviço envolvida e as especificidades das aplicações.

A Figura 4.3 apresenta como o valor da cláusula *SEND INTERVAL* é incrementado de acordo com a disponibilidade de energia (v). A função $f(v)$ varia de um valor mínimo (d), que corresponde ao valor da cláusula *SEND INTERVAL*, a um valor máximo (t), que corresponde ao valor da cláusula *TIME WINDOW*. Se a energia disponível estiver perto de 100%, a função $f(v)$ retorna o valor d ; entretanto, conforme a energia diminui, o valor de $f(v)$ incrementa. Já a Figura 4.4 ilustra como o valor da cláusula *SENSE INTERVAL* é incrementado de acordo com a disponibilidade de memória (m). A função $h(m)$ varia de um valor mínimo (g), que corresponde ao valor da cláusula *SENSE INTERVAL*, a um valor máximo (f), que corresponde ao valor da cláusula *SEND INTERVAL* obtido pela aplicação da função $f(v)$. Conforme a disponibilidade de memória reduz, o valor de $h(m)$ aumenta. Se o percentual de memória disponível atingir 0%, o valor da cláusula *SENSE INTERVAL* assumirá o valor obtido através da função $f(v)$, que significa que nenhuma coleta será efetuada até o tempo estabelecido na cláusula *SEND INTERVAL* ($f(v)$) ser alcançado. Como pode ser notado através da Figura 4.4, a função para adaptação da cláusula *SENSE INTERVAL* depende de uma constante denominada α , a qual determina a tendência da curva da função. Em Brayner *et al.* (2006, 2008), os resultados dos experimentos com a referida constante mostraram que, quanto maiores os valores a ela atribuídos, mais rapidamente a cláusula *SENSE INTERVAL*, inicialmente estabelecida para consulta, será adaptada, mesmo com boa margem de memória disponível (90%). Essa constante poderá ser modificada dinamicamente pelo *AdaQuali* durante o processamento da consulta.

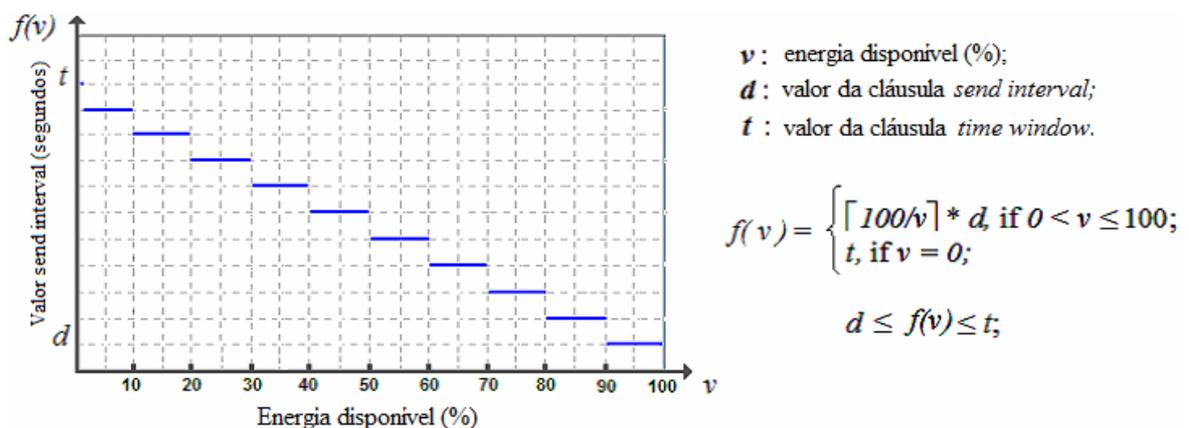


Figura 4.3 - Função para adaptação da cláusula *SEND INTERVAL*. Figura adaptada de Brayner *et al.* (2008, p. 334).

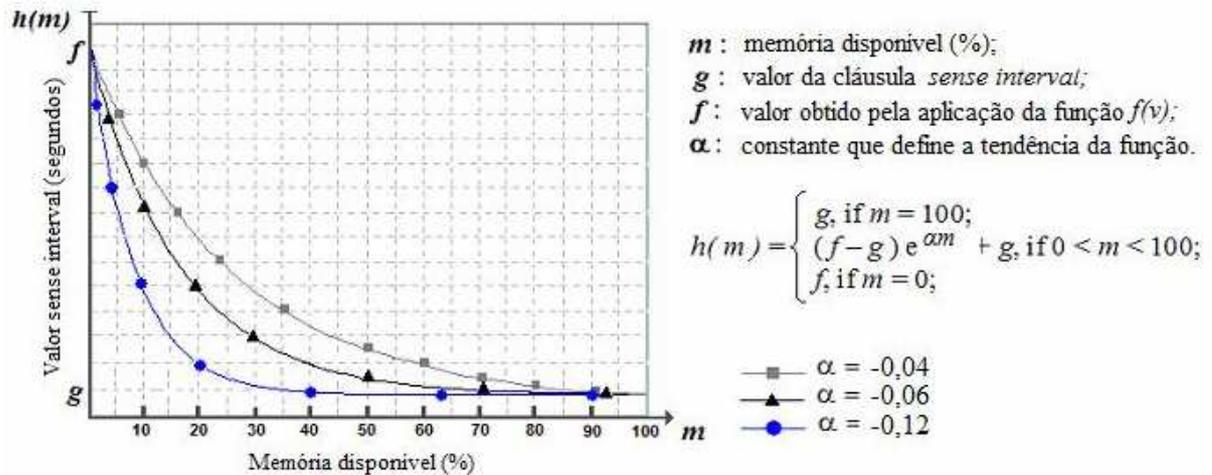


Figura 4.4 - Função para adaptação da cláusula *SENSE INTERVAL*. Figura adaptada de Brayner *et al.* (2008, p. 335).

Para ambas as classes de qualidade de serviço que conduzem a essas adequações entre intervalos de tempo, a adaptação da cláusula *SEND INTERVAL* ocorre da mesma maneira que a proposta original; ou seja, conforme a energia disponível do nó-sensor diminui, o valor da cláusula *SEND INTERVAL* aumenta (Figura 4.3). Com respeito à cláusula *SENSE INTERVAL*, a adaptação é adequada ao tipo de classe de qualidade escolhido. Enquanto a classe de qualidade *Fine-grained Accuracy* mantém uma adaptação mais moderada, fazendo-se uso de α com um valor padrão (-0,06), a classe *Coarse-grained Accuracy* poderá alterar α dinamicamente de forma mais intensa conforme as observações das amostras sejam consideradas normais. Assim sendo, α terá um valor determinado com base na normalidade dos dados; isto é, quanto maior a aderência das coletas ao modelo de conceito normal, maior poderá ser o intervalo entre as coletas de dados em comparação à especificação da cláusula *SENSE INTERVAL* inicialmente informada para a consulta. Desse modo, α poderá variar entre -0,06 e -0,01 e estará mais próximo do valor -0,01 quanto maior for a aderência ao modelo. Para α igual a -0,06 e -0,01, o valor da cláusula *SENSE INTERVAL* ($h(m)$) estará relativamente próximo do valor original (g) quando a memória disponível estiver entre 50% e 100% e entre 90% e 100%, respectivamente. O intervalo para α foi estabelecido com base nos resultados dos experimentos conduzidos em Brayner *et al.* (2006, 2008). O cálculo de α é realizado com base na proporcionalidade da aderência dos dados contidos na amostra ao modelo de conceito normal da seguinte forma:

1. Obtém-se o percentual de coletas consideradas normais na amostra (*perNormal*), e.g., 70%;
2. Calcula-se a diferença entre o maior e o menor valor permitido para α (*difaMin α Max*), sendo que, o maior valor para α (*α Max*) é igual a -0,01 e o menor valor para α (*α Min*) é igual a -0,06; e
3. Computa-se o novo valor de α : $\alpha = \alpha Min + (perNormal \times difaMin\alpha Max / 100)$, o qual é arredondado para duas casas decimais.

Na próxima seção, será detalhado o pseudocódigo do *AdaQuali*.

4.3 Execução de Consultas nos Nós-sensores com Classes de Qualidade de Serviço

O processamento de consultas em RSSFs para execução de consultas com diferentes classes de qualidade de serviço é fundamentado em quatro estágios processados nos próprios nós-sensores. A Figura 4.5 ilustra esses estágios e as suas relações.

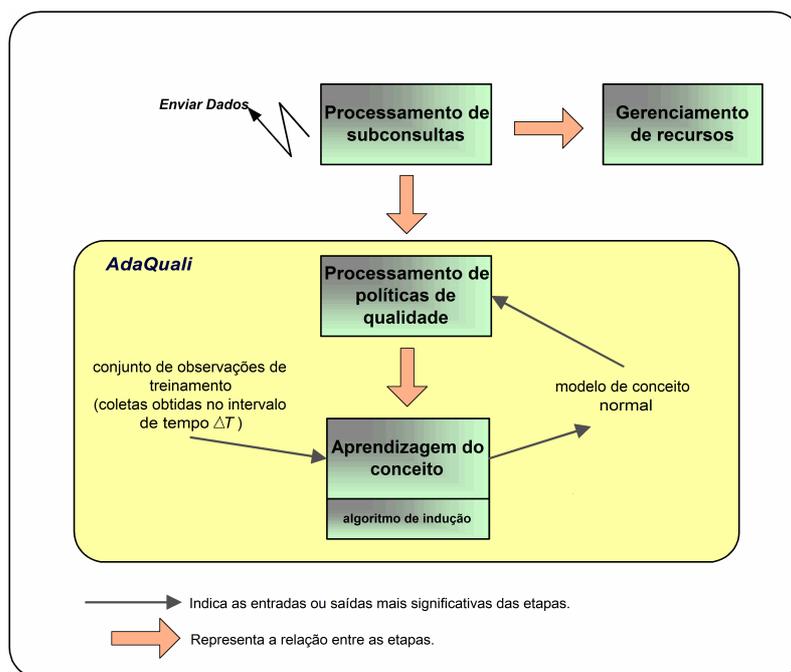


Figura 4.5 - Diagrama esquemático dos estágios do processamento de consultas com classes de qualidade de serviço executado nos nós-sensores.

O estágio **Processamento de subconsultas** é responsável por encaminhar os dados à rede e acionar os estágios **Gerenciamento de recursos** e **Processamento de políticas de qualidade**, e este, por sua vez, é responsável por acionar o estágio **Aprendizagem do conceito**. O modelo de conceito normal é usado pelo **Processamento de políticas de qualidade** e é mantido pelo estágio **Aprendizagem do conceito**, que faz uso de um conjunto de observações de treinamento (conjunto de dados coletados do ambiente em um determinado intervalo de tempo ΔT), e de um algoritmo de indução. *AdaQuali* é composto essencialmente pelos estágios **Processamento de políticas de qualidade** e **Aprendizagem do conceito**.

O pseudocódigo referente aos quatro estágios é apresentado nas Figuras 4.6 a 4.9, e a notação considerada neste trabalho é descrita a seguir. Uma RSSF é formada por um conjunto de nós-sensores $N = \{N_1, \dots, N_n\}$, sendo que cada nó-sensor é composto por um conjunto de sensores $S = \{S_1, \dots, S_s\}$. Cada sensor está apto a efetuar a leitura de um único tipo de fenômeno. O fragmento de consulta (q), quando processado pelos nós-sensores, produz uma amostra por grupo de sensores e nó-sensor $AN_iGS_k = \{AN_iGS_{k1}, \dots, AN_iGS_{km}\}$, onde $1 \leq i \leq n$ e $1 \leq k \leq s$, em um determinado intervalo de tempo ΔT ($\Delta T = q.SendInterval$). Esse fragmento de consulta reporta um conjunto RN_i de dados, e esse conjunto, quando adaptado pelo *AdaQuali* em concordância com a classe de qualidade de serviço de uma dada consulta, passa a ser representado por RN_i' . O pseudocódigo de cada estágio é interpretado a seguir.

- (i) **Estágio 1, Processamento de subconsultas:** este estágio gerencia todo o processamento da subconsulta no nó-sensor e tem como referência as cláusulas *SCHEDULE*, *TIME WINDOW* (ou *DATA WINDOW*), *SENSE INTERVAL* e *SEND INTERVAL* da SNQL (BRAYNER *et al.*, 2006, 2007, 2008), que permitem, respectivamente, controlar o número de execuções da consulta (linha 15), o tempo de sua execução (linha 18), o intervalo de tempo entre as coletas (linha 23) e o intervalo de tempo entre as remessas dessas coletas à estação-base (linha 22). É através deste estágio que os dados são sensoriados em conformidade com cada grupo de sensores envolvido na subconsulta (linha 26). As leituras efetuadas são temporariamente armazenadas (linha 29). Se a cláusula *QUALITY CLASS* for especificada (linha 35), o Estágio 2 (linha 36) será iniciado e *AdaQuali* será executado para aplicar as políticas de qualidade segundo a classe definida. Este estágio configura o parâmetro α com um valor padrão (linha

16) para adaptar o valor da cláusula *SENSE INTERVAL* através do Estágio 4, acionado nas linhas 32 e 40. Ao final, os dados processados são efetivamente encaminhados à estação-base (linhas 38-39);

- (ii) **Estágio 2, Processamento de políticas de qualidade:** instanciado pelo Estágio 1 e somente quando a cláusula *QUALITY CLASS* for especificada, o principal objetivo deste estágio é categorizar os dados coletados para que eles sejam enviados ou não à estação-base em conformidade com o que a classe de qualidade requer. As categorizações são aplicadas para cada grupo de sensores participante da consulta (linha 9). Este estágio é responsável por iniciar o Estágio 3 se o modelo de conceito normal não existir (linha 11), ou se uma mudança no comportamento dos dados for identificada e a classe de qualidade for diferente de *Outlier-Aware* (linhas 26-28). Os *outliers* são tratados em conformidade com a classe de qualidade informada e com o modelo de conceito normal. Para a classe *Outlier-Aware*, uma vez que uma coleta for considerada normal, esta será descartada da consulta, tendo em vista que esta classe requer que somente *outliers* sejam remetidos à estação-base (linhas 19-23). Para as demais classes, os *outliers* serão inicialmente rotulados como “desconhecido” (linhas 14-19) e tratados pelo *AdaQuali* (linhas 26-34). Uma vez que um dado percentual de observações desconhecidas na amostra é identificado por intermédio do parâmetro β (linha 27), o aprendizado do conceito é iniciado (linha 28), sendo a amostra o conjunto de observações de treinamento, o que permitirá o reconhecimento de uma novidade ou uma mudança de conceito. Se uma observação “desconhecida” realmente não pertencer a uma novidade ou uma mudança de conceito, então será descartada (linhas 30 e 32), já que as três classes em questão requerem que somente os dados considerados normais sejam reportados. Restritamente à classe *Coarse-grained Accuracy*, o cálculo do parâmetro α também é realizado neste estágio (linha 37). Conforme mencionado na Seção 4.2, o valor deste parâmetro pode variar no intervalo $[-0,06; -0,01]$ e é configurado proporcionalmente de acordo com a aderência do conjunto de dados coletados ao modelo de conceito normal. Ainda para esta classe, somente um valor aproximado será efetivamente encaminhado quando nenhuma mudança no

comportamento dos dados for identificada (linha 39). Os dados adequados pelas políticas de qualidade são temporariamente armazenados ao final deste estágio (linha 44) para serem transmitidos à rede por meio do Estágio 1. Conforme mencionado na Seção 4.2, a qualidade de serviço de consultas é garantida pelos parâmetros β (linha 27, que garante que β será sempre respeitado) e γ (linha 14, que garante que γ também será sempre respeitado);

- (iii) **Estágio 3, Aprendizagem do conceito:** este estágio trata especificamente do procedimento de identificação de uma novidade ou de uma mudança de conceito e opera sobre os dados coletados no período ΔT , conforme cláusulas *SEND INTERVAL* e *SENSE INTERVAL*, cujo tamanho da amostra contém um percentual de *outliers* maior ou igual a β . Antes de este estágio iniciar, o modelo de conceito normal já deve ter sido induzido pela média dos valores coletados contidos em uma amostra mais antiga. Uma vez que um número de conceitos primitivos (distribuições gaussianas) pode ser encapsulado nesse modelo de conceito normal inicial, a maior distância d_{max} entre o valor médio do modelo de conceito normal e a média de qualquer conceito primitivo é calculada. Para detectar uma novidade ou uma mudança de conceito (linha 4), *AdaQuali* avalia quão próximo o novo conceito aprendido (linha 3) por meio do algoritmo de indução e com base no conjunto de observações de treinamento está do valor médio do modelo de conceito normal. Se a distância exceder d_{max} , a amostra será rotulada como uma novidade; caso contrário, será considerada como uma mudança de conceito (ver Seção 4.2). No primeiro caso, o modelo de conceito normal será atualizado para também representar a novidade, enquanto, no segundo caso, o conceito primitivo que estiver mais próximo do valor médio da amostra aprendida será atualizado para representar a mudança de conceito. A fim de que recursos de armazenamento sejam liberados, ao atualizar o modelo de conceito normal para representar uma novidade, aquele conceito primitivo que não possuir observações recentes que o representem será automaticamente descartado caso o número máximo de conceitos primitivos (n_p) seja atingido. *AdaQuali* armazena a última data em que um conceito primitivo representou

uma observação, possibilitando que os conceitos primitivos mais antigos sejam descartados e que o tamanho do modelo de conceito normal seja gerenciado; e

- (iv) **Estágio 4, Gerenciamento de recursos:** o propósito deste estágio é adaptar as cláusulas *SEND INTERVAL* (linha 8) e *SENSE INTERVAL* (linha 9) no nó-sensor para as classes de qualidade *Fine-grained Accuracy* e *Coarse-grained Accuracy* (linhas 5-7), conforme as disponibilidades de energia e memória, respectivamente, com a meta de preservar os recursos de comunicação e de energia. O valor do parâmetro α é passado à função que adapta a cláusula *SENSE INTERVAL*, assim como é feito em Brayner *et al.* (2006, 2008).

ESTÁGIO 1: Processamento de subconsultasPROCESSAR-SUBCONSULTA (N_i, q)

```

1  entradas:
2     $N_i$ , informações do nó-sensor
3     $q$ , fragmento de consulta a ser executado no nó-sensor
4  variáveis locais:
5    #Consultas, número de execuções da consulta, refere-se à cláusula SCHEDULE da SNQL
6    tempoTimeWindow, temporizador para gerência da cláusula TIME WINDOW
7    tempoSendInterval, temporizador para gerência da cláusula SEND INTERVAL
8    tempoSenseInterval, temporizador para gerência da cláusula SENSE INTERVAL
9    dadoColetado, dado coletado do ambiente pelo nó-sensor
10    $\alpha$ , variável utilizada na adaptação da cláusula SENSE INTERVAL
11 variáveis globais:
12    $AN_iGS_k[l]$ , dados coletados de  $N_i$  por grupo de sensor
13
14    $N_i.SendInterval \leftarrow q.SendInterval$ ;
15   para  $j = 1$  até #Consultas faça
16      $\alpha \leftarrow -0,06$ ;
17     tempoTimeWindow  $\leftarrow$  inicia temporizador; /* similar para a cláusula data window */
18     enquanto tempoTimeWindow <  $q.TimeWindow$  faça
19       tempoSendInterval  $\leftarrow$  inicia temporizador;
20       tempoSenseInterval  $\leftarrow$  inicia temporizador;
21        $N_i.SenseInterval \leftarrow q.SenseInterval$ ;
22       enquanto tempoSendInterval <  $N_i.SendInterval$  e tempoTimeWindow <  $q.TimeWindow$  faça
23         se tempoSenseInterval >=  $N_i.SenseInterval$  então
24           /* um nó-sensor poderá efetuar mais de um tipo de coleta em uma mesma */
25           /* subconsulta, por exemplo: temperatura, umidade e pressão. */
26           para cada grupo de sensores  $N_iGS_k$  especificado em  $q$  faça
27             dadoColetado  $\leftarrow$  coleta dado do sensor;
28             /* armazena dadoColetado */
29              $AN_iGS_k[tamanho(AN_iGS_k) + 1] \leftarrow dadoColetado$ ;
30           fim para
31             tempoSenseInterval  $\leftarrow$  inicia temporizador;
32             GERENCIAR-RECURSOS ( $N_i, q, \alpha$ );
33         fim se
34       fim enquanto
35     se  $q.QualityClass \langle \rangle$  null então
36       PROCESSAR-POLÍTICAS-QUALIDADE ( $N_i, \alpha, q$ );
37     fim se
38     prepara pacote de retorno  $RN_i/RN_i'$  com dados contidos em  $AN_iGS_k[l]$ ;
39     remete pacote de dados à rede para ser encaminhado à estação-base;
40     GERENCIAR-RECURSOS ( $N_i, q, \alpha$ );
41     para cada grupo de sensores  $N_iGS_k$  especificado em  $q$  faça
42        $AN_iGS_k[l] \leftarrow$  esvazia memória;
43     fim para
44   fim enquanto
45 fim para

```

Figura 4.6 - Pseudocódigo do estágio Processamento de subconsultas.

ESTÁGIO 2: Processamento de políticas de qualidadePROCESSAR-POLÍTICAS-QUALIDADE (N_i, α, q)

```

1  entradas:
2       $N_i$ , informações do nó-sensor
3       $q$ , fragmento de consulta em execução
4       $\alpha$ , variável  $\alpha$ 
5  variáveis locais:
6       $ColetasEnvio[]$ , dados sensorizados tratados após aplicação da política de qualidade
7      # $ColetasDesconhecidas$ , quantidade de coletas desconhecidas
8       $\beta \leftarrow$  parâmetro  $\beta$ , percentual mínimo de outliers em uma amostra para iniciar um novo aprendizado
9  para cada grupo de sensores  $N_iGS_k$  especificado em  $q$  faça
10      $ColetasEnvio[] \leftarrow AN_iGS_k[]$ ; /* carrega dados sensorizados referentes ao grupo de sensor */
11     se modelo = null então APRENDER-CONCEITO ( $ColetasEnvio[]$ );
12
13     para  $j = 1$  até tamanho ( $ColetasEnvio[]$ ) faça
14         se ! $ColetaNormal(ColetasEnvio[j])$  então /* conforme limiar  $\gamma$  e modelo de conceito normal */
15             se  $q.QualityClass \diamond Outlier-Aware$  então
16                 intitula  $ColetasEnvio[j]$  como desconhecido;
17                 # $ColetasDesconhecidas++$ ;
18             fim se
19         senão
20             se  $q.QualityClass = Outlier-Aware$  então
21                 remove  $ColetasEnvio[j]$ ;
22             fim se
23         fim se
24     fim para
25
26     se  $q.QualityClass \diamond Outlier-Aware$  então
27         se # $ColetasDesconhecidas$  / tamanho ( $AN_iGS_k[]$ ) * 100  $\geq \beta$  então
28             APRENDER-CONCEITO ( $ColetasEnvio[]$ );
29             requalifica os dados como outlier ou normal com base no modelo de conceito normal;
30             remove dados de  $ColetasEnvio[]$  intitulados como outlier;
31         senão
32             remove dados de  $ColetasEnvio[]$  intitulados como desconhecido; /* nesse momento, outliers */
33         fim se
34     fim se
35
36     se  $q.QualityClass = Coarse-grained\ accuracy$  então
37         calcula  $\alpha$ ; // pode variar de -0,06 a -0,01
38         se # $ColetasDesconhecidas$  / tamanho ( $AN_iGS_k[]$ ) * 100  $< \beta$  então
39              $ColetasEnvio[] \leftarrow$  valor aproximado // média das coletas consideradas normais;
40         fim se
41     fim se
42
43     se  $ColetasEnvio[] \diamond AN_iGS_k[]$  então
44          $AN_iGS_k[] \leftarrow ColetasEnvio[]$ ; /* desconsidera dados removidos da amostra */
45     fim se
46 fim para

```

Figura 4.7 - Pseudocódigo do estágio Processamento de políticas de qualidade.

ESTÁGIO 3: Aprendizagem do conceitoAPRENDER-CONCEITO (*ColetasEnvio[]*)

- 1 **entradas:**
- 2 *ColetasEnvio[]*, conjunto de dados coletados
- 3 aprender conceito da amostra coletada *ColetasEnvio[]*;
- 4 determinar mudança de conceito ou novidade;
- 5 atualizar o modelo de conceito normal ou o conceito primitivo contido nesse modelo;

Figura 4.8 - Pseudocódigo do estágio Aprendizagem do conceito.**ESTÁGIO 4: Gerenciamento de recursos**GERENCIAR-RECURSOS (N_i, q, α)

- 1 **entradas:**
- 2 N_i , informações do nó-sensor
- 3 q , fragmento de consulta em execução
- 4 α , variável α
- 5 **se** $q.QualityClass = Without\ Outliers$ **ou** $q.QualityClass = Outlier-Aware$ **ou** $q.QualityClass = null$ **então**
- 6 retorna;
- 7 **fim se**
- 8 $N_i.SendInterval \leftarrow adaptSendInterval(N_i.getAvailableEnergy(), q.SendInterval, q.TimeWindow)$;
- 9 $N_i.SenseInterval \leftarrow adaptSenseInterval(N_i.getAvailableMemory(), q.SenseInterval, N_i.SendInterval, \alpha)$;

Figura 4.9 - Pseudocódigo do estágio Gerenciamento de recursos.

4.4 Discussão

Alguns trabalhos correlatos à abordagem proposta nesta dissertação foram inicialmente expostos nas Seções 2.5 e 3.3. As propostas contidas nesses trabalhos serão brevemente revisitadas a seguir, de modo a se permitir um melhor contraste com a nova abordagem.

Algumas propostas para a detecção de novidades e *outliers* relacionadas às leituras dos sensores puderam ser observadas na Seção 3.3. As tomadas de decisões dessas são geralmente baseadas em funções de probabilidade do dado coletado ser inconsistente ou não. Algumas dessas propostas geram os respectivos modelos de conceito normal de forma centralizada (DESHPANDE *et al.*, 2004; MUKHERJEE; SEN, 2007; PLESS *et al.*, 2004; REZNIK *et al.*, 2005) ou são utilizadas exclusivamente para identificar *outliers* nos nós-sensores, podendo realizar tratamentos diferenciados, tais como: remover, corrigir ou reportar apenas *outliers* à estação-base (BRANCH *et al.*, 2006; DESHPANDE *et al.*, 2005; ELNAHRAWY; NATH, 2003; MA *et al.*, 2004; MUKHERJEE; SEN, 2007; PALPANAS *et al.*, 2003; PLESS *et al.*,

2004; REZNIK *et al.*, 2005; SUBRAMANIAN *et al.*, 2006; XIAO *et al.*, 2007; ZHUANG; CHEN, 2006).

Outros trabalhos consideram a qualidade de dados apreciando a variabilidade dos requisitos dos usuários, mas empregam abordagens distintas. Delicato (2005) e Delicato *et al.* (2005) não fazem uso de conceitos de banco de dados para o processamento de consultas. Já Deshpande *et al.* (2004, 2005) empregam percentuais probabilísticos que determinam limites de tolerância e de confiança desejados pelo usuário e, conforme esses limites e após estimar as coletas para a consulta, avalia-se se há precisão suficiente para fornecer a resposta à consulta. Em caso afirmativo, a RSSF não é acessada. Por outro lado, Lam *et al.* (2004) utilizam requisitos baseados em percentuais probabilísticos para proporcionar respostas confiáveis a consultas contínuas (e.g., “Que salas possuem 60% de chance de ter uma temperatura maior do que 20°C?”).

Como se pode notar, esses trabalhos não utilizam uma estratégia que considera as especificidades associadas a diferentes aplicações (ou tipos de consultas), que podem ter acesso aos serviços providos por uma mesma RSSF da forma como se propôs nesta dissertação. Através do processamento de consultas com classes de qualidade de serviço, uma nova abordagem de processamento adaptativo de consultas, baseada em noções de qualidade de serviço da consulta e em técnicas detecção de novidades, foi introduzida. Apesar de existir uma ampla variedade de publicações relacionadas a essas técnicas, como as destacadas para as RSSFs e as contidas em Markou e Singh (2003), esses trabalhos não empregam distinções entre *outliers*, novidade e mudança de conceito em RSSFs e, também, não usufruem dessas distinções para ajustar automaticamente, e de diferentes maneiras, o modo como a consulta será processada no nó-sensor, a fim de tornar possível uma preservação de energia global da rede ainda mais intensa.

4.5 Considerações Finais

O presente capítulo teve como intuito apresentar uma extensão do mecanismo de processamento de consultas em RSSFs contido em Brayner *et al.* (2007). Nesse mecanismo introduziu-se o componente *AdaQuali*, que é executado nos próprios nós-sensores, com o objetivo de adequar o processamento das consultas às classes de qualidade de serviço especificadas em alto nível por meio da linguagem declarativa SNQL (BRAYNER *et al.*, 2006, 2007, 2008), também estendida nesta dissertação para dar suporte à proposta. A abordagem emprega técnicas de aprendizado de máquina, mais especificamente, detecção de novidades, e permite balancear dinamicamente a qualidade dos dados esperada pelos usuários e o consumo dos limitados recursos disponíveis nas RSSFs. As classes de qualidade de serviço de consultas foram detalhadas com seus objetivos. Com essas classes de qualidade é possível fornecer diferentes serviços de consultas executadas sobre uma mesma RSSF, considerando diferentes níveis de requisitos, em termos de qualidade dos dados coletados e seus comportamentos temporais, que podem diferir fortemente entre usuários. O pseudocódigo da abordagem também foi exposto e descrito.

No próximo capítulo, poderão ser apreciados os experimentos conduzidos com um protótipo da nova abordagem sobre dados reais, bem como a análise e a discussão dos respectivos resultados.

5 EXPERIMENTOS EFETUADOS E SEUS RESULTADOS

No presente capítulo, os procedimentos experimentais, os materiais e os equipamentos empregados, bem como a análise quantitativa e a discussão dos resultados atinentes aos experimentos, são expostos na Seção 5.1. As considerações finais deste capítulo são efetuadas na Seção 5.2.

5.1 Experimentos, Análise e Discussão dos Resultados

Para a avaliação da abordagem proposta nesta dissertação, considerou-se um simulador desenvolvido em C++ Builder executado em um computador com um processador Intel Pentium Dual 1.6 GHz e 2GB de memória RAM. Esse simulador é uma extensão do simulador proposto por Brayner *et al.* (2006, 2008).

A aplicação do usuário concebida para os experimentos administra informações ambientais, mais especificamente, temperaturas. As coletas consideradas são reais e foram obtidas em Intel Berkeley Research Lab. (2004). Tais leituras foram adquiridas de 54 nós-sensores, implantados no laboratório de pesquisa da Intel, que foram transmitidas à estação-base a cada 31s entre 28/02/2004 e 05/04/2004. Dessas coletas, foram utilizados dados de 53 nós-sensores, excluindo-se o nó-sensor 5 devido à insuficiência de coletas para os experimentos em questão.

O modelo de custo de energia segue o definido em Brayner *et al.* (2007): 125nJ por bit para o sensoriamento de uma coleta e o respectivo processamento, e 75nJ por bit transmitido. Para cada coleta, 8 bytes são destinados para o sensoriamento, a alocação de memória e o processamento, enquanto, para a transmissão dos dados, são requeridos 4 bytes por coleta e 20 bytes para o gerenciamento do pacote.

Para todos experimentos e como prova de conceito, empregou-se a distribuição gaussiana para a aprendizagem e a aplicação das políticas de qualidade. Conforme mencionado na Seção 4.1, além de a distribuição gaussiana ser simples, abordagens estatísticas paramétricas como essa têm sido aplicadas com sucesso em uma variedade de cenários envolvendo técnicas de detecção de novidades, como pode ser apreciado em Markou e Singh (2003).

Com o intuito de verificar a distribuição dos dados utilizados nos experimentos, histogramas das temperaturas coletadas de alguns nós-sensores foram gerados conforme Figura 5.1. Tais histogramas denotam um comportamento dos dados aproximadamente normal, sendo esses dados, portanto, aplicáveis ao modelo empregado nos experimentos. Observa-se que, para a construção desses histogramas, foram descartadas as coletas finais contidas nos arquivos por aparentarem terem sido produzidas em condições anormais.

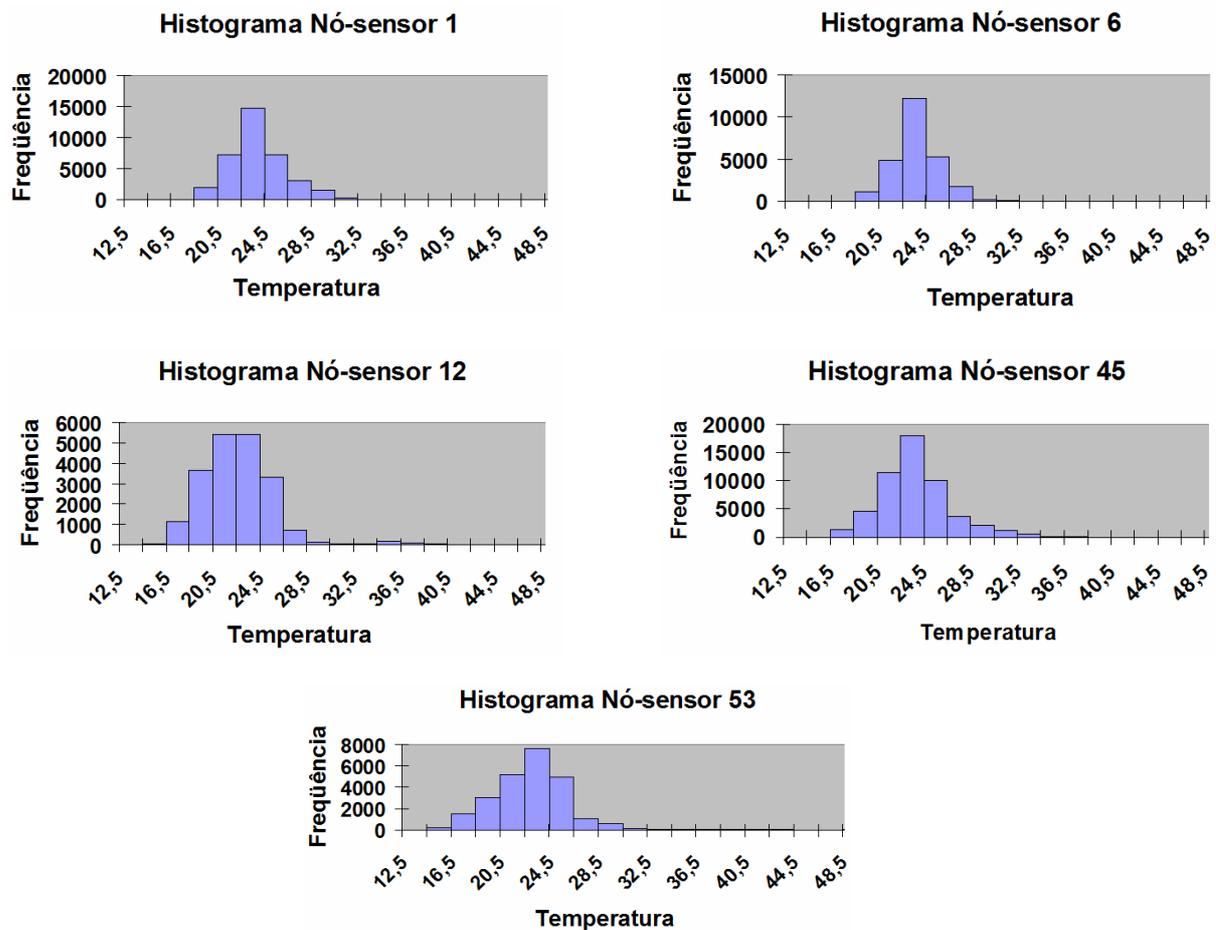


Figura 5.1 - Histogramas gerados para alguns nós-sensores empregados nos experimentos.

A fim de avaliar o consumo de energia, por cada classe de qualidade de serviço, foram feitos experimentos variando o número de nós-sensores participantes da consulta. A consulta empregada é ilustrada na Figura 5.2 em que as temperaturas foram obtidas e tratadas conforme a cláusula *QUALITY CLASS*, onde *<class>* depende da classe de qualidade de serviço definida (*Without Outliers*, *Fine-grained Accuracy*, *Coarse-grained Accuracy* ou *Outlier-Aware*). A consulta foi especificada para durar 100 segundos, obter os dados a cada 100 milissegundos e transmiti-los à rede a cada 10 segundos. Nos experimentos, as temperaturas foram arredondadas para valores inteiros. Para cada execução, considerou-se o mesmo conjunto de dados e respectiva ordenação, exceto para a classe de qualidade *Outlier-Aware*. Para este experimento, as observações são consideradas *outliers* ou desconhecidas se elas estiverem a mais de 2 desvios-padrão (parâmetro γ) distantes de todos os conceitos primitivos existentes no modelo de conceito normal; caso contrário, são consideradas normais. O processo de identificação de uma novidade ou uma mudança de conceito é iniciado quando a amostra analisada contém um percentual de observações desconhecidas (parâmetro β) maior ou igual a 70. Os valores do parâmetro α variam dinamicamente entre -0,06 e -0,01, enquanto o número máximo de conceitos primitivos n_p é igual a 10, o que mantém uma ocupação máxima de memória de 120 bytes para o armazenamento das médias, dos desvios-padrão e das datas que os conceitos primitivos representaram uma observação. As regras para identificação de *outliers*, novidades e mudanças de conceito e as regras para adaptação das cláusulas *SENSE INTERVAL* e *SEND INTERVAL* foram detalhadas na Seção 4.2. Todos os nós-sensores foram inicialmente configurados com as mesmas capacidades de memória e energia, 1.024 bytes e 10^7 nJ, respectivamente. A consulta submetida para a classe de qualidade *Outlier-Aware* foi executada sobre o modelo de conceito normal obtido através da execução da consulta referente à classe de qualidade *Without Outliers*; além disso, as suas coletas foram realizadas a partir da última leitura efetuada para a classe *Without Outliers*.

```
SELECT t.nodeId, t.valorColetado
FROM Temperatura as t
TIME WINDOW 100s
SEND INTERVAL 10s
SENSE INTERVAL 100ms
QUALITY CLASS <class>
```

Figura 5.2 - Consulta SNQL aplicada para avaliar o consumo de energia por classe de qualidade.

A Figura 5.3 exibe os valores de energia consumida de acordo com o número de nós-sensores envolvidos na consulta. Como esperado, a classe *Without Outliers* implicou a maior quantidade de energia utilizada. Em contraste com a abordagem ingênua (*naïve*); isto é, sem a definição da classe de qualidade, somente uma economia de 1,73%, na pior situação, e de 1,90%, na melhor, foram alcançados. Como esta classe de qualidade e a *Outlier-Aware* não permitem mudanças nas taxas de coletas e envios de dados, somente o sensoriamento e o processamento das 1.000 leituras já demandam um consumo de energia de 8×10^6 nJ por nó-sensor. Além disso, para a classe de qualidade *Outlier-Aware*, o volume de *outliers* detectados se manteve estável para este experimento, aproximadamente 17% do total de coletas efetuadas (ver Figura 5.5). Assim, a energia preservada para esta classe com respeito à abordagem ingênua variou de 19,68% a 19,98%. Entretanto, em relação à preservação de energia provida pelas classes *Fine-grained Accuracy* e *Coarse-grained Accuracy*, foi possível se obter os marcos de 28,87% a 29,19% e de 42,85% a 42,99%, respectivamente. Tais ganhos de desempenho são principalmente devido ao comportamento adaptativo dos nós-sensores ocasionado por essas classes de qualidade, em razão de elas permitirem modificações dinâmicas que reduzem o número de coletas inicialmente demandado para a consulta e a quantidade de mensagens encaminhadas à rede.

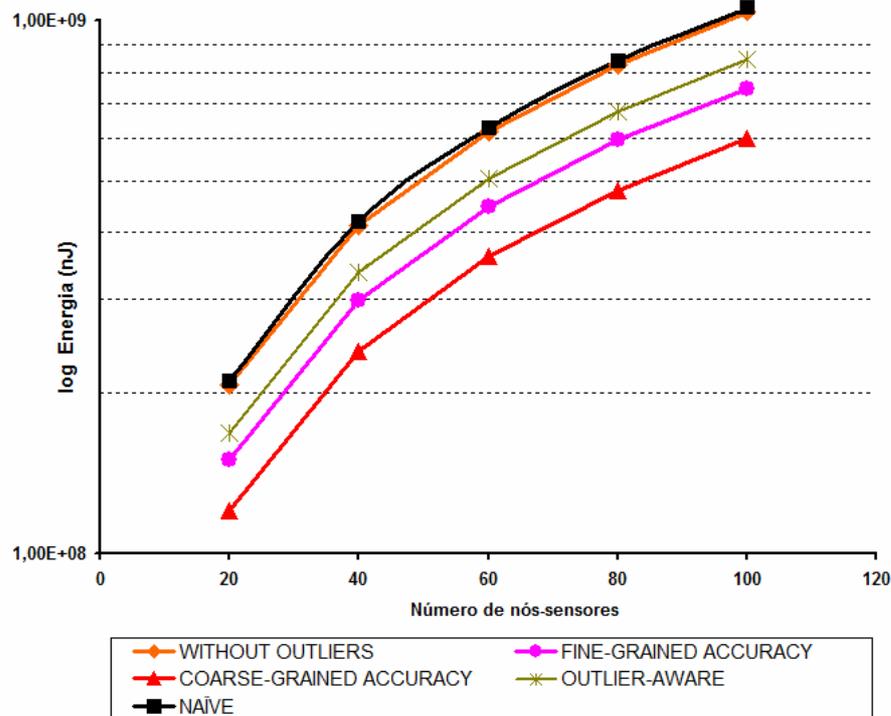
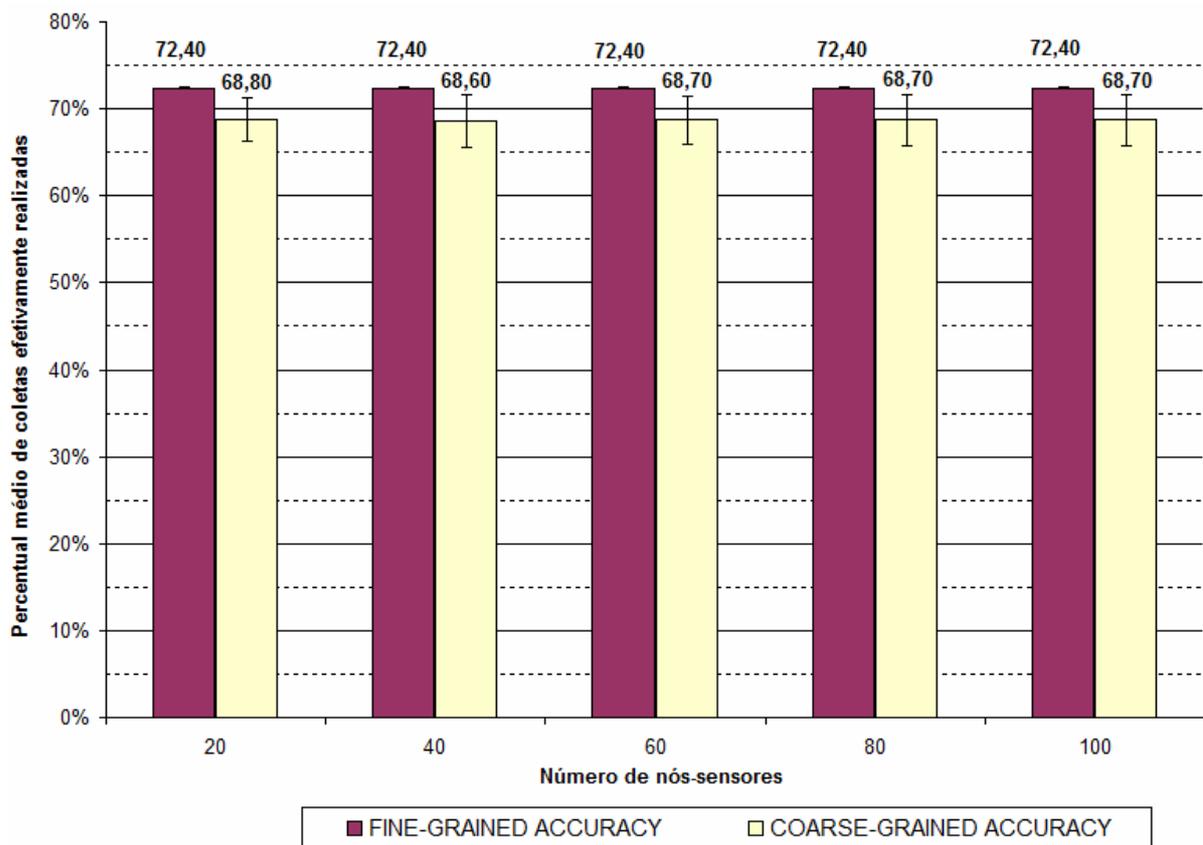


Figura 5.3 - Energia total consumida por quantidade de nós-sensores para cada classe de qualidade.

A Figura 5.4 traz o percentual médio das coletas efetivamente realizadas para as classes *Fine-grained Accuracy* e *Coarse-grained Accuracy* em comparação ao inicialmente previsto para a consulta, isto é, 1.000 leituras. A redução média de coletas efetuadas para a classe de qualidade *Coarse-grained Accuracy* em relação à classe de qualidade *Fine-grained Accuracy* foi de 5,11%. Nota-se que ajustes mais intensos nos intervalos de tempo de sensoriamento ainda podem ocorrer para a classe *Coarse-grained Accuracy*, em razão da função de adaptação ser dependente da disponibilidade de energia, por meio da função que ajusta a cláusula *SEND INTERVAL*, e da disponibilidade de memória, além de depender do parâmetro α . Neste experimento, por exemplo, tanto a memória quanto a energia média disponíveis no momento da adaptação da cláusula *SENSE INTERVAL* eram superiores para a classe de qualidade *Coarse-grained Accuracy*, já que menos coletas foram efetuadas, armazenadas na memória, processadas e remetidas à rede.



Algumas barras de erro (desvios-padrão) são muito pequenas para aparecerem na figura.

Figura 5.4 - Percentual médio de coletas efetivamente realizadas por número de nós-sensores para as classes de qualidade *Fine-grained Accuracy* e *Coarse-grained Accuracy* com respectivo desvio-padrão.

Através dos resultados apresentados na Figura 5.5, percebe-se que, apesar de a classe de qualidade *Outlier-Aware* ter tido um ótimo desempenho na diminuição dos dados enviados à rede para este experimento, o consumo de energia nos nós-sensores (Figura 5.3) foi superior se comparada às classes *Fine-grained Accuracy* e *Coarse-grained Accuracy* por conta da energia despendida para sensoriar e processar os 1.000 valores do fenômeno físico por nó-sensor. Observa-se ainda que as classes de qualidade *Outlier-Aware* e *Without Outliers* apresentam condutas relativamente antagônicas, o que ocorre em razão dos próprios serviços serem opostos: enquanto a primeira classe envia apenas os *outliers*, removendo da consulta os dados considerados normais, a segunda, remove os *outliers* e mantém os dados considerados normais. Essas condutas estão diretamente relacionadas ao modelo de conceito normal induzido existente no nó-sensor, além do próprio comportamento corrente dos dados.

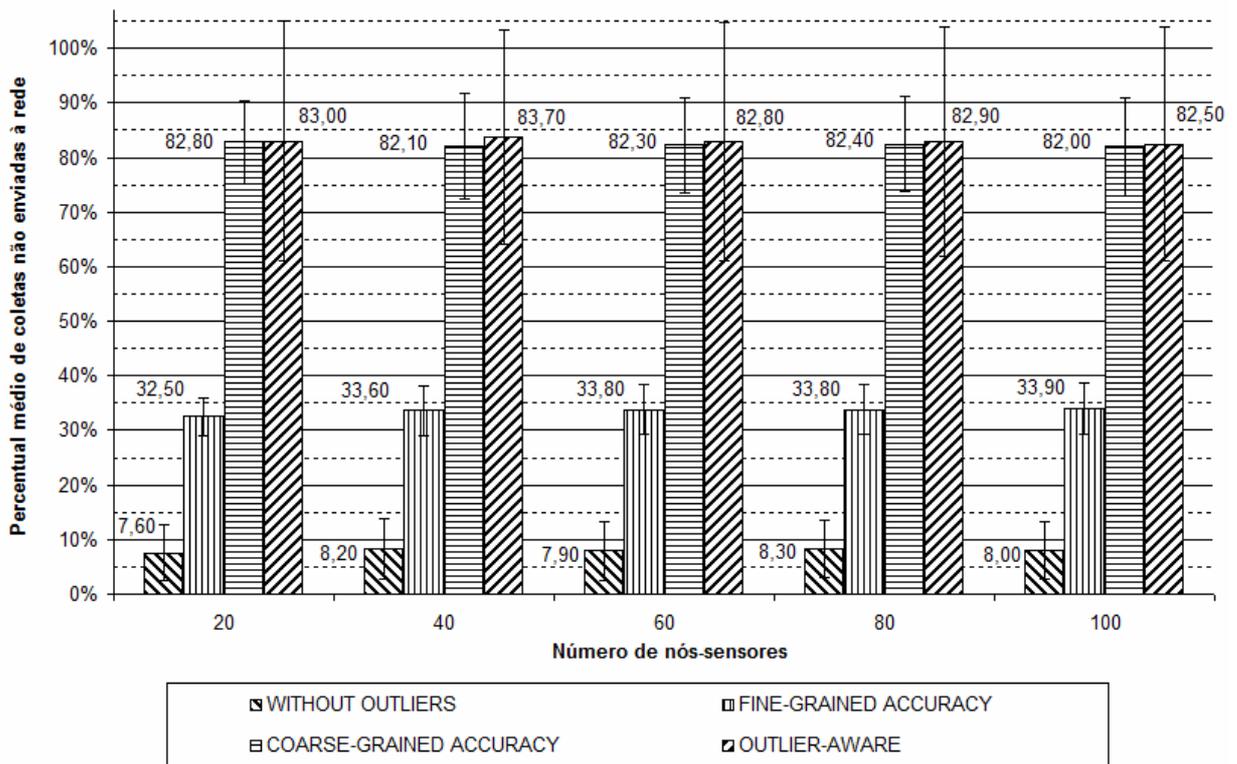


Figura 5.5 - Percentual médio de coletas não enviadas à rede por número de nós-sensores para cada classe de qualidade com respectivo desvio-padrão.

A Figura 5.6 exibe a quantidade de mensagens enviadas à rede, por classe de qualidade de serviço da consulta, conforme número de nós-sensores. A classe de qualidade *Without-
Outlier* encaminhou o mesmo volume de mensagens que a abordagem ingênua (*naïve*). Para a

classe de qualidade *Outlier-Aware*, o percentual de mensagens reduzidas em relação à abordagem ingênua variou entre 75% e 76,50%, enquanto para a classe de qualidade *Fine-grained Accuracy* e *Coarse-grained Accuracy* o percentual de mensagens reduzidas foi de aproximadamente 40% e 30%, respectivamente. Em virtude da classe *Coarse-grained Accuracy* consumir menos energia do que a classe *Fine-grained Accuracy*, as adaptações da cláusula *SEND INTERVAL* são mais suaves para a classe *Coarse-grained Accuracy* quando comparadas com a classe *Fine-grained Accuracy*, propiciando a ocorrência de uma maior quantidade de mensagens na rede.

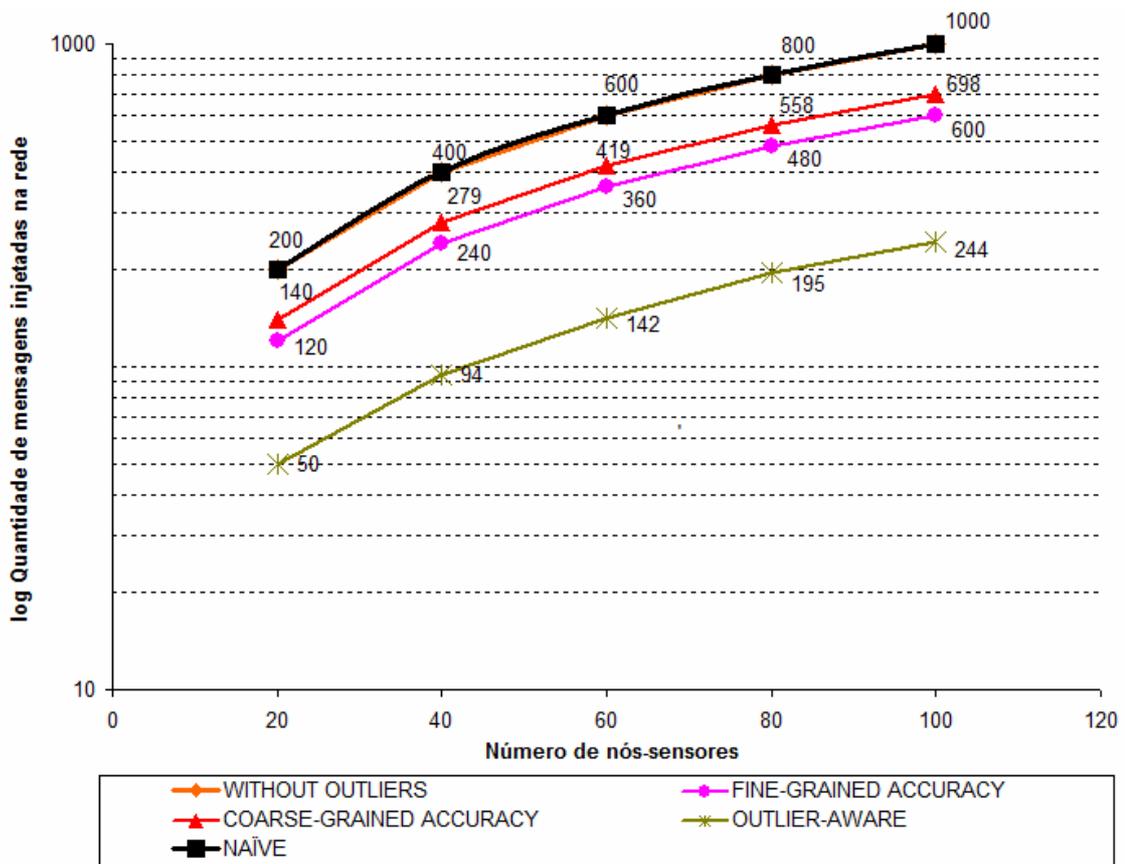


Figura 5.6 - Quantidade de mensagens injetadas na rede por número de nós-sensores para cada classe de qualidade.

Para avaliar o efeito temporal dos dados no comportamento do *AdaQuali*, simulou-se ainda, para os 53 nós-sensores, uma variação no modo como o dado associado ao fenômeno é obtido do arquivo de coletas, obtendo-se, indiretamente, uma variação aproximada nos intervalos de tempo entre coletas sucessivas de três formas: (i) uma coleta a cada 30s – o arquivo é percorrido seqüencialmente sem saltos de registros entre uma leitura e outra, abrangendo os dados e seus instantes de tempo como de fato ocorreram; (ii) uma coleta a cada

3.000s (50min), efetuando-se saltos entre coletas de 100 em 100 registros; e (iii) a cada 12.000s (200min), efetuando-se saltos entre coletas de 400 em 400 registros.

A Figura 5.7 indica que, à medida que o intervalo entre coletas aumentou, as classes de qualidade *Without Outliers* e *Fine-grained Accuracy* classificaram um menor volume de dados como normais, conseqüentemente remetendo menos dados à rede. Observou-se ainda neste experimento que a quantidade de ocorrências de mudanças de conceito e de detecção de novidades também diminuiu conforme o intervalo entre coletas foi maior. Assim, os dados coletados para a classe de qualidade *Coarse-grained Accuracy* tiveram um comportamento mais aderente aos modelos de conceito normal, proporcionando uma adequação mais intensa, em que foi possível obter o marco de 99,30% na redução do volume de dados entregues à rede para o intervalo entre coletas de 12.000s. Por outro lado, as coletas subseqüentes para este mesmo intervalo não foram tão aderentes ao modelo quanto para o intervalo de 3.000s, como pode ser percebido pela classe de qualidade *Outlier-Aware* que inicia suas coletas imediatamente após a última coleta efetuada para a classe *Without Outliers* e cujo percentual de coletas não encaminhadas foi de 85,50% versus 90,30%. Ainda a respeito da classe de qualidade *Outlier-Aware*, a respectiva conduta não foi tão antagônica em relação à classe *Without Outliers*. Ao passo que o número de *outliers* aumentou para a classe *Without Outliers*, o percentual de coletas não enviadas à rede não foi reduzido para a classe de qualidade *Outlier-Aware*.

A Figura 5.8 mostra o percentual de energia preservada da rede conforme variação do intervalo de tempo entre coletas. Conforme pode ser notado, mesmo com uma redução de 99,30% de coletas enviadas à rede para a classe de qualidade *Coarse-grained Accuracy* no intervalo de tempo simulado entre coletas de 12.000s, o percentual de energia preservada foi ligeiramente menor se comparado ao intervalo de tempo de 3.000s, o qual encaminhou mais coletas à rede. Isso se deve em razão de um incremento do volume de coletas efetuadas para o intervalo de tempo de 12.000s em relação ao de 3.000s (ver Figura 5.9).

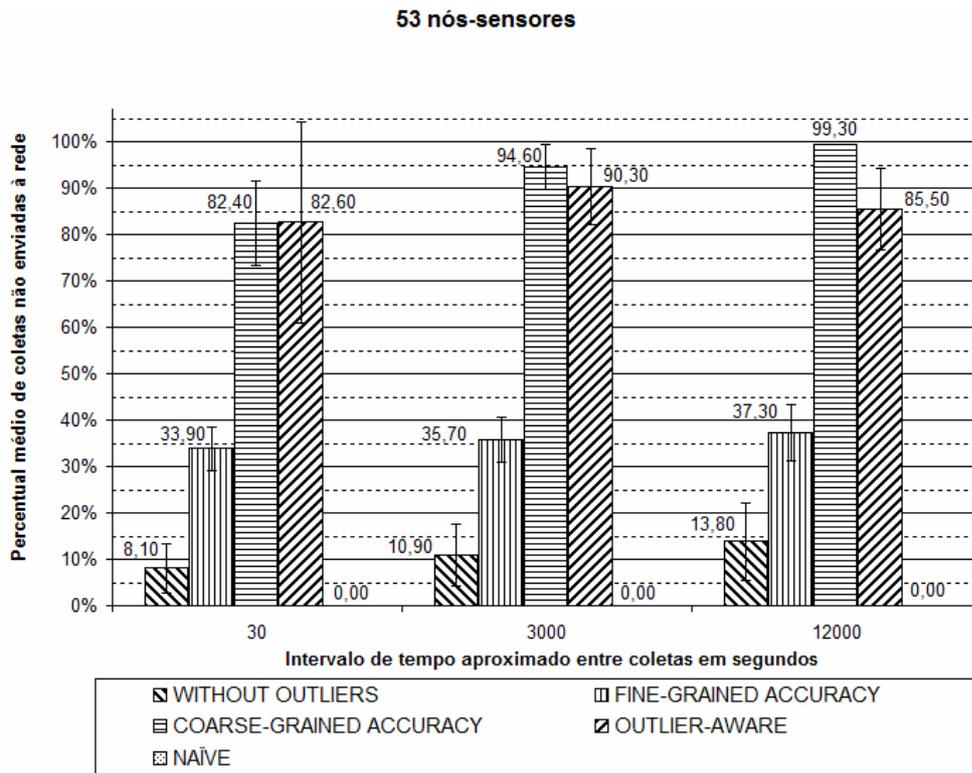


Figura 5.7 - Percentual médio de coletas não enviadas à rede por intervalo de tempo entre coletas para cada classe de qualidade com respectivo desvio-padrão.

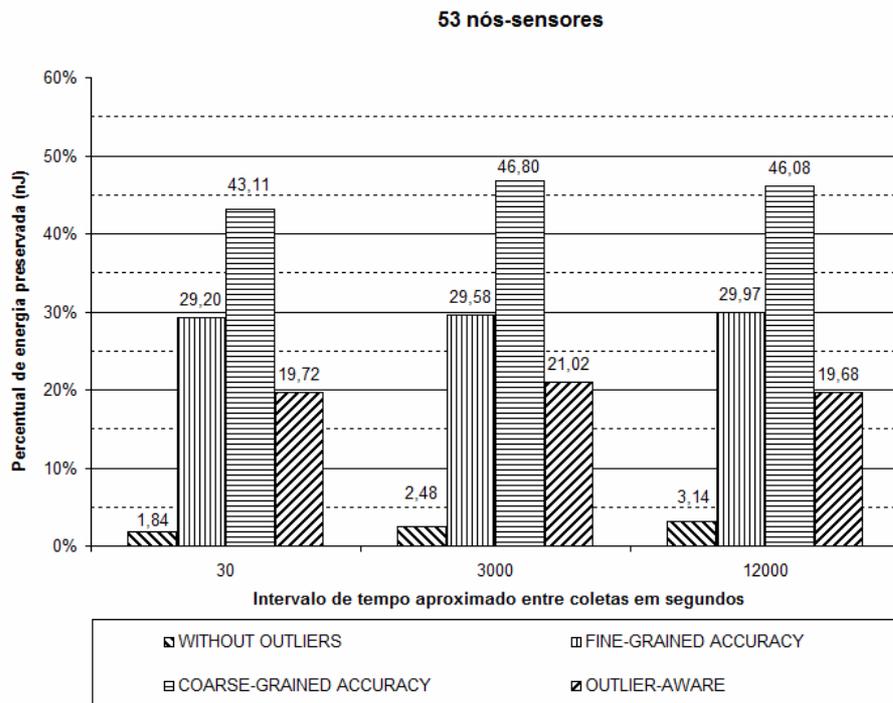


Figura 5.8 - Percentual de energia total preservada por intervalo de tempo entre coletas para cada classe de qualidade.

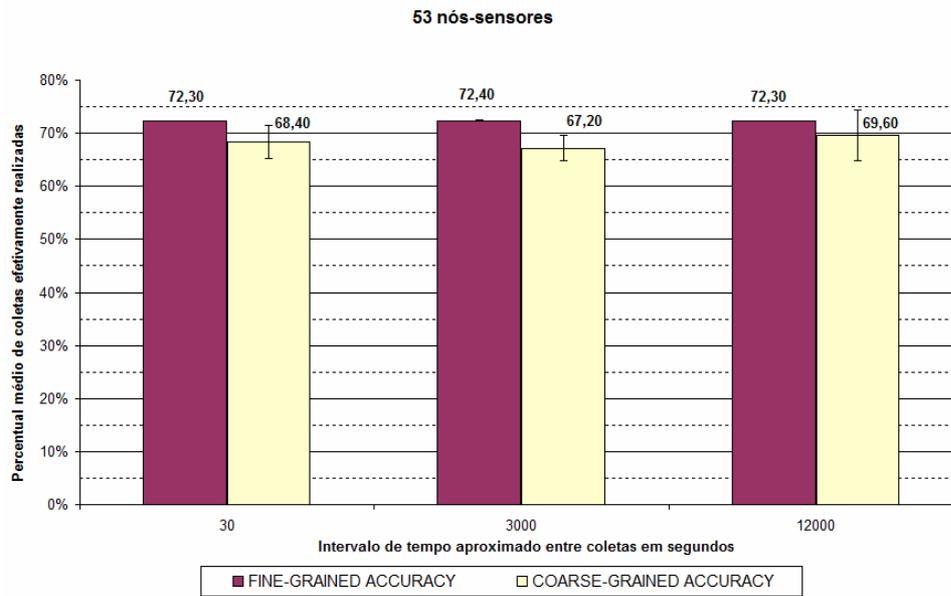


Figura 5.9 - Percentual médio de coletas efetivamente realizadas por intervalo de tempo entre coletas para as classes de qualidade *Fine-grained Accuracy* e *Coarse-grained Accuracy* com respectivo desvio-padrão.

Experimentos também foram conduzidos para avaliar a consequência da escolha do percentual de *outliers* que indica a necessidade de um novo aprendizado, parâmetro β . Nos experimentos, foram considerados os 53 nós-sensores com as mesmas configurações empregadas para o primeiro experimento exposto, sendo que β variou de 20% a 100%. As classes de qualidade de serviço *Without Outliers* e *Outlier-Aware* mantiveram seus comportamentos antagônicos, como pode ser observado nas Figuras 5.10 e 5.11. Apesar de a classe *Outlier-Aware* ter consumido mais energia para parâmetros β superiores, esta classe ainda foi mais eficiente em termos de consumo de energia do que a classe *Without Outliers*, variando o percentual de conservação de energia, comparado à abordagem ingênua, de 16,54% a 21,89%, contra 0,14% a 4,15% da classe *Without Outliers*. Destaca-se que, quanto menor o percentual β , mais rapidamente mudanças de conceito e identificação de novidades poderão ocorrer fazendo com que poucos *outliers* sejam de fato confirmados como tal, o que pode ser percebido na Figura 5.11. Mais especificamente sobre a classe de qualidade *Coarse-grained Accuracy*, houve um pequeno incremento no consumo de energia ao passo que o parâmetro β também aumentou, passando de 44,13% de preservação de energia, no melhor caso, para 42,39%, no pior, sustentando ainda, o melhor desempenho neste quesito. Novamente, este acréscimo de consumo de energia é devido à variação do volume de dados

coletados. Já a classe de qualidade *Fine-grained Accuracy* obteve os marcos de 27,90% (pior caso) e 31,04% (melhor caso) no índice de resguardo da energia dos nós-sensores.

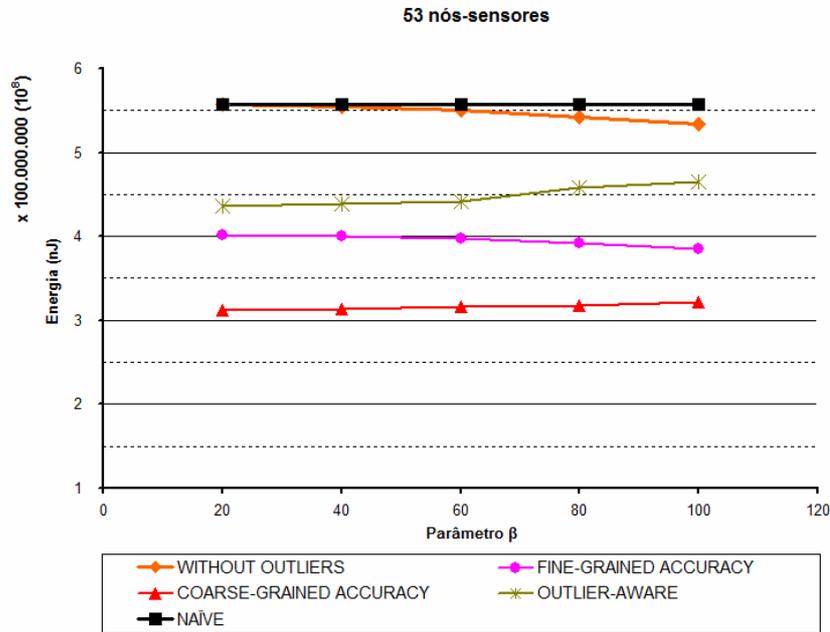


Figura 5.10 - Energia total consumida conforme variação do parâmetro β para cada classe de qualidade.

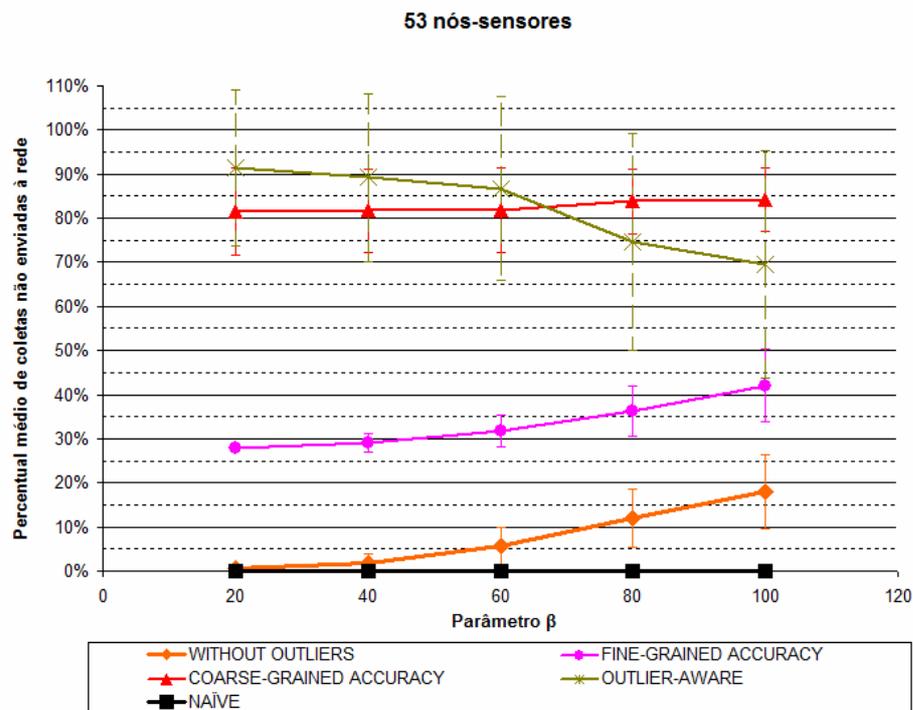


Figura 5.11 - Percentual médio de coletas não enviadas à rede por variação do parâmetro β para cada classe de qualidade com respectivo desvio-padrão.

O efeito do tamanho da amostra das coletas (quantidade de leituras efetuadas por *SEND INTERVAL*, conforme *SENSE INTERVAL*) também foi abordado nos experimentos, mantendo-se a quantidade de 53 nós-sensores e os parâmetros configurados para o primeiro experimento. Para tal, variou-se apenas o valor da cláusula *SEND INTERVAL* entre 2.000ms e 10.000ms, permitindo envios parciais a cada 20 e 100 coletas, respectivamente, desde que não haja adaptação da cláusula *SEND INTERVAL*. A Figura 5.12 mostra que a energia preservada foi superior à medida que a cláusula *SEND INTERVAL* foi maior (exceto para a classe de qualidade *Outlier-Aware*) devido: (i) à diminuição de remessas parciais de coletas pelos nós-sensores por conta da própria cláusula *SEND INTERVAL*; e (ii) à redução de coletas enviadas à rede por conta da detecção e remoção de *outliers*, da adaptação da cláusula *SENSE INTERVAL* para as classes *Fine-grained Accuracy* e *Coarse-grained Accuracy*, e do comportamento da classe de qualidade *Coarse-grained Accuracy*, que ainda possibilita o envio de um valor aproximado em substituição à amostra quando não há mudança comportamental dos dados.

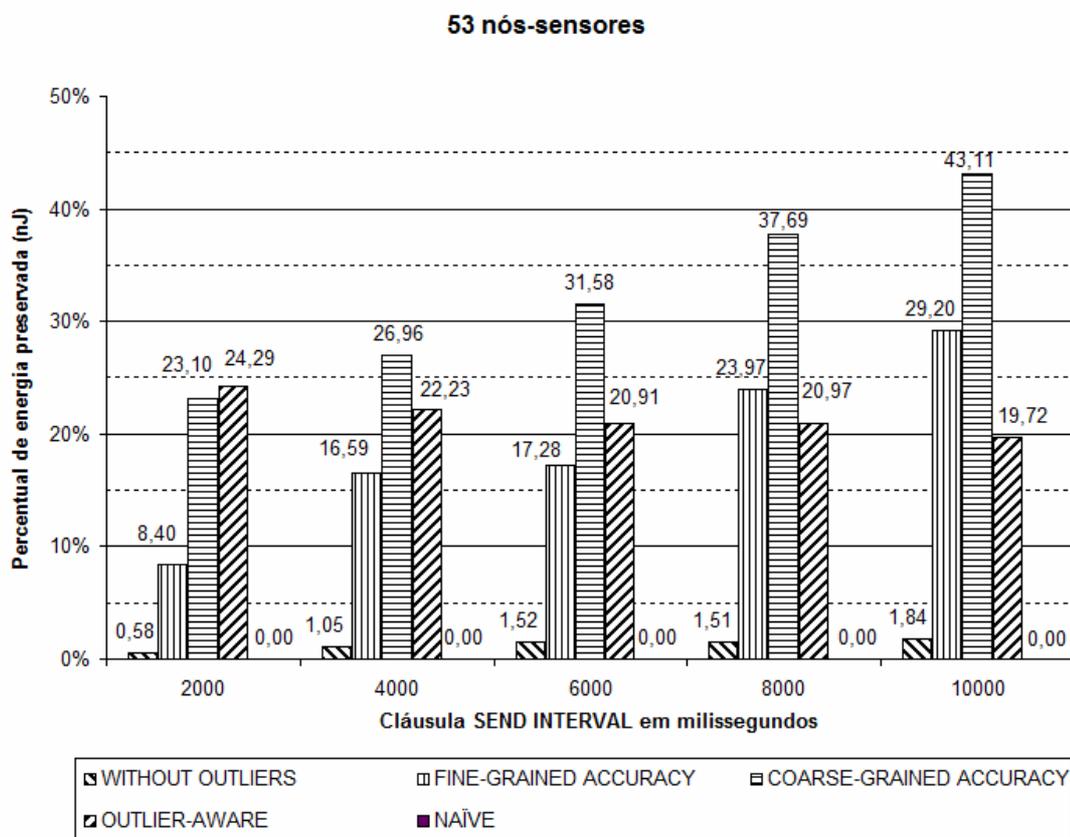


Figura 5.12 - Percentual de energia preservada por variação da cláusula *SEND INTERVAL* para cada classe de qualidade.

Na Figura 5.13, pode-se notar uma variação diferente no percentual médio de coletas não enviadas à estação-base para a classe *Coarse-grained Accuracy*, se comparada às classes *Without Outliers* e *Fine-grained Accuracy*, conforme modificação da cláusula *SEND INTERVAL*. Para este experimento, essa variação está fortemente associada às ocorrências de mudanças comportamentais dos dados, que interferem no volume de dados reportados à estação-base cujo novo conceito foi aprendido, além da adaptação da cláusula *SENSE INTERVAL* de acordo com a disponibilidade de memória. Como exemplo dessa variação, o Quadro 5.1 descreve o quantitativo de coletas para o nó-sensor 54 de acordo com a cláusula *SEND INTERVAL* e a classe de qualidade *Coarse-grained Accuracy*. Para a cláusula *SEND INTERVAL* igual a 2.000ms, 6 mudanças comportamentais dos dados foram identificadas ocasionando a remessa de 319 coletas relacionadas a essas mudanças, enquanto para a cláusula *SEND INTERVAL* igual a 4.000ms, 5 mudanças comportamentais foram detectadas e 411 coletas relacionadas foram encaminhadas à estação-base. A quantidade de coletas efetuadas reduz conforme *SEND INTERVAL* aumenta em razão de mais dados estarem alocados em memória, ocasionando mais rapidamente o ajuste da cláusula *SENSE INTERVAL*.

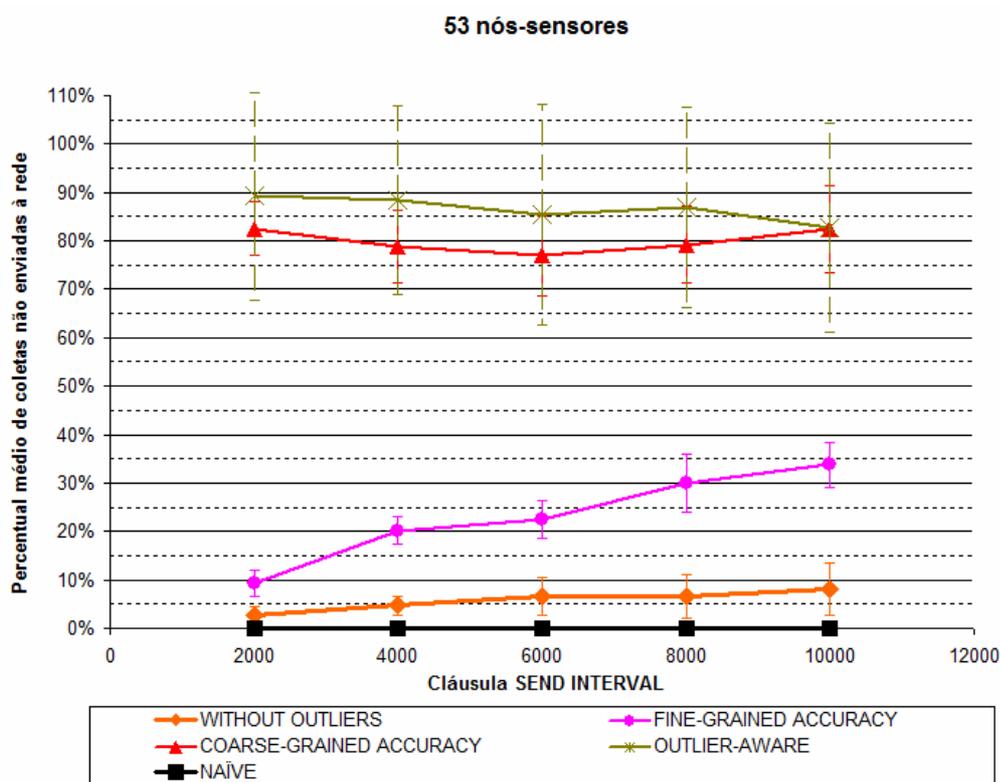


Figura 5.13 - Percentual médio de coletas não enviadas à rede por variação da cláusula *SEND INTERVAL* para cada classe de qualidade com respectivo desvio-padrão.

Cláusula <i>SEND INTERVAL</i>	Quantidade prevista de coletas para a consulta	Quantidade de coletas realizadas	Quantidade de coletas não realizadas	Quantidade de coletas enviadas	Quantidade de coletas não enviadas
2000ms	1000	977	23	341	1000 - 341 = 659
4000ms	1000	901	99	420	1000 - 420 = 580
6000ms	1000	785	215	277	1000 - 277 = 723
8000ms	1000	725	275	198	1000 - 198 = 802
10000ms	1000	650	350	89	1000 - 89 = 911

Quadro 5.1 - Quantitativo de coletas para o nó-sensor 54 por cláusula *SEND INTERVAL* para a classe de qualidade *Coarse-grained Accuracy*.

Com o propósito de verificar o tempo de processamento despendido pelo *AdaQuali* (Estágios 2 e 3, Seção 4.3) à medida que o tamanho da amostra das coletas aumenta, experimentos foram conduzidos variando a quantidade de coletas da amostra de 100 a 500, tendo-se como base a classe de qualidade *Without Outliers*. Cada consulta compreendeu a quantidade de 10 amostras processadas, delimitadas pelas cláusulas *TIME WINDOW* e *SEND INTERVAL*, e 53 nós-sensores. O Quadro 5.2 exibe os valores das cláusulas *TIME WINDOW*, *SEND INTERVAL* e *SENSE INTERVAL* empregadas nas consultas SNQL em que cada uma permitiu uma quantidade de coletas diferenciada por amostra, segundo as cláusulas *SEND INTERVAL* e *SENSE INTERVAL*. Cada consulta foi executada 10 vezes e o arquivo de coletas foi percorrido seqüencialmente, iniciando-se pelas 10 consultas que coletam 100 valores do fenômeno físico, posteriormente as 10 consultas que coletam 200 valores do fenômeno físico, e, assim, sucessivamente. A Figura 5.14 apresenta o tempo médio gasto em milissegundos das 10 simulações para os 53 nós-sensores conforme a quantidade de coletas analisada por amostra. Devido à complexidade computacional estar associada ao número de registros envolvidos durante o processamento do algoritmo, já era de se esperar um incremento no tempo de execução do *AdaQuali* proporcional à quantidade de coletas processadas. Mesmo assim, para o processamento de 500 coletas em um nó-sensor, foram necessários apenas 4,9596ms, o que poderá ser insignificante se comparado aos benefícios que a abordagem com a utilização de técnicas de detecção de novidades e classes de qualidade de serviço de consultas pode trazer. Além disso, o experimento denota que o tempo despendido para *AdaQuali* processar uma amostra tende a ser linear conforme o tamanho da amostra aumenta.

<i>TIME WINDOW</i>	<i>SEND INTERVAL</i>	<i>SENSE INTERVAL</i>	Tamanho Amostra
100s	10s	100ms	100 coletas
200s	20s	100ms	200 coletas
300s	30s	100ms	300 coletas
400s	40s	100ms	400 coletas
500s	50s	100ms	500 coletas

Quadro 5.2 - Cláusulas *TIME WINDOW*, *SEND INTERVAL* e *SENSE INTERVAL* empregadas nas consultas para avaliação do tempo médio de execução do *AdaQuali*.

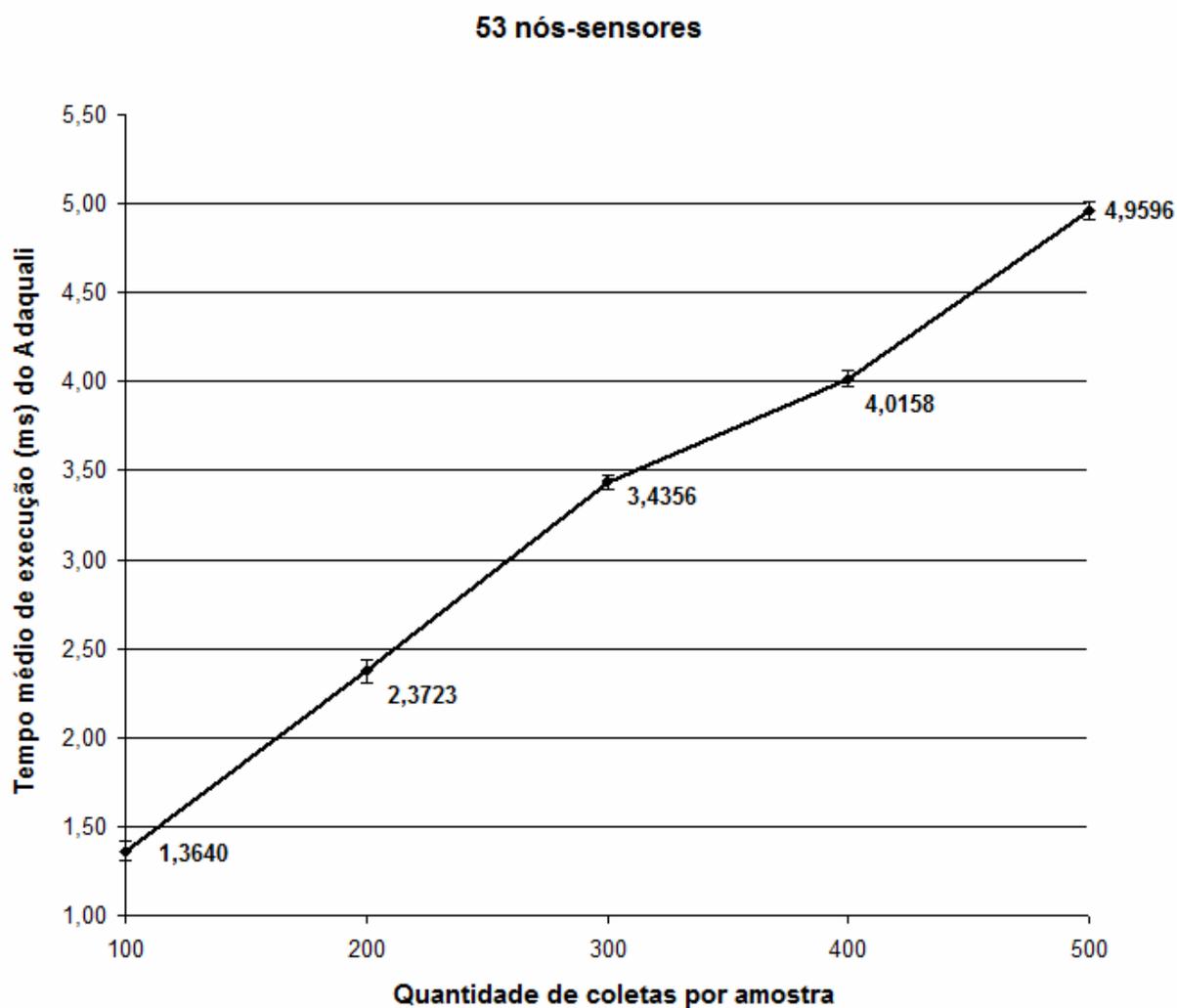


Figura 5.14 - Tempo médio de execução do *AdaQuali* em milissegundos por tamanho da amostra para a classe *Without Outliers* com respectivo desvio-padrão.

Através dos experimentos conduzidos, nota-se que há uma relação direta entre acurácia, latência e eficiência no consumo de energia. O processamento de consultas com classes de qualidade de serviço proposto neste trabalho permite aos usuários requisitarem as consultas

para serem atendidas com acurácias e latências diferenciadas e ainda reduzir o consumo de energia dos nós-sensores. Quanto maior a acurácia relacionada à classe de qualidade de serviço especificada na consulta, isto é, a qualidade do dado e a precisão temporal da coleta, maior poderá ser a energia demandada para o processamento. Apesar de algumas classes de qualidade de serviço adaptarem de forma explícita o comportamento dos nós-sensores com respeito ao tempo de remessa de dados à estação-base, postergando a entrega, há ainda a questão do tempo despendido para a aprendizagem do conceito inerente aos dados coletados, o que poderá variar de acordo com o método aplicado e o conjunto de observações de treinamento envolvido. Quanto maior a quantidade de observações (registros) e características (atributos ou campos), maior poderá ser a complexidade do aprendizado, conseqüentemente, o tempo e o custo para processamento também poderão ser ampliados.

5.2 Considerações Finais

Neste capítulo, os experimentos efetuados a respeito da abordagem introduzida nesta dissertação foram expostos, analisados e discutidos. Esses experimentos indicam que a proposta apresentada fornece um recurso eficaz para a redução do consumo de energia dos nós-sensores ao passo que tenta atender às especificidades das aplicações e às diferentes necessidades dos usuários.

6 CONCLUSÃO E TRABALHOS FUTUROS

Neste trabalho, foi apresentada uma nova abordagem para o processamento adaptativo de consultas em RSSFs baseada em noções de qualidade de serviço da consulta e em técnicas de detecção de novidades. Essas técnicas foram embutidas nos dispositivos de sensoriamento e usufruídas para modificar dinamicamente seus comportamentos com o intuito de melhorar o processamento de consultas nessas redes.

A abordagem proposta nesta dissertação teve por finalidade o consumo inteligente dos limitados recursos disponíveis das RSSFs, a fim de permitir um prolongamento da vida útil dessas redes, enquanto ainda enfatiza o atendimento à qualidade esperada pelos seus usuários durante o processamento das consultas nos próprios nós-sensores.

Como resultado deste trabalho, quatro classes de qualidade de serviço de consultas foram definidas, tendo-se em consideração diferentes tipos de requisitos em termos de qualidade dos dados e comportamento temporal das coletas e respectivo envio à estação-base. Cada classe de qualidade de serviço de consulta é associada a uma determinada política de qualidade que indica, por meio de procedimentos e critérios aqui apresentados, como as consultas relacionadas a cada classe devem ser servidas pelos nós-sensores, o que concede flexibilidade aos usuários.

Para suporte à nova abordagem, a linguagem de consulta declarativa SNQL (*Sensor Network Query Language*) (BRAYNER *et al.*, 2007) foi estendida para introdução da cláusula *QUALITY CLASS*. O algoritmo *AdaQuali* (*ADaptive QUALity control for query processing in wireless sensor networks*) (BRAYNER *et. al.*, 2010) também foi introduzido detalhadamente como uma forma de controlar as atividades dos nós-sensores por meio do ajuste de suas frequências de sensoriamento e transmissão de dados. Esse algoritmo foi concebido para ser executado nos nós-sensores como um componente do processador de consultas proposto em Brayner *et al.* (2007) quando a consulta é processada. O comportamento de cada nó-sensor

é ajustado em conformidade com o contexto corrente, fundamentado nos seguintes fatores: (i) disponibilidade de memória; (ii) disponibilidade de energia; e (iii) classificação dos dados coletados de acordo com o modelo de conceito normal.

Através de experimentos simulados, ganhos significativos em termos de conservação de energia puderam ser alcançados quando a nova abordagem foi comparada à abordagem ingênua. Isso se deu tanto por conta da redução de dados efetivamente coletados quanto por conta da redução de dados transmitidos à rede para serem entregues à estação-base, especialmente para a classe de qualidade *Coarse-grained Accuracy* cujos experimentos apontaram para um ganho de conservação de energia de até 46,80% no experimento que avalia o efeito temporal dos dados.

Para as aplicações em que a acurácia e o comportamento temporal dos envios dos dados coletados podem ser aproximados, a classe de qualidade *Coarse-grained Accuracy* pode ser a melhor opção, já que os resultados experimentais apontaram para os maiores percentuais de preservação de energia, chegando ao marco de 15,76% a mais de resguardo de energia se comparado com o maior percentual obtido para a classe *Fine-grained Accuracy* (46,80% versus 31,04%). Já para as situações em que o comportamento temporal dos dados não pode ser aproximado, a classe *Outlier-Aware* mostra-se como uma boa alternativa à classe *Without Outliers*; para esta, o maior percentual de conservação de energia detectado foi 4,15%, e o menor, 0,14% versus 21,89% e 16,54% da classe *Outlier-Aware*, no experimento em que se avalia o efeito do percentual β na abordagem.

Ressalta-se que, com a diminuição de dados remetidos à rede pelo nó-sensor processador da consulta, o efeito será generalizado por toda a rede e não se dará apenas nos nós-sensores participantes da consulta, tendo em vista que o custo de recebimento e transmissão de dados por outros nós-sensores responsáveis pela entrega dos dados à estação-base também poderá ser menor.

Como trabalhos futuros, planejam-se as seguintes atividades:

- (i) a condução de novos experimentos abrangendo outros métodos de detecção de novidades e mudanças de conceitos embutidos nos nós-sensores, e.g., regra dos k -vizinhos mais próximos;

- (ii) a alteração da proposta para permitir a troca de conhecimento entre nós-sensores em uma mesma circunvizinhança, possibilitando que *AdaQuali* também avalie, durante a realização de seus ajustes, as informações dos nós-sensores na vizinhança onde ele está sendo executado;
- (iii) a definição de métricas para avaliar o quanto os dados coletados estão atendendo ao que foi demandado através da classe de qualidade de serviço da consulta;
- (iv) a realização de outros experimentos com as mesmas características dos já conduzidos nesta dissertação, mas abrangendo outros tipos de fenômenos, tais como os de umidade e luminosidade, o que pode agregar novos resultados à abordagem aqui proposta;
- (v) a investigação de novas classes de qualidade de serviço de consultas; por exemplo, uma alternativa à classe *Outlier-Aware*, para a qual adaptações pudessem ser efetuadas de acordo com a frequência de *outliers* detectados. Nesse caso, uma possível abordagem seria diminuir o número de coletas nas situações em que *outliers* não tenham sido identificados durante uma determinada janela de tempo;
- (vi) a execução de experimentos mais detalhados para comparar o comportamento das classes *Coarse-grained Accuracy* e *Fine-grained Accuracy* conforme variação da disponibilidade de memória inicial, mantendo-se as mesmas características dos experimentos efetuados neste trabalho. Com percentuais de memória disponíveis mais elevados, espera-se alcançar ganhos mais significativos para a classe *Coarse-grained Accuracy* em relação à classe *Fine-grained Accuracy*; e
- (vii) a implementação do *AdaQuali* em uma RSSF real.

REFERÊNCIAS

AKYILDIZ, I. F.; SU, W.; SANKARASUBRAMANIAM, Y.; CAYIRCI, E. “*A Survey on Sensor Networks*”. IEEE Communication Magazine, IEEE Communications Society, volume 40, issue 8, 2002. pp. 102-114.

ALMEIDA, G. M. “*Detecção de Situações Anormais em Caldeiras de Recuperação Química*”. 2006. Tese (Doutorado em Engenharia Química) – Escola Politécnica da Universidade de São Paulo, São Paulo, Brasil. Disponível em: <<http://www.teses.usp.br/teses/disponiveis/3/3137/tde-01122006-155750/>>. Acesso em: 02 out. 2007.

ATMEL CORPORATION. “*[ATmega128L] Features*”. 2007. Disponível em: <http://www.atmel.com/dyn/resources/prod_documents/doc2467.pdf>. Acesso em: 25 maio 2008.

BRAGA, A. P.; CARVALHO, A. C. P. L. F.; LUDERMIR, T. B. Redes Neurais Artificiais. In: REZENDE, S. O. (Org.) “*Sistemas Inteligentes: Fundamentos e Aplicações*”. 1. ed. Barueri, SP: Manole, 2003. cap. 6, pp. 141-168.

BRANCH, J.; SZYMANSKI, B.; GIANNELLA, C.; WOLFF, R.; KARGUPTA, H. “*In-Network Outlier Detection in Wireless Sensor Networks*”. In: International Conference on Distributed Computing Systems (ICDCS), 26, 2006, Lisboa, Portugal. Proceedings... Washington, USA: IEEE Computer Society, 2006. pp. 51-58.

BRAYNER, A.; LOPES, A.; MEIRA, D.; VASCONCELOS, R.; MENEZES, R. “*ADAGA: Adaptive AGgregation Algorithm for Sensor Networks*”. In: Simpósio Brasileiro de Banco de Dados (SBBDD), 21, 2006, Florianópolis, Brasil. Anais... Florianópolis: Sociedade Brasileira de Computação, 2006. pp. 191-205.

BRAYNER, A.; LOPES, A.; MEIRA, D.; VASCONCELOS, R.; MENEZES, R. “*Toward Adaptive Query Processing in Wireless Sensor Networks*”. Signal Processing, Amsterdam, The Netherlands: Elsevier North-Holland, volume 87, issue 12, 2007. pp. 2911-2933.

BRAYNER, A.; LOPES, A.; MEIRA, D.; VASCONCELOS, R.; MENEZES, R. “*An Adaptive In-network Aggregation Operator for Query Processing in Wireless Sensor Networks*”. Journal of Systems and Software, New York, USA: Elsevier Science, volume 81, issue 3, 2008. pp. 328-342.

BRAYNER, A.; COELHO, A.; SOUZA, K. M. “*Dealing with Application Requirements and Energy Consumption in Wireless Sensor Networks: A Novelty Detection Approach for Quality of Query Services*”. In: Symposium on Applied Computing (SAC), 25, 2010, Sierre, Switzerland. Accepted Paper.

CHEN, D.; VARSHNEY, P. K. “*QoS Support in Wireless Sensor Networks: A Survey*”. In: International Conference on Wireless Networks, 2004, Las Vegas, USA. Proceedings... CSREA Press, volume 2, 2004.

CHOU, Y. “*Statistical Analysis: with Business and Economic Applications*”. Holt, Rinehart and Winston, 1969.

CROSSBOW TECHNOLOGY INC. 2008. Disponível em: <<http://www.xbow.com>>. Acesso em: 11 fev. 2008.

DELICATO, F. C. “*Middleware Baseado em Serviços para Redes de Sensores Sem Fio*”. 2005. Tese (Doutorado em Engenharia Elétrica) – Universidade Federal do Rio de Janeiro, Rio de Janeiro, Brasil. Disponível em: <<http://www.gta.ufrj.br/ftp/gta/TechReports/Delicato05/Delicato05.pdf>>. Acesso em: 09 mar. 2008.

DELICATO, F.; PROTTI, F.; REZENDE, J. F.; PIRMEZ, L. “*Uma Abordagem Baseada em QoS para Seleção de Nós Ativos em Redes de Sensores Sem Fio*”. In: Simpósio Brasileiro em Redes de Computadores (SBRC), 23, 2005, Fortaleza, Brasil. Anais... Fortaleza: Sociedade Brasileira de Computação, 2005. pp. 367-380.

DESHPANDE, A.; GUESTRIN, C.; MADDEN, S. R.; HELLERSTEIN, J. M.; HONG, W. “*Model-Driven Data Acquisition in Sensor Networks*”. In: International Conference on Very Large DataBase (VLDB), 30, 2004, Toronto, Canadá. Proceedings... VLDB Endowment, 2004. pp. 588-599.

DESHPANDE, A.; GUESTIN, C.; MADDEN, S. R. **“Using Probabilistic Models for Data Management in Acquisitional Environments”**. In: Conference on Innovative Data Systems Research (CIDR), 2, 2005, Asilomar, USA. Proceedings... VLDB Endowment, 2005. pp. 317-328. Disponível em: <<http://www.cidrdb.org/cidr2005/papers/P26.pdf>> Acesso em: 15 abr. 2008.

ELNAHRAWY, E.; NATH, B. **“Cleaning and Querying Noisy Sensors”**. In: International Workshop on Wireless Sensor Network and Applications (WSNA), 2, 2003, San Diego, USA. Proceedings... New York, USA: ACM, 2003. pp. 78-87.

ESTRIN, D.; GOVINDAN, R.; HEIDEMANN, J.; KUMAR, S. **“Next Century Challenges: Scalable Coordination in Sensor Networks”**. In: International Conference on Mobile Computing and Networking (MOBICOM), 5, 1999, Seattle, USA. Proceedings... New York, USA: ACM, 1999. pp. 263-270.

FONSECA, J. S.; MARTINS, G. A. **“Curso de Estatística”**. 3. ed. São Paulo: Atlas, 1982.

FRAWLEY, J. W.; PIATETSKY-SHAPIRO, G.; MATHEUS, C. J. **“Knowledge Discovery in Databases: An Overview”**. AI Magazine, Menlo Park, USA: AAAI Press, volume 13, issue 3, 1992. pp. 57-70.

FREUND, J. E.; SIMON, G. A. **“Estatística Aplicada: Economia, Administração e Contabilidade”**. 9. ed. Porto Alegre, RS: Bookman, 2000.

GARCÍA-HERNÁNDEZ, C. F.; IBARGÜENGOYTIA-GONZÁLEZ, P. H.; GARCÍA-HERNÁNDEZ, J.; PÉREZ-DÍAZ, J. A. **“Wireless Sensor Networks and Applications: A Survey”**. International Journal of Computer Science and Network Security (IJCSNS), Dr. Sang H. Lee, volume 7, no. 3, 2007. pp. 264-273.

HILL, J.; SZEWCZYK, R.; WOO, A.; HOLLAR, S.; CULLER, D.; PISTER, K. **“System Architecture Directions for Networked Sensors”**. ACM SIGPLAN Notices, New York, USA: ACM, volume 35, issue 11, 2000. pp. 93-104.

HODGE, V. J.; AUSTIN, J. **“A Survey of Outlier Detection Methodologies”**. Artificial Intelligence Review, Norwell, USA: Kluwer Academic Publishers, volume 22, issue 2, 2004. pp. 85-126.

INTEL BERKELEY RESEARCH LAB. **“Intel Lab Data”**. 2004. Disponível em: <<http://db.lcs.mit.edu/labdata/labdata.html>> Acesso em: 27 abr. 2009.

LAM, K.; CHENG, R.; LIANG, B.; CHAU, J. “*Sensor Node Selection for Execution of Continuous Probabilistic Queries in Wireless Sensor Networks*”. In: International Workshop on Video Surveillance & Sensor Networks, 2, New York, USA, 2004. Proceedings... New York, USA: ACM, 2004. pp. 63-71.

LEVIS, P. “*TinyOS 2.0 Overview*”. 2006. Disponível em: <<http://www.tinyos.net/tinyos-2.x/doc/html/overview.html>>. Acesso em: 08 fev. 2008.

MA, X.; YANG, D.; TANG, S.; LUO, Q.; ZHANG, D.; LI, S. “*Online Mining in Sensor Networks*”. In: IFIP International Conference on Network and Parallel Computing (NPC), 2004, Wuhan, China. Proceedings... Springer, LNCS 3222, 2004. pp. 544-550.

MADDEN, S. R.; FRANKLIN, M. J.; HELLERSTEIN, J. M.; HONG, W. “*TAG: A Tiny Aggregation Service for Ad-hoc Sensor Networks*”. In: Symposium on Operating Systems Design and Implementation (OSDI), 5, 2002, Boston, USA. Proceedings... New York, USA: ACM, volume 36, issue SI, 2002. pp. 131-146.

MADDEN, S. R.; FRANKLIN, M. J.; HELLERSTEIN, J. M.; HONG, W. “*TinyDB: An Acquisitional Query Processing System for Sensor Networks*”. Transactions on Database Systems (TODS), New York, USA: ACM Press, volume 30, issue 1, 2005. pp. 122-173.

MARKOU, M.; SINGH, S. “*Novelty Detection: A Review - Part 1: Statistical Approaches*”. Signal Processing, Amsterdam, The Netherlands: Elsevier North-Holland, volume 83, issue 12, 2003. pp. 2481-2497.

MINI, R. A. F.; LOUREIRO, A. A. F. “*The Energy Role in the Design of Applications for Wireless Sensor Networks*”. In: Symposium on Applied Computing (SAC), 23, 2008, Fortaleza, Brasil. Tutorial... [Fortaleza]: [ACM], 2008.

MONARD, C. M.; BARANAUSKAS, J. A. Conceitos sobre Aprendizado de Máquina. In: REZENDE, S. O. (Org.) “*Sistemas Inteligentes: Fundamentos e Aplicações*”. 1. ed. Barueri, SP: Manole, 2003. cap. 4, pp. 89-114.

MUKHERJEE, P.; SEN, S. “*Detecting Malicious Sensor Nodes from Learned Data Patterns*”. In: International Workshop on Agent Technology for Sensor Networks (ATSN), 1, 2007, Honolulu, Hawaii. Proceedings... 2007.

NAKAMURA, E. F.; LOREIRO, A. A. F.; FRERY, A. C. "**Information Fusion for Wireless Sensor Networks: Methods, Models, and Classifications**". ACM Computing Surveys (CSUR), New York, USA: ACM, volume 39, issue 3, 2007.

PALPANAS, T.; PAPADOPOULOS, D.; KALOGERAKI, V.; GUNOPULOS, D. "**Distributed Deviation Detection in Sensor Networks**". ACM SIGMOD Record, New York, USA: ACM, volume 32, issue 4, 2003. pp. 77-82.

PLESS, V. G.; KARIM, T. A.; REZNIK, L. "**Time-Based Multi-Layer Perceptron for Novelty Detection in Sensor Networks**". In: International Conference on Machine Learning and Applications, 2004, Louisville, USA. Proceedings... [IEEE Computer Society], 2004. pp. 156-163.

REZNIK, L.; PLESS, V. G.; KARIM, T. A. "**Signal Change Detection in Sensor Networks with Artificial Neural Network Structure**". In: International Conference on Computational Intelligence for Homeland Security and Personal Safety (CIHSPS), 2005, Orlando, USA. Proceedings... [IEEE Computer Society], 2005. pp. 44-51.

RUSSELL, S.; NORVIG, P. "**Inteligência Artificial**". Rio de Janeiro, RJ: Elsevier, 2004.

SHEN, C.; SRISATHAPORNPHAT, C.; JAIKAEAO, C. "**Sensor Information Networking Architecture and Applications**". IEEE Personal Communications, IEEE Communications Society, volume 8, issue 4, 2001. pp. 52-59.

SHIH, E. *et al.* "**Physical Layer Driven Protocol and Algorithm Design for Energy-Efficient Wireless Sensor Networks**". In: International Conference on Mobile Computing and Networking (MOBICOM), 7, 2001, Rome, Italy. Proceedings... New York, USA: ACM, 2001. pp. 272-286.

SONG, H.; ZHU, S.; CAO, G. "**Attack-Resilient Time Synchronization for Wireless Sensor Network**". In: International Conference on Mobile Ad Hoc and Sensor Systems, 2005, Washington, USA. Proceedings... [IEEE Computer Society], 2005.

SPINOSA, E. J.; CARVALHO, A. P. L. F.; GAMA, J. "**OLINDDA: A Cluster-based Approach for Detecting Novelty and Concept Drift in Data Streams**". In: Symposium on Applied Computing (SAC), 22, 2007, Seoul, Korea. Proceedings... New York, USA: ACM, 2007. pp. 448-452.

SUBRAMANIAN, S.; PALPANAS, T.; PAPADOULOS, D.; KALOGERAKI, V.; GUNOPULOS, D. “*Online Outlier Detection in Sensor Data Using Non-Parametric Models*”. In: International Conference on Very Large Data Bases (VLDB), 32, Seoul, Korea. Proceedings... VLDB Endowment, 2006. pp. 187-198. Disponível em: <<http://www.vldb.org/conf/2006/p187-subramaniam.pdf>>. Acesso em: 13 out. 2007.

TILAK, S.; ABU-GHAZALEH, N. B.; HEINZELMAN, W. “*A Taxonomy of Wireless Micro-Sensor Networks Model*”. Mobile Computing and Communications Review (SIGMOBILE), New York, USA: ACM, volume 6, issue 2, 2002. pp. 28-36.

TINYOS. 2004. “*Mission Statement*”. Disponível em: <<http://www.tinyos.net/special/mission>>. Acesso em: 08 fev. 2008.

WITTEN, I. H.; FRANK, E. “*Data Mining: Practical Machine Learning Tools and Techniques*”. 2. ed. [San Francisco, CA, EUA]: Elsevier, 2005.

WÜTHRICH, B.; CHO, V.; PUN, J.; ZHANG, J. Data Quality in Distributed Environments. In: KARGUPTA, H.; CHAN, P. (Eds.) “*Advances in Distributed and Parallel Knowledge Discovery*”. AAAI Press, 2000. cap. 11, pp. 295-316.

XIAO, X.; PENG, W.; HUNG, C.; LEE, W. “*Using SensorRanks for In-Network Detection of Faulty Readings in Wireless Sensor Networks*”. In: International Workshop on Data Engineering for Wireless and Mobile Access, 6, 2007, Beijing, China. Proceedings... New York. USA: ACM Press, 2007. pp. 1-8.

YAO, Y. GEHRKE, J. “*Query Processing for Sensor Networks*”. In: Conference Innovative Data Research (CIDR), 1, 2003. Asilomar, USA. Proceedings... VLDB Endowment, 2003. pp. 233-244. Disponível em: <<http://www.cidrdb.org/cidr2003/program/p21.pdf>>. Acesso em: 02 maio 2008.

ZHUANG, Y.; CHEN, L. “*In-Network Outlier Cleaning for Data Collection in Sensor Networks*”. In: International VLDB Workshop on Cleaning Databases, 1, 2006, Seoul, Korea. Proceedings... [VLDB Endowment], 2006. Disponível em: <http://pike.psu.edu/cleandb06/papers/CameraReady_124.pdf>. Acesso em: 13 out. 2007.

GLOSSÁRIO

<i>Broadcast</i>	<p>Tipo de comunicação em que os nós remetem suas informações aos seus vizinhos que, por sua vez, as retransmitem sucessivamente até alcançar o destino final. Os dados podem ser tratados durante o percurso, e.g., descartados ou agrupados, diminuindo a sobrecarga causada por essa abordagem que, apesar disso, torna a rede mais robusta em relação às mudanças dinâmicas de topologia. Nas RSSFs, encontram-se ainda as abordagens <i>unicast</i>, em que a comunicação ocorre diretamente com a estação-base ou um líder de <i>cluster</i> usando <i>unicast</i> (um para um), e a <i>multicast</i>, que procede entre todos os membros de um <i>cluster</i> (TILAK <i>et al.</i>, 2002, p. 31).</p>
Covariância	<p>A covariância amostral é dada pela seguinte fórmula: $\sum (o - \bar{o})(u - \bar{u}) / (n - 1)$. A soma dos produtos dos desvios de duas variáveis aleatórias em relação à respectiva média aritmética mede como as duas variáveis mudam conjuntamente (FREUND; SIMON, 2000, p. 323).</p>
Memória <i>Flash</i>	<p>Memória não-volátil, ou seja, mantida mesmo após a interrupção de energia elétrica. Diferentemente da ROM, essa memória pode ser alterada em blocos. A memória <i>flash</i> é usada, por exemplo, em câmeras digitais, cartões de memória, pendrives, palmtops, celulares, nós-sensores, entre outros.</p>
RAM	<p><i>Random Access Memory</i>. Memória de acesso aleatório, também conhecida como memória principal, onde qualquer informação pode ser alterada sem necessariamente ter que se alterar dados existentes anteriormente. Esse tipo de memória é volátil, i.e., seus dados são perdidos sempre que a energia é interrompida. Sua principal função é de conter as informações necessárias para o processador em um determinado momento.</p>
ROM	<p><i>Read-Only Memory</i>. Memória somente de leitura. Esse tipo de memória é não-volátil, i.e., seus dados não são perdidos quando a energia é interrompida. Em geral, programas críticos para o funcionamento do hardware são pré-gravados nesse tipo de memória (e.g., BIOS).</p>
SRAM	<p><i>Static RAM</i>. É um tipo de memória RAM (estática).</p>
Variável Aleatória	<p>Sejam E um experimento e S o espaço associado ao experimento, uma função X, que associe a cada elemento $s \in S$ um número real $X(s)$, é denominada Variável Aleatória. Na estatística, uma variável aleatória é de fato uma função e não uma variável. As variáveis aleatórias podem ser discretas ou contínuas (FONSECA; MARTINS, 1982, p. 25).</p>
Variável Aleatória Contínua	<p>Uma variável aleatória X é considerada contínua se o número de valores possíveis de X for um intervalo ou uma coleção de intervalos (FONSECA; MARTINS, 1982, p. 26).</p>
Variável Aleatória Discreta	<p>Uma variável aleatória X é considerada discreta se o número de valores possíveis de X for finito ou infinito enumerável (FONSECA; MARTINS, 1982, p. 26).</p>

Livros Grátis

(<http://www.livrosgratis.com.br>)

Milhares de Livros para Download:

[Baixar livros de Administração](#)

[Baixar livros de Agronomia](#)

[Baixar livros de Arquitetura](#)

[Baixar livros de Artes](#)

[Baixar livros de Astronomia](#)

[Baixar livros de Biologia Geral](#)

[Baixar livros de Ciência da Computação](#)

[Baixar livros de Ciência da Informação](#)

[Baixar livros de Ciência Política](#)

[Baixar livros de Ciências da Saúde](#)

[Baixar livros de Comunicação](#)

[Baixar livros do Conselho Nacional de Educação - CNE](#)

[Baixar livros de Defesa civil](#)

[Baixar livros de Direito](#)

[Baixar livros de Direitos humanos](#)

[Baixar livros de Economia](#)

[Baixar livros de Economia Doméstica](#)

[Baixar livros de Educação](#)

[Baixar livros de Educação - Trânsito](#)

[Baixar livros de Educação Física](#)

[Baixar livros de Engenharia Aeroespacial](#)

[Baixar livros de Farmácia](#)

[Baixar livros de Filosofia](#)

[Baixar livros de Física](#)

[Baixar livros de Geociências](#)

[Baixar livros de Geografia](#)

[Baixar livros de História](#)

[Baixar livros de Línguas](#)

[Baixar livros de Literatura](#)
[Baixar livros de Literatura de Cordel](#)
[Baixar livros de Literatura Infantil](#)
[Baixar livros de Matemática](#)
[Baixar livros de Medicina](#)
[Baixar livros de Medicina Veterinária](#)
[Baixar livros de Meio Ambiente](#)
[Baixar livros de Meteorologia](#)
[Baixar Monografias e TCC](#)
[Baixar livros Multidisciplinar](#)
[Baixar livros de Música](#)
[Baixar livros de Psicologia](#)
[Baixar livros de Química](#)
[Baixar livros de Saúde Coletiva](#)
[Baixar livros de Serviço Social](#)
[Baixar livros de Sociologia](#)
[Baixar livros de Teologia](#)
[Baixar livros de Trabalho](#)
[Baixar livros de Turismo](#)