

Universidade Federal da Paraíba
Centro de Informática
Programa de Pós-Graduação em Informática

Seleção de Sensores em Redes de Sensores Sem Fio Heterogêneas para Cidades Inteligentes

FRANCISCO DE ASSIS SIQUEIRA JÚNIOR

João Pessoa, Paraíba, Brasil

30 de Agosto de 2016

FRANCISCO DE ASSIS SIQUEIRA JÚNIOR

Seleção de Sensores em Redes de Sensores Sem Fio Heterogêneas para Cidades Inteligentes

Dissertação submetida à Coordenação do Curso de Pós-Graduação em Informática da Universidade Federal da Paraíba como parte dos requisitos necessários para obtenção do grau de Mestre em Informática.

Área de Concentração: Ciência da Computação

Linha de Pesquisa: Computação Distribuída

Prof. Dr. Fernando Menezes Matos

(Orientador)

Prof. Dr. Aldri Luiz dos Santos

(Co-Orientador)

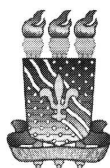
João Pessoa, Paraíba, Brasil

©Francisco de Assis Siqueira Júnior, 30 de Agosto de 2016

S618s Siqueira Júnior, Francisco de Assis.
 Seleção de sensores em redes de sensores sem fio
 heterogêneas para cidades inteligentes / Francisco de Assis
 Siqueira Júnior.- João Pessoa, 2016.
 81f. : il.
 Orientador: Fernando Menezes Matos
 Coorientador: Aldri Luiz dos Santos
 Dissertação (Mestrado) - UFPB/CI
 1. Informática. 2. Seleção dinâmica. 3. Redes de sensores
 sem fio heterogêneas. 4. Cidades inteligentes. 5. IoT.

UFPB/BC

CDU: 004(043)



UNIVERSIDADE FEDERAL DA PARAÍBA
CENTRO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA



Ata da Sessão Pública de Defesa de Dissertação de
 Mestrado de **FRANCISCO DE ASSIS SIQUEIRA JUNIOR**,
 candidato ao título de Mestre em Informática na Área de
 Sistemas de Computação, realizada em 30 de agosto de
 2016.

1 Aos trinta dias do mês de agosto, do ano de dois mil e dezesseis, às nove horas, no Centro
 2 de Informática da Universidade Federal da Paraíba, em Mangabeira, reuniram-se os
 3 membros da Banca Examinadora constituída para julgar o Trabalho Final do Sr.
 4 FRANCISCO DE ASSIS SIQUEIRA JUNIOR, vinculado a esta Universidade sob a matrícula
 5 nº 2014108467, candidato ao grau de Mestre em Informática, na área de "Sistemas de
 6 Computação", na linha de pesquisa "*Computação Distribuída*", do Programa de Pós-
 7 Graduação em Informática, da Universidade Federal da Paraíba. A comissão examinadora
 8 foi composta pelos professores: Fernando Menezes Matos, Orientador e Presidente da
 9 Banca, Iguatemi Eduardo da Fonseca, Examinador Interno ao Programa, e Luiz Affonso
 10 Henderson Guedes de Oliveira, Examinador Externo à Instituição. Dando início aos
 11 trabalhos, o Presidente da Banca, cumprimentou os presentes, comunicou aos mesmos a
 12 finalidade da reunião e passou a palavra ao candidato para que o mesmo fizesse a
 13 exposição oral do trabalho de dissertação intitulado "Seleção de Sensores em Redes de
 14 Sensores Sem Fio Heterogêneas para Cidades Inteligentes". Concluída a exposição, o
 15 candidato foi arguido pela Banca Examinadora que emitiu o seguinte parecer: "*aprovado*". Do
 16 ocorrido, eu, Claurton de Albuquerque Siebra, Coordenador do Programa de Pós-
 17 Graduação em Informática, lavrei a presente ata que vai assinada por mim e pelos
 18 membros da Banca Examinadora. João Pessoa, 30 de agosto de 2016.

Prof. Dr. Claurton de Albuquerque Siebra

Claurton de Albuquerque Siebra
 Coordenador do Programa de
 Pós-Graduação em Informática
 SIAPE 1723491

Prof. Dr. Fernando Menezes Matos
 Orientador (PPGI-UFPB)

Prof. Dr. Iguatemi Eduardo da Fonseca
 Examinador interno ao Programa (PPGI-UFPB)

Prof. Dr. Luiz Affonso Henderson Guedes de Oliveira
 Examinador externo à Instituição (UFRN)

Aos meus pais Francisco Siqueira e Graciliana Siqueira. À minha avó Marta Vieira (In memoriam) e minha tia Severina Ramos (In memoriam).

Agradecimentos

Primeiramente agradeço ao Senhor Deus por todos os livramentos e graças concedidas durante toda a minha vida, mantendo-me firme e confiante mesmo nas horas difíceis.

Agradeço aos meus pais Francisco Siqueira e Graciliana Siqueira por toda dedicação, amor, incentivo e investimento que fizeram em mim, acreditando sempre no meu potencial e apoiando-me minhas decisões, mesmo sabendo que estas me levariam para longe de casa.

Em especial, agradeço Dr. Fernando Matos que com toda a sua competência e profissionalismo coordenou a produção do presente trabalho. Agradeço-lhe por acreditar no meu potencial e também pelas inúmeras contribuições e ensinamentos dadas durante minha jornada no PPGI.

Ao Dr. Aldri Santos por todo o conhecimento dado e pelas contribuições realizadas durante este tempo.

Ao meu irmão Diego Siqueira e a minha cunhada Teresa Cristina por todas as vezes que substituíram-me no trabalho para que eu pudesse dar continuidade neste trabalho.

À minha namorada Munirez que durante o tempo de Mestrado compartilhou comigo de todas as renúncias necessárias a conclusão deste trabalho. Obrigado por tudo que fez e faz por mim durante esses 5 anos de namoro, espero conseguir retribuí-la.

Ao amigo Everaldo Andrade pela grande ajuda no projeto e pelas inúmeras noites de estudos no LUMO.

Aos amigos, Anderson Alves, Carlos Alberto, Daniel Deyson, Diego Sousa, Djanino Fernandes, Jackson, Jefferson e Ozonias Junior agradeço pelos muitos favores a mim concedidos.

À Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (Capes), pelo apoio financeiro.

Enfim, agradeço a todos que, de forma direta ou indireta, contribuíram para a minha formação mas que no momento não tive oportunidade de citar.

Resumo

Apesar de ser uma tecnologia já consolidada, pesquisas em Redes de Sensores Sem Fio (RSSFs) vêm passando por desafios devido à sua intrínseca relação com os paradigmas de Cidades Inteligentes e Internet das Coisas (*Internet of Things* - IoT). Tradicionalmente, as RSSFs são implantadas e utilizadas para atender a uma única aplicação e são compostas de nós sensores com as mesmas características físicas. Contudo, ao se considerar ambientes inteligentes, onde vários objetos podem coletar dados, desencadear ações e interagir entre si de forma a prover serviços sofisticados, é fácil perceber a heterogeneidade dos tipos de aplicações que podem realizar requisições em tais ambientes. Além disso, é real a possibilidade de interação entre diferentes tipos de equipamentos, como por exemplo, sensores com diferentes características físicas, criando assim uma RSSF Heterogênea (RSSFH). Por este motivo, este trabalho objetiva contribuir para a gerência de RSSFHs em ambientes de Cidades Inteligentes ao propor uma abordagem para seleção dinâmica e sob-demanda de sensores, levando em consideração as características físicas dos sensores e as demandas das aplicações requisitantes. Esta abordagem foi utilizada no *framework* POLicy-based Sensor SElection in Heterogeneous Wireless Sensor Network (POSSE-HWSN), o qual foi implementado para validar a proposta. Testes realizados demonstram que através da utilização de POSSE-HWSN foi possível escolher os sensores com os recursos de hardware mais adequados para atender a cada aplicação, melhorando assim a eficácia da seleção.

Palavras-chave: Seleção dinâmica, Redes de Sensores Sem Fio Heterogêneas, Cidades Inteligentes, IoT

Abstract

Despite being a well-established technology, research in Wireless Sensor Networks (WSNs) are facing challenges due to its intrinsic connection to the paradigms of Smart Cities and IoT (Internet of Things - IoT). Traditionally, WSNs are deployed to meet the needs of a single application and are composed of sensors with the same physical characteristics. However, when considering smart environments, where several objects can collect data, trigger actions and interact with each other in order to provide sophisticated services, it is easy to see the diversity of the applications that can perform requests in such environments. It is also possible that different types of equipment interact with each other, such as sensors with different physical characteristics, thus creating a Heterogeneous WSN (HWSN). For this reason, this work aims to contribute to the management of HWSNs in Smart Cities environments by proposing an approach for dynamic and on-demand selection of sensors. This selection is performed by taking into account the physical characteristics of the sensors and the applications' demands. This approach has been integrated into the Policy-based Sensor SElection in Heterogeneous Wireless Sensor Network framework (POSSE-HWSN), which was implemented in order validate the proposal. Tests have shown that by using POSSE-HWSN it was possible to select the sensors with the most suitable hardware features to meet every application needs, thus improving the selection efficacy.

Keywords: Dynamic Selection, Heterogeneous Wireless Sensor Networks, Smart Cities, IoT.

Sumário

1	Introdução	17
1.1	Motivação	17
1.2	Objetivos e Contribuições	22
1.3	Estrutura da Dissertação	23
2	Fundamentação Teórica	24
2.1	Redes de Sensores sem Fio (RSSF)	24
2.2	Uso de Políticas em RSSF	27
2.3	Redes Definidas por Software	29
2.4	IoT e Cidades Inteligentes	32
2.5	Considerações	33
3	Estado da Arte	34
3.1	Alocação de sensores em RSSF	34
3.2	Gerência de RSSF utilizando políticas	38
3.3	Considerações	40
4	Framework POSSE-HWSN - <i>POL</i>icy-based <i>SE</i>nsor <i>SE</i>lection in HWSN	41
4.1	Arquitetura de POSSE-HWSN	41
4.2	Funcionamento de POSSE-HWSN	43
4.2.1	Visão global da rede	45
4.2.2	Políticas de seleção	46
4.2.3	Seleção de nós sensores	49
4.3	Considerações	52

5	Protótipo e Resultados Experimentais	53
5.1	Implementação do protótipo	53
5.2	Ambiente de simulação	56
5.3	Cenários de testes e resultados	57
5.3.1	Overhead de POSSE-HWSN	58
5.3.2	Testes de validação	59
5.3.3	Testes de desempenho	61
5.3.3.1	Cenário 1: aplicações que requisitam quantidades iguais de nós sensores	62
5.3.3.2	Cenário 2: aplicações que requisitam quantidades aleató- rias de nós sensores	67
5.4	Considerações	74
6	Conclusão e Trabalhos Futuros	75
	Referências Bibliográficas	82

Lista de acrônimos e siglas

API : *Application Program Interface*

BAN : *Body Area Networks*

CH : *Cluster Head*

CM : *Configuration Manager*

CPU : *Central Processing Unit*

DCLBA : *Dynamic CPU Load Balanced Allocation*

EDDEEC : *Enhanced Develop Distributed Energy Efficient Clustering*

FK : *Foreign Key*

GECSA : *Game Theory Based Energy Efficient Cluster-Head Selection Approach*

IDS : *Intrusion Detection System*

IoT : *Internet of Things*

MAC : *Medium Access Control*

MEF : *Maximum Energy First*

NDM : *Networking Data Manager*

P2P : *Peer-to-peer*

PK : *Primary Key*

PProb : *Porcentagem de Probabilidade*

QOS : *Quality of Service*

RFID : *Radio-Frequency Identification*

RSSF : *Redes de Sensores Sem Fio*

RSSFH : *Redes de Sensores Sem Fio Heterogênea*

SANCHSEN : *reSource AlloCation in Heterogeneous SEnsor Networks*

SDN : *Software Defined Network*

SPNE : *Perfect Nash Equilibrium*

TDM : *Take Decision Manager*

TIC : *Tecnologias da Informação e Comunicação*

Lista de Figuras

1.1	Campus inteligente	20
2.1	Visão geral de uma RSSF	26
2.2	Exemplo de política simples	28
2.3	Representação dos planos de gerência no paradigma SDN.	30
4.1	Arquitetura do <i>Framework</i> POSSE-HWSN	42
4.2	Controlador do <i>Framework</i> POSSE-WSN	42
4.3	Diagrama de sequência do <i>Framework</i> POSSE-WSN	44
4.4	Estrutura da base de dados	45
4.5	Escolha de políticas	48
4.6	Fluxograma da seleção dos nós sensores	51
5.1	Módulos do Castalia. Adaptado de (BOULIS, 2011)	54
5.2	Modelo lógico dos componentes de nós sensores (BOULIS, 2011)	55
5.3	Linha do tempo para o cenário utilizado	60
5.4	Atraso médio na seleção de nós sensores	61
5.5	Eficácia na seleção de nós sensores	62
5.6	Atraso médio do cenário 1 com <i>Prob</i> variável para uma demanda	64
5.7	Atraso médio do cenário 1 com <i>Prob</i> variável para duas demandas	65
5.8	Percentual de eficácia do cenário 1 com <i>Prob</i> variável para uma demanda	67
5.9	Percentual de eficácia do cenário 1 com <i>Prob</i> variável para duas demandas	68
5.10	Atraso médio do cenário 2 com <i>Prob</i> variável para uma demanda	70
5.11	Atraso médio do cenário 2 com <i>Prob</i> variável para duas demandas	71
5.12	Percentual de eficácia do cenário 2 com <i>Prob</i> variável para uma demanda	72

5.13	Percentual de eficácia do cenário 2 com <i>Prob</i> variável para duas demandas .	73
------	---	----

Lista de Tabelas

4.1	Parâmetros da requisição das aplicações	47
5.1	Recursos de <i>Hardware</i> dos nós sensores utilizados na simulação	57
5.2	Comparativo de mensagens trocadas	59
5.3	Características das aplicações requisitantes	60
5.4	Percentual de aplicações atendidas	69

Lista de Algoritmos

1	ESCOLHA DOS RECURSOS DOS NÓS SENSORES	50
---	---	----

Capítulo 1

Introdução

1.1 Motivação

As Redes de Sensores Sem Fio (RSSF) são uma das tecnologias emergentes capazes de mudar o modo de vida das pessoas, de acordo com a revista *MIT Enterprise Technology Review* (MIT, 2003). Neste sentido, as RSSF passaram a serem utilizadas de forma mais frequente. RSSF podem ser encontradas nos mais diversos aspectos do cotidiano, desde casas, hospitais, fábricas, florestas e até estradas são exemplos de possíveis ambientes que utilizam RSSF. Apesar de ser uma tecnologia com inúmeras aplicações bem-sucedidas, as RSSF possuem limitações que são acarretadas, principalmente, devido às limitações de hardware existentes nos nós sensores, tais como baixo poder de processamento e armazenamento, baixa taxa de transmissão quando comparada à redes cabeadas e capacidade energética limitada por dependerem de baterias. Tais limitações faz com que seja necessário o emprego de técnicas de gerenciamento de recursos para que os nós sensores possam atuar da maneira mais adequada possível por longos períodos de tempo (ALAJLAN; ELLEITHY, 2014).

Outro aspecto característico desta tecnologia é que uma RSSF usualmente é projetada e implantada em um ambiente para atender a uma única aplicação. Isto faz com que muitas vezes, mais de uma RSSF seja instalada no mesmo local, sendo cada uma responsável pelo sensoriamento de um ou poucos fenômenos físicos (luminosidade, pressão, movimento, etc.) e do envio desses dados para a sua aplicação requisitante correspondente (LUO; TAN; QUEK, 2012). Finalmente, RSSF são majoritariamente compostas de nós sensores com as mesmas características físicas, tais como quantidade e consumo de baterias, poder de pro-

cessamento, memória e alcance de transmissão (LI et al., 2014). Estes dois aspectos podem ser justificados pela necessidade de simplificar a gerência da rede. A utilização de nós sensores com diferentes características e a possibilidade de atender a aplicações com demandas diversas por uma mesma RSSF contribuem para aumentar a complexidade da gerência desta rede.

Nos últimos anos, dois paradigmas vêm ganhando importância tanto na academia quanto na indústria. Um deles é o de Cidades Inteligentes (SILVA et al., 2013). Neste paradigma, as Tecnologias de Informação e Comunicação (TICs) são utilizadas para melhorar diversos aspectos na vida das pessoas, tais como saúde, transporte, comércio, meio-ambiente, educação, agricultura dentre outros. Já o paradigma de Internet das Coisas (*Internet of Things* - IoT) busca a integração de objetos do mundo real à Internet. Estes objetos podem monitorar e/ou controlar diversos aspectos em vários ambientes. Assim, qualquer "coisa" poderia ser um objeto em potencial no contexto deste paradigma. A união destes dois paradigmas surge como uma promessa para conectar tudo a todos, possibilitando a criação de infraestruturas inteligentes que facilitam a oferta dos mais variados serviços a qualquer momento e para qualquer pessoa.

Desta forma, as RSSF podem ser consideradas como um dos alicerces para a criação destas infraestruturas inteligentes, uma vez que os ambientes precisam ser monitorados e controlados para permitir a realização de cenários tão complexos. Esta constatação vai de encontro aos dois aspectos característicos das RSSF mencionados anteriormente (uma RSSF atende a uma única aplicação e é composta por nós sensores com as mesmas características físicas). Em ambientes extensos, como um campus universitário, bairros, empresas ou mesmo cidades, a quantidade de aplicações requisitantes pode ser muito grande. Assim, não é razoável assumir que cada aplicação será atendida por uma RSSF dedicada, bem como não é realista a utilização de um mesmo padrão de nós sensores em tais cenários. Por esta razão, a implantação e utilização de RSSF Heterogêneas (RSSFH) torna-se necessária (ZHI-YAN; JIAN-ZHEN, 2012).

Em RSSFH, nós sensores com diferentes características físicas interagem para processar requisições de aplicações que podem apresentar diferentes demandas. Tal heterogeneidade de aplicações e nós sensores, acaba por dificultar ainda mais a gerência de recursos, a qual já é uma tarefa complexa em RSSF tradicionais. Por exemplo, para atender a um pedido de

uma aplicação que requer um pré-processamento dos dados coletados antes de enviar para a aplicação, é preferível escolher os nós sensores com menor consumo de energia ou aqueles com melhor poder de processamento? Esta situação ilustra um cenário simples em que diferentes decisões podem ser tomadas para selecionar os dispositivos mais apropriados a fim de alcançar um objetivo (estender o tempo de vida da rede ou otimizar o pré-processamento dos dados).

As decisões a serem tomadas acima ficam ainda mais complexas ao considerar ambientes de Cidades Inteligentes. Em tais ambientes realmente inteligentes e que utilizam o paradigma IoT de forma substancial, é fácil perceber que o número de aplicações que podem fazer requisições é imenso. Assim sendo, é possível chegar a duas conclusões:

- As aplicações tanto podem ser pré-definidas quanto podem ser *on-the-fly*. Aplicações pré-definidas são aquelas as quais já se sabe de antemão que vão existir no ambiente inteligente. Assim, o ambiente já pode ser preparado para atender a tais aplicações. Em contrapartida, aplicações *on-the-fly* são aquelas que não estavam previstas no ambiente, ou seja, podem requisitar ao ambiente sem que o mesmo esteja preparado para atendê-las.
- Levando em consideração as aplicações *on-the-fly* mencionadas acima, é razoável aceitar que os requisitos demandados pelas aplicações podem variar enormemente. Assim sendo, um ambiente inteligente não tem como prever qual o tipo de demanda uma aplicação *on-the-fly* terá.

Além disso, nestes ambientes que são preponderantemente urbanos, são grandes as chances dos nós sensores não dependerem de bateria por serem alimentados por fonte de energia elétrica convencional. Isto faz com que o tempo de vida da rede seja uma preocupação a menos. Consequentemente, não faz sentido considerar apenas a energia atual e/ou consumo energético como fator de decisão na escolha dos nós sensores, no entanto, é utilizado pela esmagadora maioria das propostas que, em sua grande maioria, são aplicadas em ambientes hostis e que necessitam da utilização de baterias.

Por estes motivos, é de extrema importância que ambientes inteligentes consigam selecionar de forma dinâmica e sob demanda os equipamentos com diferentes características para atender a estas aplicações *on-the-fly*, considerando não apenas a energia, mas também outros

parâmetros físicos dos nós sensores. A Figura 1.1 ilustra um cenário simples, porém factível, de um campus inteligente no qual podem haver vários nós sensores de diferentes tipos. Para efeitos de simplificação, considera-se qualquer equipamento utilizado para monitorar/sensoriar um fenômeno físico como um nó sensor, o qual é composto por processador, memória, equipamento de transmissão (rádio) e um ou mais módulos de sensoriamento (sensor).

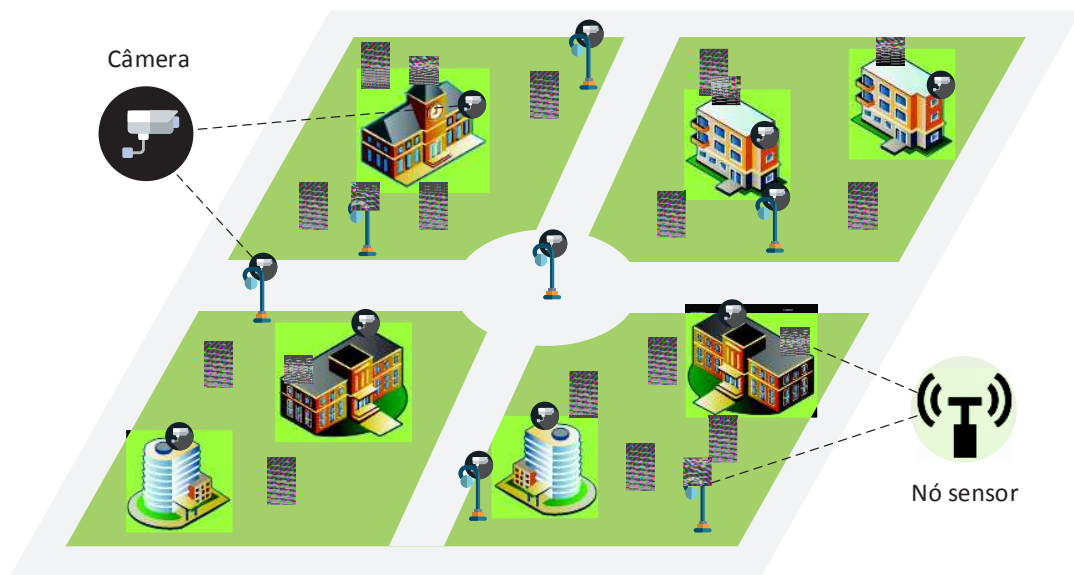


Figura 1.1: Campus inteligente

Neste ambiente ilustrado na Figura 1.1, há câmeras de vigilância por todo o campus que filmam em baixa resolução (para não saturar a rede). Há também uma RSSFH composta por diversos nós sensores capazes de sensoriar os mais variados fenômenos do ambiente. Suponha que a RSSFH detectou a atuação indesejada de alguns indivíduos num dos ambientes, por questões de segurança, a RSSFH solicita que as câmeras de segurança devam filmar em alta resolução e transmitir as imagens em tempo real para a equipe de segurança. Para isso, é necessário escolher as câmeras que possuam um rádio com capacidade de transmissão mais elevada. Ou seja, o novo evento desencadeou uma nova demanda no ambiente. Outro cenário possível de ocorrer neste ambiente é a utilização das câmeras para realizar processamento de imagens. Caso aconteça um assalto nas dependências do campus e os assaltantes fujam em um carro, é desejável que as câmeras de vigilância consigam reconhecer a placa do veículo. Para isto, contudo, é necessário poder computacional, o que nem todas as câmeras possuem. Finalmente, que os nós sensores presentes no ambiente, em paralelo com as demandas existentes, são requisitados para medir a radiação ultravioleta que incide no ambi-

ente e enviar estas informações em tempo real. Por causa disto, não é requisito essencial que os nós sensores possuam alta capacidade de armazenamento, já que não retêm os dados por longo período. Contudo, estes nós também possuem nós sensores capazes de aferir a qualidade do ar. Devido a um fenômeno meteorológico qualquer, foi verificado que a qualidade do ar no campus degradou. Por causa disto, pesquisadores necessitam medir a concentração de partículas nocivas ao longo de um período de tempo e enviar uma média ao final do dia. Desta forma, é necessário que sejam escolhidos nós sensores com uma capacidade de armazenamento (memória) maior para guardar e enviar ao final de um período de tempo os dados coletados. Todos estes exemplos mencionados mostram como variadas aplicações podem exigir diferentes recursos de hardware para desenvolver suas funcionalidades de acordo com eventos que não podem ser previstos.

Baseado nestas premissas, esta dissertação propõe o *framework POLicy-based Sensor SElection in Heterogeneous Wireless Sensor Network* (POSSE-HWSN), o qual é baseado em políticas para seleção dinâmica e sob demanda de nós sensores em RSSFH. Através deste *framework* é possível a utilização de diferentes configurações de sensoriamento para atender a diferentes aplicações. Tais configurações são formadas utilizando, tanto as demandas das aplicações quanto as características físicas dos diferentes nós sensores. A diversidade de configurações de sensoriamento que pode ocorrer devido à heterogeneidade dos cenários, faz com que a utilização de políticas seja uma escolha adequada. Políticas facilitam tarefas de gerência ao definir regras que governam o comportamento do sistema de forma dinâmica e sem a necessidade de reimplantação ou reinicialização deste sistema caso alterações ocorram ou novas regras sejam definidas (FIGUEIREDO et al., 2005). Além disso, o uso de políticas, abre o caminho para a utilização de mecanismos mais sofisticados para seu armazenamento, como em (GONÇALVES; SANTOS; HARA, 2014).

POSSE-HWSN também utiliza a ideia de um software controlador advinda do paradigma de Redes Definidas por Software (*Software Defined Networks* - SDN). Em SDN, o controlador possui uma visão geral e centralizada da rede, o que o capacita a tomar as decisões de encaminhamento de tráfego. O controlador utilizado em POSSE-HWSN também possui esta visão geral e centralizada de todos os nós sensores da rede, conseguindo assim selecionar os mais adequados de acordo com as demandas das aplicações. Finalmente, para auxiliar na seleção de nós sensores, são utilizados os parâmetros de poder de processamento, ob-

tidos através da *Central Processing Unit* (CPU), capacidade de armazenamento (memória) e capacidade de transmissão. Testes realizados no simulador Castalia mostram a eficácia de POSSE-HWSN ao atender requisições de diferentes aplicações e levando em consideração as diferentes características físicas dos nós sensores. Esta eficácia diz respeito à quão adequados são os nós sensores selecionados para atender à demanda de cada aplicação.

1.2 Objetivos e Contribuições

Esta Dissertação tem como objetivo principal contribuir para o processo de gerência de RS-SFH em ambientes de Cidades Inteligentes, ao propor um *framework* para seleção dinâmica e sob demanda de nós sensores. Esta seleção é regida por políticas, baseia-se nas demandas das aplicações requisitantes e nos recursos de hardware disponíveis em cada sensor e tem por objetivo selecionar os nós sensores que possuem os recursos de hardware mais adequados para cada aplicação requisitante. As contribuições desta dissertação podem ser expostas como:

- **Abordagem da solução:** A abordagem proposta diferenciou-se da literatura ao utilizar parâmetros diferentes de energia e/ou consumo energético no desenvolvimento da solução. Ao trabalhar no contexto de Cidades Inteligentes, é necessário considerar a grande diversidade de aplicações e dispositivos. Por isso, ao propor uma solução que leve em consideração tanto as demandas das aplicações quanto outras características físicas dos nós sensores, como capacidade de armazenamento e transmissão e poder de processamento, o atendimento a tais aplicações tende a ser mais eficaz. Além disso, esta seleção é feita de maneira dinâmica e sob demanda, o que também condiz com os ambientes de Cidades Inteligentes, uma vez que é difícil saber previamente todos os tipos de aplicações que podem requisitar nestes ambientes. Finalmente, o processo de seleção busca escolher aqueles nós sensores que podem atender de forma mais eficaz cada aplicação, ou seja, não se limita apenas a escolher um nó sensor que consiga atender à requisição, mas que consiga atender de forma mais satisfatória.
- **Proposta do *framework* POSSE-HWSN:** Como forma de viabilizar a solução, foi proposto o *framework* POSSE-HWSN para realizar a seleção dinâmica e sob demanda

dos nós sensores. POSSE-HWSN utiliza políticas para determinar qual recurso de hardware deve ser levado em consideração em cada sensor, de forma a melhor atender as aplicações requisitantes. A utilização de políticas confere flexibilidade na seleção, uma vez que é possível alterar, adicionar ou remover políticas sem que seja necessário a reinstalação ou reinicialização do sistema. POSSE-HWSN também utiliza um controlador que possui uma visão geral e centralizada dos nós sensores que fazem parte da RSSF, baseando-se na ideia do paradigma SDN. Tal característica é essencial na tomada de decisão de quais nós sensores escolher.

- **Implementação e análise do *framework* POSSE-HWSN:** Um protótipo de POSSE-HWSN foi implementado para validar a proposta. Este protótipo inclui o módulo controlador, que interage com o simulador Castalia, no qual também foram incluídas rotinas para a definição e configuração do ambiente de nós sensores. Além disso, também foram realizados testes experimentais para avaliar o desempenho do *framework*.

1.3 Estrutura da Dissertação

O restante da dissertação está organizado da seguinte forma: O Capítulo 2 discorre sobre os fundamentos teóricos necessários ao entendimento do trabalho. O Capítulo 3 apresenta uma revisão da literatura acerca do contexto desta dissertação. O Capítulo 4 detalha o *framework* proposto para a seleção dinâmica e sob demanda dos nós sensores em RSSFH. No capítulo 5 são apresentados detalhes da implementação do protótipo utilizado para validar o *framework* proposto, além disso, neste capítulo são apresentados os cenários de testes utilizados e os resultados alcançados. Por fim, o Capítulo 6 apresenta as conclusões e delinea propostas de continuidade da dissertação.

Capítulo 2

Fundamentação Teórica

Este capítulo tem por objetivo demonstrar os conceitos fundamentais para a compreensão desta dissertação, dentre eles: Redes de Sensores sem Fio e seus tipos emergentes; Redes Definidas por Software (Software Defined Network - SDN), Cidades inteligentes e IoT.

2.1 Redes de Sensores sem Fio (RSSF)

RSSF é uma tecnologia de sensoriamento distribuído composta por diversos dispositivos de baixo custo denominados de nós sensores. Tais nós sensores são instalados em ambientes no qual se deseja monitorar, tendo como principal objetivo a obtenção de informações acerca de atributos físicos do ambiente, como por exemplo, temperatura, movimento, luminosidade, etc. De acordo com (AKYILDIZ et al., 2002), um nó sensor típico é composto por quatro módulos: módulo de sensoriamento, módulo de processamento, módulo de comunicação e módulo de energia.

- **Módulo de processamento:** É o responsável pela execução das atividades necessárias para processar e disseminar de forma eficiente os dados coletados. Tipicamente este módulo está associado a um sistema de armazenamento de dados (Memória RAM ou Flash) que trabalham conjuntamente para melhorar a sua utilização.
- **Módulo de comunicação:** Possui um componente de transmissão denominado de unidade de rádio. Este módulo tem por objetivo realizar a transmissão das informações coletadas pelo próprio nó sensor ou recebidas pelos nós sensores vizinhos. Neste mó-

dulos são executados protocolos de roteamento e de acesso ao meio para gerenciar a transmissão entre nós sensores vizinhos.

- **Módulo de sensoriamento:** É o módulo responsável pela detecção dos diversos fenômenos do ambiente, tais como temperatura, pressão e umidade, dentre outros. Em um nó sensor, pode haver um ou vários componentes de sensoriamento (sensores), cada um responsável por sensoriar um fenômeno físico diferente.
- **Módulo de energia:** O funcionamento de nós sensores está associado à disponibilidade de energia. Este módulo é composto pelas unidades de energia disponíveis nos nós sensores, dentre elas, baterias, células solares, dispositivos para obtenção de energia eólica, conectores para alimentação através da rede elétrica, dentre outros. O módulo de energia é responsável pela energização dos componentes do nó sensor necessária para possibilitar seu funcionamento.

A Figura 2.1 apresenta uma visão geral da organização de uma RSSF. Esta composta por vários nós sensores que são posicionados em um ambiente e têm por principal função medir (monitorar) seus parâmetros físicos. Um dos nós sensores, chamado de nó sorvedouro (*Sink Node*) possui uma função específica de coletar todos os dados provenientes dos demais nós sensores e encaminhá-los para uma entidade externa à rede, como por exemplo, um servidor. Para repassar os dados, os nós sensores podem comunicar-se diretamente com o nó sorvedouro, ou podem utilizar outros nós sensores como intermediários, formando assim uma rota de transmissão.

As RSSF são aplicadas nos mais diversos ambientes, desde casas inteligentes, pecuária, agricultura, estudo e monitoramento de fenômenos climáticos, ambientes militares, detecção de intrusão, dentre outros. Em geral, os nós sensores necessitam trabalhar por longos períodos de tempo de forma autônoma. Porém, devido às suas dimensões reduzidas, algumas limitações de *hardware* passaram a ser intrínsecas à arquitetura, como por exemplo, poder de processamento e armazenamento limitados e, quando alimentados por baterias, fonte de energia limitada (AKYILDIZ et al., 2002).

Há vários aspectos que diferem as RSSF das redes sem fio, tais como os recursos de hardware disponíveis. Em uma rede sem fio tradicional – na grande maioria dos casos, há equipamentos que dispõem de *hardwares* potentes ao ponto de suportarem grandes quantida-

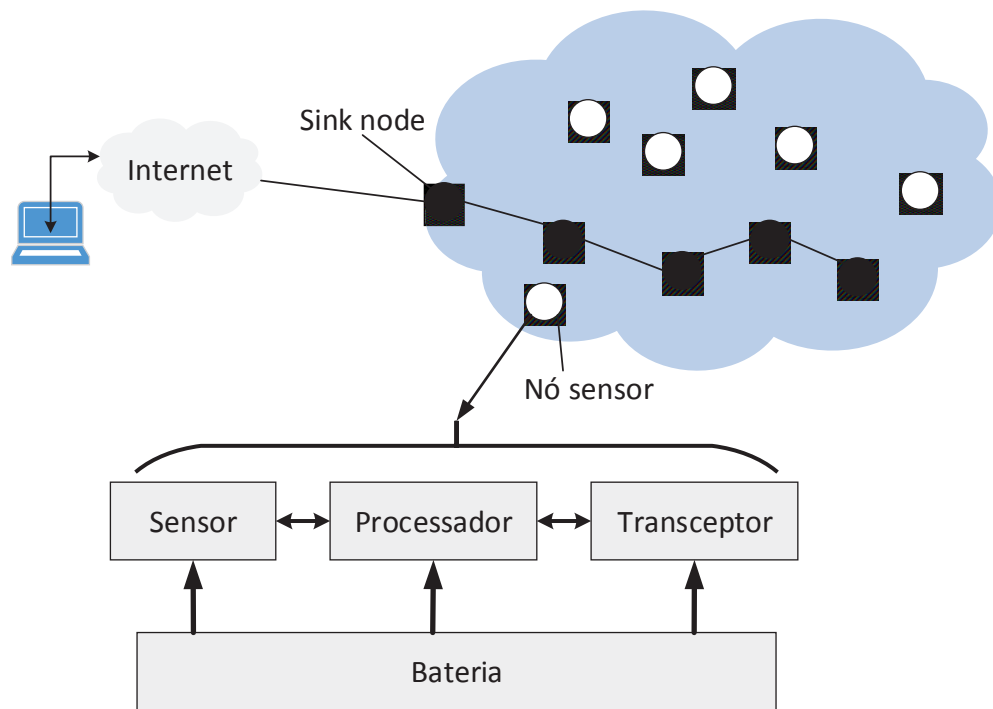


Figura 2.1: Visão geral de uma RSSF

des de processamento. Desta forma, é aconselhável que haja gerenciamento do hardware dos nós sensores para que estes seja capaz de desempenhar suas funcionalidades sem que haja prejuízo devido às limitações. Com o avanço das pesquisas na área e impulsionado principalmente pelo crescimento dos paradigmas IoT e Cidades Inteligentes, algumas derivações de RSSF começaram a ganhar visibilidade, as quais são: RSSF Heterogêneas (RSSFH) e as RSSF compartilhadas (AKBAR; IQBAL; KIM, 2006).

Tradicionalmente, as RSSF são projetadas para atender a uma única aplicação e, na maioria das vezes seu funcionamento depende de protocolos proprietários. Contudo, com o crescente avanço das pesquisas na área e impulsionado principalmente pelo crescimento dos paradigmas IoT e Cidades Inteligentes, no qual há necessidade de utilização de diversas aplicações em um mesmo ambiente, a criação de uma RSSF para cada aplicação tornou-se uma ideia obsoleta, com isso algumas derivações de RSSF começaram a ganhar visibilidade, as quais são: RSSF Heterogêneas (RSSFH) e as RSSF compartilhadas (AKBAR; IQBAL; KIM, 2006).

Em um ambiente de cidades inteligentes e IoT, para uma RSSF ser efetivamente funcional é necessário que ela possa atender aos mais diversos requisitos oriundos de múltiplas

aplicações (ROY; RAJAMANI; JULIEN, 2010), caso contrário em um ambiente “completamente” inteligente, seria necessária a instalação de várias RSSF para que fosse possível utilizar todos os serviços que este ambiente fosse capaz de oferecer. No entanto, esta arquitetura tornaria a ideia obsoleta do ponto de vista financeiro e gerencial (CHEN; KUEHNE; KLEIN, 2012).

A outra derivação de RSSF que vem chamando cada vez mais atenção são as RSSFH. Ao se instalar uma RSSF, é de praxe utilizar os mesmos tipos de nós sensores, o ponto o gerenciamento da rede. Contudo, nas RSSFH os nós sensores podem ser de diferentes tipos e possuírem componentes de *hardware* com características diferenciadas, tais como alcance de transmissão de rádio, poder de processamento, carga máxima e consumo de energia, dentre outros. Tais redes passaram a ter maior visibilidade devido à sua flexibilidade de equipamentos, permitindo por exemplo, que o projetista da rede não fique atrelado aos nós sensores de apenas um único fabricante, ou até mesmo que possam ser utilizados nós sensores mais adequados para um determinado ambiente, sem que haja incompatibilidade com os demais (PAWGASAME, 2016).

2.2 Uso de Políticas em RSSF

Arquiteturas baseadas em políticas têm sido amplamente difundidas, principalmente pela possibilidade de flexibilização no gerenciamento. Políticas podem ser especificadas como sendo um conjunto de regras que regem a configuração ou gestão do comportamento de um sistema, permitindo que modificações possam ser realizadas em tempo de execução sem necessidade de alteração de sua implementação (DAMIANOU et al., 2001).

De maneira geral, as políticas estabelecem uma alternativa para formalizar as ações que serão realizadas entre os elementos de rede caso determinado evento ocorra. Assim sendo, o método de funcionamento pode ser alterado devido a uma nova meta a ser atingida ou uma situação imprevisível, por exemplo.

A Figura 2.2 apresenta um exemplo simples de uma política. Políticas são compostas por uma ou mais regras, em que cada regra é formada por uma condição (simples ou composta). Caso uma condição seja satisfeita, então a ação correspondente é executada. Em políticas nas quais existem mais de uma regra contendo condições sobrepostas, as regras mais específicas

são executadas antes das regras mais gerais. Isto pode ser observado na Figura 2.2, na qual a primeira regra a ser verificada contém a condição mais específica (*if condition 1 AND condition 2*).

Normalmente, as políticas são estabelecidas por normas que especificam tanto o comportamento funcional quanto o não funcional dos sistemas devido a sua vasta aplicabilidade. O nível de abstração que uma política oferece a seus usuários pode variar bastante dependendo do objetivo a ser alcançado. Assim, é possível haver abstrações de alto nível, as quais são mais fáceis de entender por seres humanos, como por exemplo, o sistema de monitoramento de uma porta deve ser seguro; e abstrações de baixo nível, que geralmente regem ações que são interpretadas por sistemas subjacentes, como por exemplo, habilitar criptografia AES quando usuário for autenticado. As políticas podem ser utilizadas em muitos domínios, tais como segurança, Qualidade de Serviço (*Quality of Service* - QoS), gerenciamento de redes, ou até para gerir as preferências dos usuários (MATTHYS; JOOSEN, 2008).

A Figura 2.2 ilustra um exemplo simples de política. Esta possui duas regras, na qual cada regra é formada por uma condição que, quando satisfeita, desencadeia uma ação correspondente. Este exemplo refere-se à uma tomada de decisões com base em uma condição ser alcançada. Um exemplo factível de utilização dá-se em um ambiente em que a temperatura é controlada por uma RSSF e esta tem a missão ativar ou não o sistema de resfriamento de acordo com o perfil do usuário que encontra-se naquele instante no ambiente. Assim, temos diversas possibilidades de personalização que tornam-se mais fáceis de serem gerenciadas através da utilização de políticas.

```
Rule 1:  
    if Condition 1  
        then Action 1  
  
Rule 2:  
    if Condition 1  
        then Action 2
```

Figura 2.2: Exemplo de política simples

A utilização de políticas para gerenciamento de RSSF tem demonstrado ser uma solução viável devido à possibilidade de flexibilização no gerenciamento dos recursos limitados

dos nós sensores (QWASMI; LISCANO, 2015). Políticas têm sido utilizadas em diversas atividades de gerenciamento de RSSF, dentre elas, roteamento, alocação de nós sensores, melhoramento da qualidade de monitoramento e diminuição de gastos energéticos, dentre outros.

2.3 Redes Definidas por Software

SDN é um paradigma emergente no gerenciamento de redes em que há uma separação entre o plano de controle e o plano de dados. A Figura 2.3 ilustra a separação dos planos do paradigma SDN. A camada de aplicação é composta por todas as requisições que desejam utilizar a infraestrutura de rede, esta estabelece comunicação com o plano de controle através da utilização de algumas *application program interfaces* (API) denominadas *northbounds*. O plano de controle (ou camada de controle) consiste no gerenciamento de todas as funcionalidades da rede. Para isso é utilizado um módulo de software chamado controlador, o qual possui uma visão global do estado da rede e tem a capacidade de tomar decisões a respeito do funcionamento da mesma. Já no plano de dados (ou camada da infraestrutura), todos os demais dispositivos de rede tornam-se dispositivos de encaminhamento de pacotes, passando assim a serem gerenciados pelo controlador (JAMMAL et al., 2014). Em redes SDN, os *switches* estão localizados no plano de encaminhamento e, possuem uma estrutura denominada de tabela de fluxo que armazena informações de como o *switch* deve tratar cada pacote que chega. Para permitir a comunicação entre o plano de dados e o plano de controle, são utilizadas as chamadas API *southbound*, sendo que a mais utilizada é o OpenFlow (LARA; KOLASANI; RAMAMURTHY, 2014).

Quando uma rede SDN começa a operar, todos os seus dispositivos estão com as suas tabelas de fluxo vazias. Quando o equipamento que serve de porta de entrada na rede (ex: *switch*) recebe um pacote, a princípio ele não sabe que decisão tomar, uma vez que sua tabela de fluxo está vazia. O *switch* então repassa o pacote para o controlador utilizando o protocolo OpenFlow. O controlador por sua vez, baseado nos campos do cabeçalho do pacote e de regras já definidas, decide o que fazer com aquele pacote. Alguns exemplos de decisões são: descartar o pacote; encaminhar o pacote para uma interface de saída específica; realizar *broadcast* do pacote; dentre outras. Uma vez tomada a decisão, o controlador, também

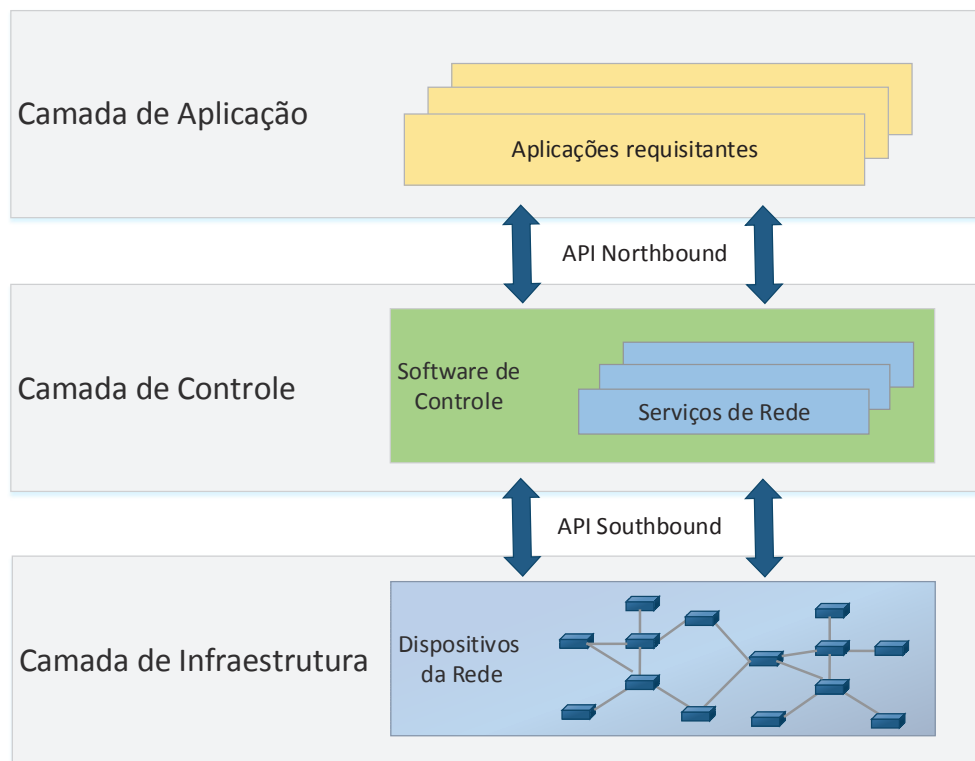


Figura 2.3: Representação dos planos de gerência no paradigma SDN.

através do OpenFlow, requisita a criação de uma entrada na tabela de fluxo do *switch* e devolve o pacote. O *switch* então realiza a operação que está designada na tabela de fluxo. A partir deste momento, sempre que um pacote com as mesmas características chega na rede, o *switch* não precisa mais consultar o controlador. Ele apenas realiza uma busca em sua tabela de fluxo para encontrar a entrada apropriada e consequentemente a ação a ser realizada.

De acordo com (JAMMAL et al., 2014) utilização do paradigma SDN traz vários benefícios quando comparado aos métodos tradicionais de gerenciamento, tais como:

- **Melhor gerenciamento:** A capacidade de gerenciamento aumenta consideravelmente, já que caso uma nova política seja estabelecida na rede, apenas se faz necessária a programação desta política no controlador. É possível, ao utilizar o paradigma SDN, fazer com que um *switch* opere como um *firewall*, ao descartar pacotes com determinadas características. Outra possibilidade é a diferenciação do tráfego através de políticas de QoS, como por exemplo, a criação de fluxos temporários e com melhor qualidade (maior largura de banda, menor atraso) para atender pacotes que possuem maior prioridade;

- **Flexibilidade para realizar alterações:** Novas funcionalidades podem ser facilmente implementadas na rede sem que haja necessidade de substituição de componentes físicos. Para isso são necessárias apenas alterações no controlador;
- **Maior compatibilidade entre equipamentos:** Outra grande vantagem de uma rede que utiliza SDN está na compatibilidade de equipamentos, pois, devido não haver utilização de softwares proprietários, torna-se mais fácil a instalação de diferentes tipos de equipamentos;
- **Diminuição da quantidade de equipamentos a serem gerenciados:** O paradigma SDN apresenta benefícios de uma abordagem centralizada para configuração de rede. Ao contrário de tecnologias de gerenciamento convencionais, o SDN propõe a utilização de um plano de controle, assim, todo o controle dos equipamentos encontra-se em uma mesma camada, isso possibilita que a realização de modificações ou adição de novas funcionalidades seja feita apenas reconfigurando o controlador, não necessitando modificar diversos equipamentos que, muitas vezes, encontram-se em locais distintos.

Inicialmente, o paradigma SDN foi projetado para o gerenciamento de redes cabeadas, porém existem na literatura diversos trabalhos que utilizam este paradigma no gerenciamento de diversos tipos de redes, inclusive RSSF (HAN; REN, 2014). A ideia de utilização de SDN em um ambiente de RSSF é propiciar uma maior eficácia no gerenciamento dos recursos que, conforme descrito anteriormente são bastante limitados (GANTE; ASLAN; MATRAWY, 2014). Contudo, a maioria destes trabalhos apresentam apenas ideias teóricas devido às limitações intrínsecas das RSSF. Por exemplo, um dos aspectos presentes no plano de dados do paradigma SDN é a tabela de fluxo. Em RSSF esta tabela torna-se difícil de ser armazenada, já que os nós sensores possuem capacidade de armazenamento limitada. Outro problema reside na utilização do protocolo OpenFlow como meio de comunicação entre o controlador (plano de controle) e os equipamentos (plano de dados) (LARA; KOLASANI; RAMAMURTHY, 2014). Em redes cabeadas e sem fio, já existem vários dispositivos que são compatíveis com este protocolo de comunicação, o que não acontece com RSSF (LUO; TAN; QUEK, 2012).

2.4 IoT e Cidades Inteligentes

Internet das Coisas (Internet of Things - IoT) é um paradigma moderno que tem o princípio básico centrado na presença de uma variedade de "coisas" ou objetos - como *Radio-Frequency IDentification* (RFID), eletrodomésticos, câmeras de vigilância, nós sensores de monitoramento, atuadores, telefones celulares, etc. - que são capazes de interagir e cooperar uns com os outros para alcançar objetivos comuns (ATZORI; IERA; MORABITO, 2010). Assim, o paradigma IoT tem por objetivo possibilitar a comunicação entre todos os objetos de um ambiente de modo a melhorar a qualidade de vida do usuário. Além disso, ao prover a interação com uma enorme variedade de dispositivos, o paradigma IoT possibilita o desenvolvimento de uma vasta gama de serviços para a população, tais como automação residencial, automação industrial, melhoramento das condições de trânsito, serviços médicos, gestão de energia inteligente, dentre outros (BELLAVISTA et al., 2013).

Dentro destes cenários, a aplicação do paradigma IoT em contextos urbanos é de total interesse do setor público, uma vez que possibilita aos governos adotarem soluções envolvendo as Tecnologias da Informação e Comunicação (TIC) na gestão dos assuntos públicos, fomentando o conceito de cidades inteligentes (*Smart Cities*) (SCHAFFERS et al., 2011). Embora não haja ainda uma definição formal de "*Smart Cities*", seu objetivo final é fazer uma melhor utilização dos recursos públicos, aumentando a qualidade dos serviços oferecidos aos cidadãos, enquanto reduz os custos operacionais. Este objetivo pode ser alcançado através da implantação do paradigma IoT, isto é, através da implantação de uma infraestrutura de comunicação que forneça acesso unificado, simples e econômico para uma infinidade de serviços públicos, desencadeando assim, melhoramento da qualidade de vida, além de aumentar a transparência para os cidadãos.

A utilização de IoT no contexto urbano, de fato, pode trazer uma série de benefícios na gestão e otimização dos serviços públicos tradicionais, como os transportes e estacionamento, iluminação, vigilância e manutenção de áreas públicas, preservação do patrimônio cultural, coleta de lixo, melhoramento dos serviços hospitalares, dentre outros (DOHLER et al., 2011), assim, a aplicação do paradigma IoT em cidades inteligentes é particularmente atraente para as administrações locais e regionais que podem se tornar os primeiros a adotar tais tecnologias, agindo assim como catalisadores para a adoção do paradigma IoT em uma

escala mais ampla.

2.5 Considerações

Este capítulo forneceu elementos conceituais fundamentais para o entendimento do presente documento de pesquisa. O capítulo seguinte, por sua vez, aponta os trabalhos mais populares, divulgados na literatura, e relacionados aos objetivos de pesquisa, no intuito de fornecer ao leitor, uma perspectiva sobre a importância do domínio sob investigação.

Capítulo 3

Estado da Arte

Este capítulo apresenta a revisão da literatura realizada no contexto desta dissertação. A Subseção 3.1 apresenta uma discussão sobre os trabalhos existentes na literatura e que propõem seleção de nós sensores, enquanto que, a Subseção 3.2, discorre sobre trabalhos relacionados à utilização de políticas para a flexibilização da gerência de RSSF.

3.1 Alocação de sensores em RSSF

O crescente interesse nos paradigmas de Cidades Inteligentes e IoT contribuiu para uma mudança no desenvolvimento e implantação de RSSF. Com todos os tipos de heterogeneidades (demanda, aplicação, protocolo, modelo de dados e hardware) que cercam estes dois paradigmas, viu-se a necessidade de adoção de RSSFH, uma vez que é inviável a utilização de uma RSSF para cada aplicação em um ambiente inteligente. Neste contexto, vários trabalhos surgiram focando neste tipo de rede de sensores. Um exemplo disso é o trabalho de (LI et al., 2014), que propõe o algoritmo denominado *reSource AlloCation in Heterogeneous Sensor Networks* (SACHSEN). Este algoritmo tem por finalidade alocar nós sensores em RSSFH utilizando como principais critérios sua energia remanescente e consumo periódico. Além disso, SACHSEN possui outros mecanismos para escolha dos nós sensores, como por exemplo a qualidade do serviço (*Quality of Service* - QOS). De acordo com os autores, a proposta se mostrou eficaz por possibilitar o aumento do tempo de vida da RSSFH.

Em (HAGHIGHI, 2014), o autor gerencia RSSFH de forma descentralizada e sob forma de licitação, onde as requisições das aplicações são equivalentes a produtos e os nós sensores

são interpretados como compradores que analisam as ofertas com o intuito de obter produtos. O preço destes produtos são calculados utilizando alguns critérios, como o consumo energético, memória e processamento necessários para executar a tarefa. Com isso, os nós a serem alocados devem possuir os requisitos necessários mas também devem ser capazes de executar tal tarefa causando o menor impacto possível aos recursos da RSSF. Ao utilizar esta abordagem, os autores demonstraram que o tempo para o processamento das requisições diminuiriam, além de conseguirem diminuir o consumo energético e, consequentemente, prolongar o tempo de vida da rede.

Alguns autores utilizam técnicas de *Clustering* para melhorar o gerenciamento de recursos em RSSF através da escolha de nós sensores específicos. *Clusters* são sub-conjuntos de nós sensores criados na RSSF com intuito de diminuir a quantidade de tráfego na rede e, consequentemente, diminuir o consumo energético ocasionado pela transmissão de pacotes. Nestes *clusters*, são escolhidos nós sensores denominados *Cluster Heads* - (CH), os quais são responsáveis por recolher todos os dados dos nós sensores do seu *cluster* e encaminhá-los à aplicação. Neste contexto, o trabalho apresentado por (JAVAID et al., 2013) propõe um protocolo de roteamento utilizando técnicas de *Clustering* para RSSFH denominado *Enhanced Developed Distributed Energy Efficient Clustering* (EDDEEC). EDDEEC escolhe dinamicamente os nós sensores que serão CH utilizando informações tais como, a energia inicial e residual dos nós e a energia média da rede. Testes realizados mostraram que o protocolo proposto ocasionou aumento do tempo de vida da RSSF.

Semelhantemente a (JAVAID et al., 2013), o trabalho de (MISHRA et al., 2015) utiliza a formação de *Clusters* com intuito de diminuir a quantidade de tráfego da RSSF, no entanto neste trabalho é proposto um algoritmo denominado *Game Theory Based Energy Efficient Cluster-Head Selection Approach* (GECSA) que realiza a seleção de CHs com base em um algoritmo denominado *Perfect Nash Equilibrium* (SPNE). Em GECSA, cada nó sensor de um *cluster* é considerado como um jogador e de cada *cluster* um CH é selecionado com base na decisão SPNE. De maneira resumida, SPNE é composto pelo conjunto $T = P, S, R, A$, onde P é o conjunto de n jogadores, S são as estratégias de encaminhamento correspondentes a cada jogador, R representa o conjunto da energia residual dos N jogadores e A é o conjunto de atribuições dos jogadores. Para a escolha há uma comparação entre as energias residuais de cada jogador e suas atribuições naquele dado momento, se este estiver ocioso e possuir

maior energia então este será escolhido como CH por um dado instante de tempo até que haja uma nova escolha. Caso hajam nós sensores que possuam mesmas quantidades de energia o critério para escolha dentre eles será a quantidade de designações anteriores, permanecendo como CH o que foi designado menos vezes.

Na literatura há inúmeros trabalhos que propõem abordagens para alocação de nós sensores, como por exemplo (KAPOOR; NANDY; MAJUMDAR, 2013) que propõe a criação de alguns algoritmos para escolha dinâmica dos nós sensores de RSSFH com base nas especificações das requisições e nos recursos dos nós sensores. Estes algoritmos objetivam utilizar os requisitos dos nós sensores (CPU, memória, consumo periódico, etc) para equilibrar a energia média dos nós sensores da RSSFH e, conseqüentemente, estender o tempo de funcionamento. Por exemplo, o algoritmo denominado *Maximum Energy First* (MEF) seleciona os nós sensores disponíveis de serem alocados com base na sua energia residual. A partir do conjunto de nós disponíveis, os nós sensores que têm a maior energia disponível são alocados para as requisições das aplicações. Outro algoritmo proposto neste trabalho é denominado *Dynamic CPU Load Balanced Allocation* (DCLBA). De acordo com a literatura, em RSSF a CPU dos nós sensores é um dos componentes de maior consumo energético. Assim, DCLBA tem por objetivo alocar nós sensores que estejam com menor percentual de utilização da CPU. Com isso, o autor consegue equilibrar o poder de processamento disponível através do equilíbrio das alocações. Isso possibilita diminuindo a utilização dos componentes de Hardware e, conseqüentemente, possibilita a diminuição do consumo de energia devido aos nós não serem sobrecarregados de requisições.

Além das técnicas utilizadas para alocação de nós sensores, há trabalhos que sugerem o gerenciamento de recursos alternando entre os modos de operação dos nós sensores ociosos. Neste contexto, o trabalho de (TAO; ZHANG, 2009) propõe algoritmos para descoberta e alternância do modo de operação de nós sensores ociosos sem impedir o correto funcionamento da RSSF. O método de funcionamento desta abordagem consiste em identificar os nós sensores ociosos e então os alterná-los para o modo *sleep* por um intervalo de tempo pré-definido. Quando este tempo é excedido então o sensor volta ao seu método de funcionamento convencional e o algoritmo realiza uma nova análise podendo ou não alterar o modo de operação, dependendo se haverão ou não requisições para o referido nó sensor.

Para que uma RSSF funcione corretamente faz-se necessária a cobertura completa do

ambiente, ou seja, é necessário que todos os nós sensores possuam comunicação com ao menos um outro nó (ROWAIHY et al., 2007). Na literatura são propostos diversos trabalhos em que os nós sensores são alocados de forma que, além de serem capazes de atender às aplicações, possam manter esta comunicação. Dentre eles podemos citar (PERILLO; HEINZELMAN, 2003) que propõe um algoritmo que organiza os nós sensores em conjuntos, de modo que alguns nós sensores possam ser desligados sem que haja perda total da área de cobertura. Para este artifício foi desenvolvido um método para seleção de nós sensores que os divide em conjuntos que tem o objetivo de maximizar o tempo de vida da rede mantendo um nível mínimo de qualidade dos serviços ofertados (tais como largura de banda disponível e cobertura).

Em seu artigo, (CARDEI; DU, 2005), aloca os nós sob forma de conjuntos disjuntos, conforme sua localização, de tal forma que qualquer um dos conjuntos é capaz de cobrir a área de interesse no referido momento, porém apenas um conjunto se mantém ativo de cada vez. Ao contrário (PERILLO; HEINZELMAN, 2003) neste trabalho os conjuntos estão escolhidos de maneira *round-robin* e o foco do problema de encontrar o número máximo de conjuntos disjuntos possível em um ambiente. À medida que o número de conjuntos de aumentar, a eficiência torna-se maior porque os nós sensores permanecem desligados por maiores intervalos de tempo.

De acordo com (ROSS, 2010) a lógica Fuzzy também pode ser utilizada para selecionar nós sensores pois esta extensão da lógica booleana possibilita cálculos de números intermediários. Em (LIANG; WANG, 2010), os autores propuseram a utilização de lógica fuzzy para selecionar nós sensores com base em três critérios: Energia residual, distância do nó até o alvo pretendido, e grau de mobilidade de cada nó. Uma vez que estes critérios são melhor compreendidos sob forma de aproximação, o problema é adequadamente resolvido através da utilização da lógica Fuzzy. Os resultados mostraram que a lógica fuzzy é capaz lidar com o problema de seleção de nós sensores, mesmo se estes fossem heterogêneos. Além disso, a utilização deste método de seleção propiciou um equilíbrio entre a energia utilizada e o desempenho de detecção.

Apesar das claras contribuições, os trabalhos mencionados anteriormente propõem abordagens em que a alocação dos nós sensores é realizada sem que haja análise prévia dos requisitos das aplicações. Além disso, a maioria dos trabalhos acima mencionados possuem

objetivo de aumentar o tempo de funcionamento da RSSF sem que haja preocupação com o desempenho das aplicações requisitantes.

3.2 Gerência de RSSF utilizando políticas

A utilização de políticas para gerenciamento de RSSF tem demonstrado ser uma abordagem bastante útil devido à possibilidade de flexibilização do comportamento da rede. Tal abordagem tem sido utilizada no trabalho de (QWASMI; LISCANO, 2015). Nele os autores propõem um *framework* baseado em políticas que são armazenadas em repositórios locais distribuídos entre os nós sensores que compõem a RSSF, assim sendo, cada nó sensor possui um repositório local que armazena uma ou mais políticas de configuração, de acordo com o *hardware* disponível. Além disso, todos os nós sensores possuem indicadores para a localização de todas as políticas na rede. Desta forma, quando um sensor necessita se auto-configurar para executar determinada funcionalidade, ele efetua uma busca ao repositório local e, caso não possua a política desejada, ele efetua a troca de políticas com os nós sensores que a possuem sob forma de *Peer-to-peer* (P2P), ou seja, todos os nós sensores possuem funcionalidades de cliente ou servidor, podendo disponibilizar ou receber políticas, conforme as necessidades da RSSF. De acordo com os autores a abordagem proposta reduz o impacto causado pela reprogramação de nós sensores pois consegue diminuir a quantidade de dados de configuração a serem trafegados na RSSF, visto que, apenas as políticas não disponíveis no nó serão transmitidas e, por se tratar de P2P muitos nós podem possuir a política desejada, então o nó solicitante pode escolher o nó mais próximo.

Os autores de (AZIM; ISLAM, 2010) propõem a utilização de políticas para gerenciamento de técnicas de *Clusters* que compõem a RSSF. Esta proposta utiliza políticas que, de acordo com o estado da RSSF, alteram o método de escolha dos *cluster heads*. Por exemplo, em uma das políticas, quando o modelo de gerenciamento detecta que o nível de energia da RSSF atingiu um determinado limiar, a política de economia de recursos é ativada, a qual determina que o intervalo de tempo entre as escolhas dos *cluster heads* seja diminuído. Esta alteração faz com que os nós sensores permaneçam menos tempo desempenhando uma funcionalidade que requer alto consumo de energia.

Em (XIAO; CHEN; XIAO, 2007) também são utilizadas técnicas de *clustering* aliadas à

políticas com objetivo de maximizar o tempo de vida dos *clusters* em uma RSSF. O modelo propõe a implementação de políticas para gerenciamento dinâmico da potência de transmissão do rádio dos nós sensores que compõem o *cluster* de acordo com os níveis de energia de cada um deles. Assim, a potência de transmissão dos nós sensores é alterada de modo a maximizar seu tempo de funcionamento, porém sem que haja perda de comunicação com os nós sensores vizinhos.

O trabalho de (ZHANG; XU, 2010) propõe uma arquitetura baseada em políticas para gerenciamento de RSSF. Este gerenciamento consiste na reconfiguração do funcionamento da RSSF com base na energia restante. Além disso, a arquitetura proposta possibilita a reconfiguração automática da RSSF caso um nó tenha seu funcionamento interrompido, possibilitando que a RSSF mantenha-se operacional até que haja esgotamento de todos os nós sensores ou haja perda da comunicação com o *Sink Node* devido a baixa disponibilidade de nós.

Conforme demonstrado em trabalhos anteriores, há diversas técnicas têm sido propostas para a otimização do consumo energético em RSSF. Além dos demais, o trabalho de (JIANG,) propõe a utilização de políticas para gerenciamento de técnicas de diminuição do consumo energético em RSSF. Esta abordagem consiste basicamente na diminuição do tempo de transmissão de informações através do canal sem fio dos nós sensores. Tais políticas tem por objetivo definir a quantidade de tempo em que o módulo de rádio permanecerá ligado, baseando-se na quantidade de energia restante no nó sensor. De acordo com o autor, esta tem se mostrado uma técnica eficiente para a conservação de energia.

A segurança dos dados trafegados é um fator primordial para o correto funcionamento de uma RSSF. Neste âmbito políticas têm sido utilizadas para criação de Sistemas de detecção de intusão (*Intrusion Detection System* - IDS). O trabalho de (MAMUN et al., 2012) propõe um IDS baseado em políticas para arquiteturas hierárquicas. Tal sistema tem por objetivo aumentar a segurança das informações trafegadas na RSSF e evitando a utilização de métodos de criptografia. Para tanto foram propostos componentes Hierárquicos que funcionam como um *proxy*, permitindo análise das informações trafegadas com intuito de detectar e mitigar a ação de intrusos. As políticas utilizadas pelo IDS possibilitam que hajam modificações no método de funcionamento da RSSF quando há detecção de intrusão.

3.3 Considerações

Durante este capítulo foram apresentados trabalhos relacionados ao domínio de pesquisa em questão, evidenciando o interesse da comunidade científica em relação a temática discutida. Na literatura existem inúmeras abordagens, de modo a contemplar diversos objetivos. No entanto demonstrou-se que os recursos de hardware disponíveis nos nós sensores e os requisitos exigidos para que uma aplicação possa funcionar corretamente, apesar de serem quesitos importantes, não foram fatores decisivos na maior parte dos trabalhos e, os que contemplam este objetivo, propõem abordagens que gerenciam somente um dos recursos por vez, (por exemplo, os algoritmos propostos no trabalho de (KAPOOR; NANDY; MAJUMDAR, 2013)). Baseado nesta premissa, o presente trabalho propõe uma abordagem para seleção dinâmica e sob-demanda de sensores, levando em consideração as características físicas dos nós sensores e as demandas das aplicações requisitantes.

Capítulo 4

Framework POSSE-HWSN -

Policy-based Sensor SElection in HWSN

Nos capítulos anteriores foram apresentados os trabalhos correlatos e alguns dos conceitos fundamentais para o entendimento desta dissertação. Este capítulo tem por objetivo apresentar o *Framework* POSSE-WSN, explicando o seu funcionamento, bem como a interação entre seus componentes.

4.1 Arquitetura de POSSE-HWSN

Esta dissertação propõe POSSE-HWSN, um *framework* baseado em políticas para seleção dinâmica de nós sensores em RSSFH. POSSE-WSN inspira-se no paradigma SDN ao utilizar um módulo controlador que possui uma visão geral da rede e toma decisões de gerenciamento. Em POSSE-HWSN, o controlador tem conhecimento dos estados de todos os nós sensores da rede, bem como de suas características físicas. De posse destas informações e dos requisitos das requisições recebidas, o controlador consegue selecionar de forma dinâmica os nós sensores mais apropriados para atender a cada requisição. Esta seleção baseia-se em políticas pré-definidas, as quais permitem a dinamicidade e flexibilidade de escolha do *framework*.

A Figura 4.1 apresenta o *framework* POSSE-HWSN. Na camada de aplicação encontram-se as aplicações que fazem requisições à rede através de POSSE-HWSN. Na camada de infraestrutura estão dispostos os nós sensores que compõem a RSFFH. Nesta ca-

mada há também um nó denominado de *Sink Node*, o qual engloba as funcionalidades do nó sorvedouro e também é o responsável por comunicar-se com a camada de controle, na qual reside o controlador.

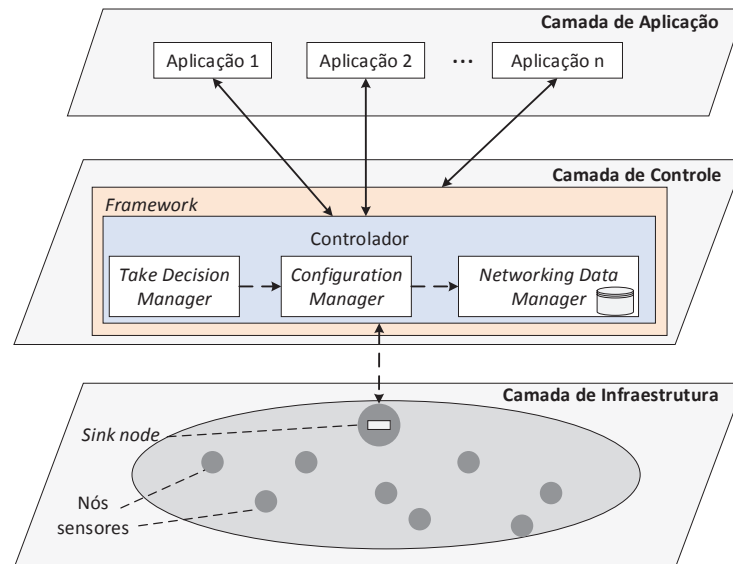


Figura 4.1: Arquitetura do *Framework* POSSE-HWSN

Em POSSE-HWSN, o controlador intercepta as requisições direcionadas à HWSN e as analisa com o objetivo de tomar uma decisão sobre como melhor atendê-las. Este atendimento refere-se a uma escolha mais inteligente de nós sensores para cada requisição recebida. Para realizar esta tarefa, o controlador conta com três componentes, como mostrado na Figura 4.2.

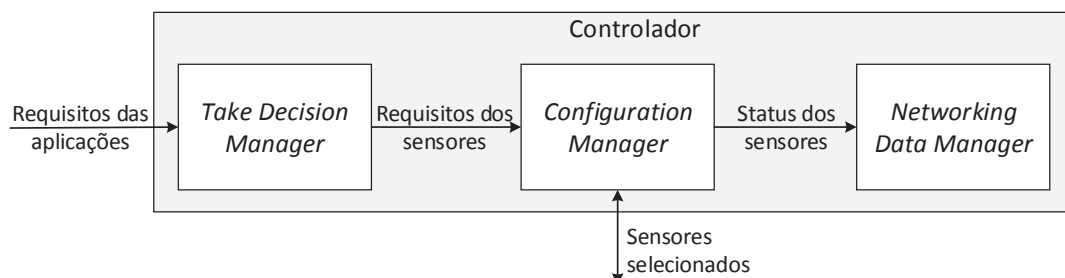


Figura 4.2: Controlador do *Framework* POSSE-WSN

Como mencionado anteriormente, o controlador possui uma visão geral dos estados de

todos os nós sensores da HWSN. Para conseguir manter esta visão, é necessário utilizar alguma forma de persistência das informações dos sensores. O *Networking Data Manager* (NDM) é o componente responsável pelo armazenamento, atualização e consulta de todas as informações relevantes de cada nó sensor da HWSN, tais como, CPU, memória, capacidade de transmissão, energia remanescente, índices de consumo energético, dentre outros. Por sua vez, o componente *Take Decision Manager* (TDM) tem como finalidade receber as requisições das aplicações e analisar seus requisitos de forma a escolher uma política de seleção de nós sensores mais adequada às demandas da aplicação. Baseando-se na política escolhida, o TDM informa ao componente *Configuration Manager* - (CM) quais as características a serem levadas em consideração ao escolher os nós sensores. O CM então realiza uma consulta ao NDM para obter as informações dos estados atuais dos nós sensores, conseguindo assim a "visão global" da rede. De posse destas informações, o CM consegue selecionar os nós sensores mais adequados para atender a determinada aplicação de acordo com as políticas implantadas no sistema. Uma vez selecionados os nós, o CM informa ao *Sink Node* quais os nós sensores a serem utilizados na aplicação atual. Além disso, o CM também possui a função de receber da camada de infraestrutura as informações de estados dos sensores e repassá-las para o NDM.

4.2 Funcionamento de POSSE-HWSN

De forma a facilitar o entendimento do *Framework* POSSE-WSN, é apresentado na Figura 4.3 um diagrama de sequência que ilustra seu funcionamento. Neste diagrama é possível observar as interações ocorridas entre as entidades do *Framework* no momento em que é realizada uma solicitação originada por uma aplicação.

Ao solicitar um serviço da RSSFH, a aplicação dispara uma requisição ao controlador através da mensagem *requestMon*. Além da identificação da aplicação, toda requisição disparada contém os parâmetros que especificam a demanda que a aplicação solicita da rede. Tais parâmetros são: tipo de fenômeno físico a ser monitorado (temperatura, pressão, umidade, movimento etc.), duração do monitoramento, intervalo de tempo em que os sensores devem monitorar e transmitir os dados coletados, o número mínimo de sensores requeridos para atender à requisição, além de informar se há alguma necessidade adicional (criptogra-

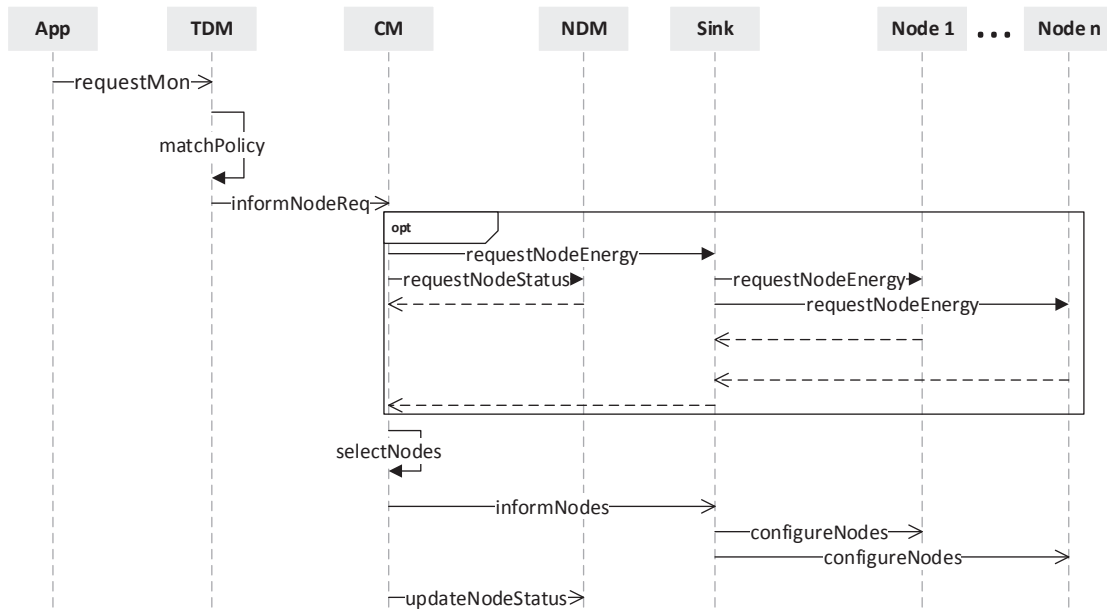


Figura 4.3: Diagrama de sequência do *Framework* POSSE-WSN

fia dos dados coletados, por exemplo). Este último parâmetro é essencial para a escolha da política de seleção de sensores, como será explicado adiante.

Ao receber uma requisição, o TDM identifica os requisitos informados pela aplicação e seleciona a política mais adequada (*matchPolicy*) de acordo com estas informações. A política selecionada define quais são os recursos necessários dos nós sensores para atender de modo mais satisfatório a requisição (esta atividade é explicada com mais detalhes na Subseção 4.2.2). Uma vez conhecidos estes recursos, o TDM os informa ao CM (*informNodeReq*). O CM por sua vez requisita, ao NDM, os estados atuais dos nós sensores (*requestNodeStatus*). O CM também requisita ao *Sink Node* os níveis de energia atuais dos nós sensores (*requestNodeEnergy*). É importante ressaltar que estas últimas mensagens (*requestNodeStatus*) só acontecem em duas condições: A requisição corrente é a primeira requisição feita à rede ou a diferença entre o tempo de chegada da requisição corrente e a requisição anterior é menor que um *tbr* (*time between requisitions*).

De posse das informações dos estados dos nós sensores, o CM seleciona os sensores mais apropriados (*selectNodes*) de acordo com os requisitos definidos pela política escolhida pelo TDM (esta atividade será descrita com mais detalhes na Subseção 4.2.3). O CM então informa ao *Sink Node* os nós sensores escolhidos (*informNodes*), que por sua vez os contacta

para repassar os parâmetros de configuração (*configureNodes*). Por fim, o CM solicita ao NDM para atualizar os estados dos nós sensores na base de dados (*updateNodeStatus*).

4.2.1 Visão global da rede

Uma das principais premissas de POSSE-HWSN é a utilização de um controlador que possui a visão geral dos estados de todos os nós sensores da RSSF. Para conseguir isto, foi definida uma estrutura para armazenamento das informações ao longo do tempo de cada nó sensor. A Figura 4.4 mostra o modelo relacional desta estrutura, contendo três entidades: (i) *Nó sensor*, que representa um nó sensor da rede; (ii) *Aplicação*, que representa a requisição de cada aplicação feita à rede; e (iii) *Alocação*, que representa a associação de um nó sensor a uma determinada aplicação. Além disso, um nó sensor pode estar alocado **N** vezes e uma aplicação pode ter **N** alocações. Em outras palavras, cada nó sensor pode ser alocado a **N** aplicações e uma aplicação pode ser atendida por **N** nós sensores. Desta forma, é possível saber quantos nós sensores estão alocados a cada aplicação. A seguir, são descritos os campos de cada entidade:

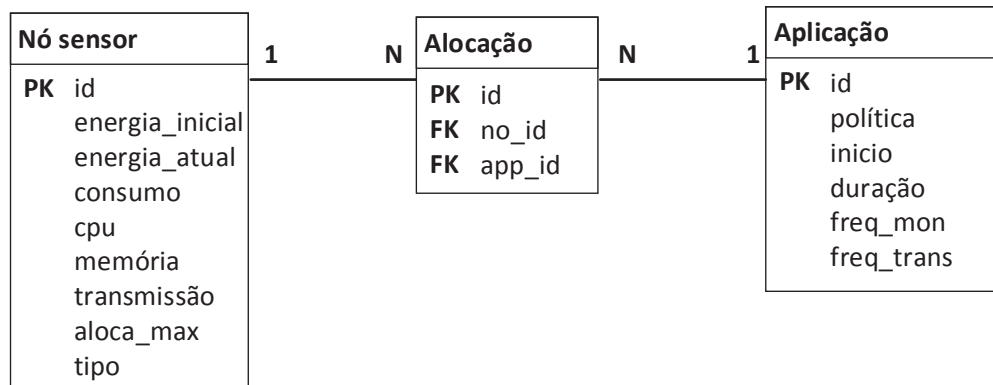


Figura 4.4: Estrutura da base de dados

Nó sensor

- *id*: Identificador do nó sensor - Chave primária (PK);
- *energia inicial*: Energia inicial do nó sensor;
- *energia atual*: Energia atual do nó sensor;
- *consumo*: Consumo energético do nó sensor;
- *cpu*: Frequência (velocidade) do processador do nó sensor;

- *memória*: Quantidade de memória do nó sensor;
- *transmissão*: Taxa de transmissão suportada pelo equipamento de transmissão do nó sensor;
- *aloca_max*: Número máximo de aplicações que um nó sensor consegue atender;
- *tipo*: Tipo de sensor (luminosidade, temperatura, presença, etc).

Aplicação

- *id*: Identificador da aplicação requisitante - Chave primária (PK);
- *política*: Política associada à aplicação requisitante;
- *inicio*: Início de atendimento da aplicação requisitante;
- *duração*: Duração de atendimento da aplicação requisitante;
- *freq_mon*: Frequência com que o sensoriamento deve ser realizado;
- *freq_trans*: Frequência de envio dos dados sensorizados para a aplicação requisitante.

Alocação

- *id*: Identificador da alocação - Chave primária (Primary Key - PK);
- *no_id*: Identificador do nó sensor associado - Chave estrangeira (Foreign Key - FK);
- *app_id*: Identificador da aplicação associada - Chave estrangeira (FK).

4.2.2 Políticas de seleção

O *framework* POSSE-HWSN utiliza políticas para auxiliar na seleção dos sensores mais apropriados. Quando o controlador recebe uma mensagem de requisição de uma aplicação, várias informações podem ser enviadas através desta mensagem. No *framework* proposto, foram utilizados sete parâmetros para definir o formato desta requisição, conforme apresentado na Tabela 4.1.

Ao receber a mensagem de requisição da aplicação, o TDM extrai o parâmetro *appDemand*, o qual é o fator decisivo para a escolha da política. Nesta dissertação, as demandas da aplicações requisitantes foram categorizadas em cinco tipos:

- **Pré-processamento**: Quando uma aplicação necessita de um pré-processamento por parte dos nós sensores dos sensorizados antes do envio para a aplicação. Um exemplo típico é quando os dados precisam ser criptografados.

Tabela 4.1: Parâmetros da requisição das aplicações

Parâmetro	Descrição
<i>id</i>	Identificação da aplicação requisitante
<i>phyPhen</i>	Fenômeno físico a ser monitorado pelos nós sensores
<i>duration</i>	Total de tempo que os nós sensores devem monitorar o fenômeno físico
<i>monFrequency</i>	Frequência com que os nós sensores devem monitorar o fenômeno físico
<i>transFrequency</i>	Frequência com que os nós sensores devem transmitir os dados monitorados para a aplicação
<i>numSensors</i>	Quantidade de nós sensores requeridos pela aplicação
<i>appDemand</i>	Tipo de demanda que a aplicação requer da HSWN

- **Tratamento de imagens:** Demanda de aplicações que requerem algum processamento em imagens capturadas. Apesar de ser também um tipo de pré-processamento, esta demanda também requer espaço de armazenamento para as imagens. Um exemplo de aplicação que possui este tipo de demanda é o de reconhecimento facial.
- **Armazenamento de informações:** Quando é necessário o armazenamento de um volume considerável de dados no nó sensor antes do envio para a aplicação requisitante. Esta demanda é típica de cenários nos quais os nós sensores devem monitorar um fenômeno físico durante um intervalo de tempo e somente após o fim deste intervalo os dados devem ser enviados. Por exemplo, o monitoramento da temperatura da água em reservatórios ao longo do dia.
- **Alta taxa de transmissão:** Quando aplicações demandam que os nós sensores colem e enviem os dados de forma ininterrupta, necessitando assim de uma alta taxa de transmissão por parte do equipamento. Por exemplo, monitoramento constante do tráfego de uma via pública movimentada.
- **Outros:** Caso a demanda da aplicação requisitante não se encaixe em qualquer das demandas anteriores.

Uma vez definida a demanda, o TDM escolhe qual política será utilizada para atender a

aplicação requisitante e consequentemente, qual recurso deve ser levado em consideração na seleção dos nós sensores. Ou seja, através das políticas, é possível realizar um mapeamento das necessidades das aplicações aos recursos dos nós sensores que melhor satisfazem estas necessidades. Este processo é ilustrado na Figura 4.5, a qual é dividida em três áreas: na área superior são mostradas as demandas que podem chegar ao TDM, na área central são apresentadas as políticas utilizadas nesta dissertação e na área inferior estão os recursos dos nós sensores. Os padrões de seta indicam o fluxo do mapeamento demanda-política-recurso.

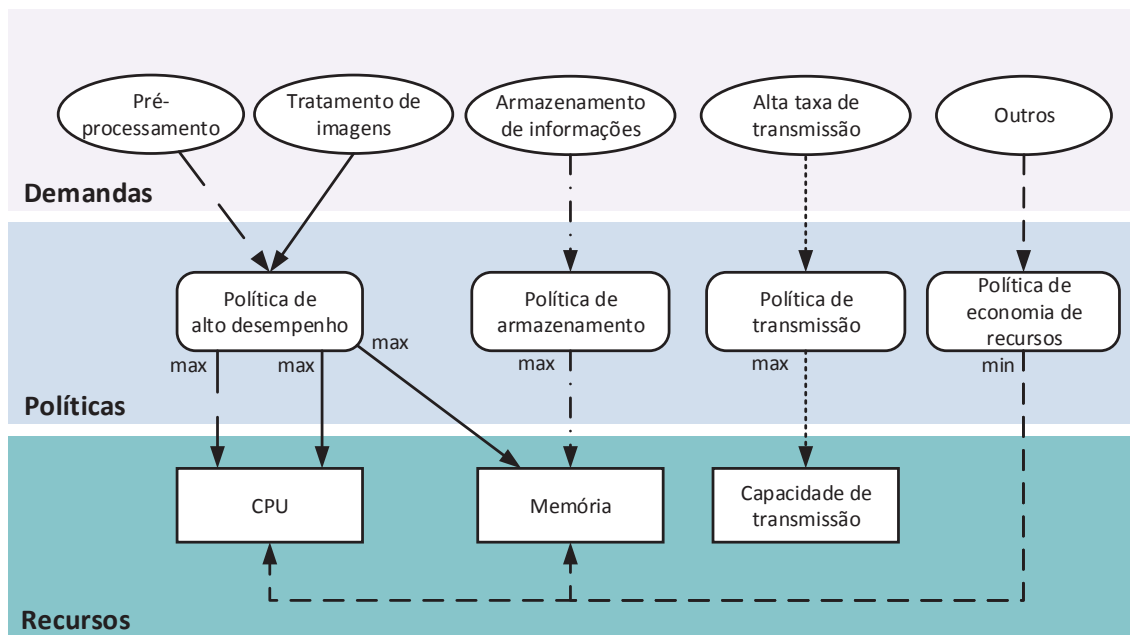


Figura 4.5: Escolha de políticas

A política de alto desempenho é escolhida se a demanda da aplicação for por pré-processamento ou de tratamento de imagens. Caso seja de pré-processamento a política mapeia o recurso na necessidade de poder de processamento (CPU) de nós sensores. Assim, ao selecionar os sensores para atender esta requisição, serão escolhidos os sensores com os maiores níveis de poder de processamento. Por outro lado, se a demanda for por tratamento de imagens, então a política de alto desempenho mapeia a demanda no recurso de CPU e quantidade de memória do nó sensor. Da mesma forma, ao selecionar os nós sensores, serão considerados aqueles com maior quantidade de memória, além do poder de processamento.

Para o caso da demanda da aplicação ser o de armazenamento de informações, então é selecionada a política de armazenamento, a qual leva em consideração a quantidade de me-

mória disponível nos nós sensores. Se a demanda da aplicação for a taxa de transmissão, então a capacidade de transmissão será o fator determinante na escolha dos nós sensores. Finalmente, caso a demanda não seja qualquer das mencionadas anteriormente, então é escolhida a política de economia de recursos. O objetivo desta política é economizar os recursos da rede, levando em consideração os nós sensores com os menores níveis de CPU e memória.

O Algoritmo 1 mostra o pseudocódigo de como é realizada a tomada de decisão por cada política e de qual recurso de hardware deverá ser levado em consideração de acordo com a demanda de cada aplicação. Como já mencionado, o objetivo desta dissertação não é o desenvolvimento de políticas para seleção de nós sensores, mas sim a utilização destas. Portanto, nesta dissertação foram desenvolvidas políticas que possuem regras simples apenas para a validação da proposta. Finalmente, uma vez definido(s) o(s) recurso(s) dos nós sensores a ser(em) levado(s) em consideração, o TDM repassa esta informação ao CM, que por sua vez inicia o processo de seleção dos nós sensores.

4.2.3 Seleção de nós sensores

Após a política escolhida definir qual(is) requisito(s) a ser(em) levado(s) em consideração, o CM inicia a etapa de seleção dos nós sensores que irão atender à aplicação requisitante. Esta etapa inicia-se a partir da obtenção dos estados dos nós sensores por parte do CM. Ao solicitar estas informações ao NDM, é retornada uma lista contendo todos os nós sensores disponíveis ordenados pelo recurso desejado (CPU, memória, etc). Entende-se por nó sensor disponível aquele que ainda não alcançou o seu número máximo de aplicações atendidas. Além disso, para simplificação da proposta, não é considerado o tipo de fenômeno físico a ser monitorado pelo nó sensor. Em outras palavras, considera-se que um nó sensor consegue monitorar qualquer fenômeno físico (ex: luminosidade, temperatura, movimento, etc). Extensões desta dissertação tratarão desta questão.

De posse da lista dos nós sensore disponíveis, o CM verifica o número de nós sensores requeridos pela aplicação para criar uma segunda lista com os nós sensores selecionados. O objetivo deste processo é selecionar primeiramente os nós sensores que possuem os recursos desejáveis e que possuam energia suficiente, porém que não estejam atendendo a outras solicitações. Caso não haja nós sensores suficientes, então são selecionados os nós sensores com as mesmas características, porém que estejam alocados a outras requisições. Um nó

Algorithm 1: ESCOLHA DOS RECURSOS DOS NÓS SENSORES

```

Input: Demanda
Output: NSList
// NSList: Lista de nós sensores possíveis de serem
//           selecionados
// SQLResult: Resultado da consulta ao banco de dados
1 switch Política do
2   case Alto Desempenho
3     if Demanda = Pré-processamento then
4       NSList = SQLResult // Em ordem decrescente de CPU
5     if Demanda = Tratamento de Imagens then
6       NSList = SQLResults // Em ordem decrescente de CPU e
        Memória
7   case Armazenamento
8     if Demanda = Armazenamento de Informações then
9       NSList = SQLResult // Em ordem decrescente de Memória
10  case Transmissão
11    if Demanda = Alta taxa de transmissão then
12      NSList = SQLResult // Em ordem decrescente de
        capacidade de transmissão
13  case Economia de Recursos
14    if Demanda = Outros then
15      NSList = SQLResult // Em ordem crescente de CPU e
        Memória
16  return NSList

```

sensor somente estará impossibilitado para novas alocações caso não possua energia ou se houver atingido um número máximo de alocações.

O fluxograma da Figura 4.6 ilustra a etapa de seleção. Para uma melhor compreensão do diagrama, assume-se as seguintes variáveis:

LD: Lista que contém os nós sensores que estão disponíveis;

LS: Lista que contém os nós sensores que foram selecionados para atender a aplicação;

QNR: Quantidade de nós sensores requeridos pela aplicação;

QNS: Quantidade de nós sensores selecionados para atender a aplicação;

NC: Nó sensor corrente em *LD* sendo analisado;

NS: Nó sensor seguinte em *LD* sendo analisado;

Prox(Nó): Indica o próximo nó sensor depois de *Nó* em *LD*, para qualquer nó sensor;

$R(Nó)$: Recurso do nó sensor $Nó$, para qualquer nó.

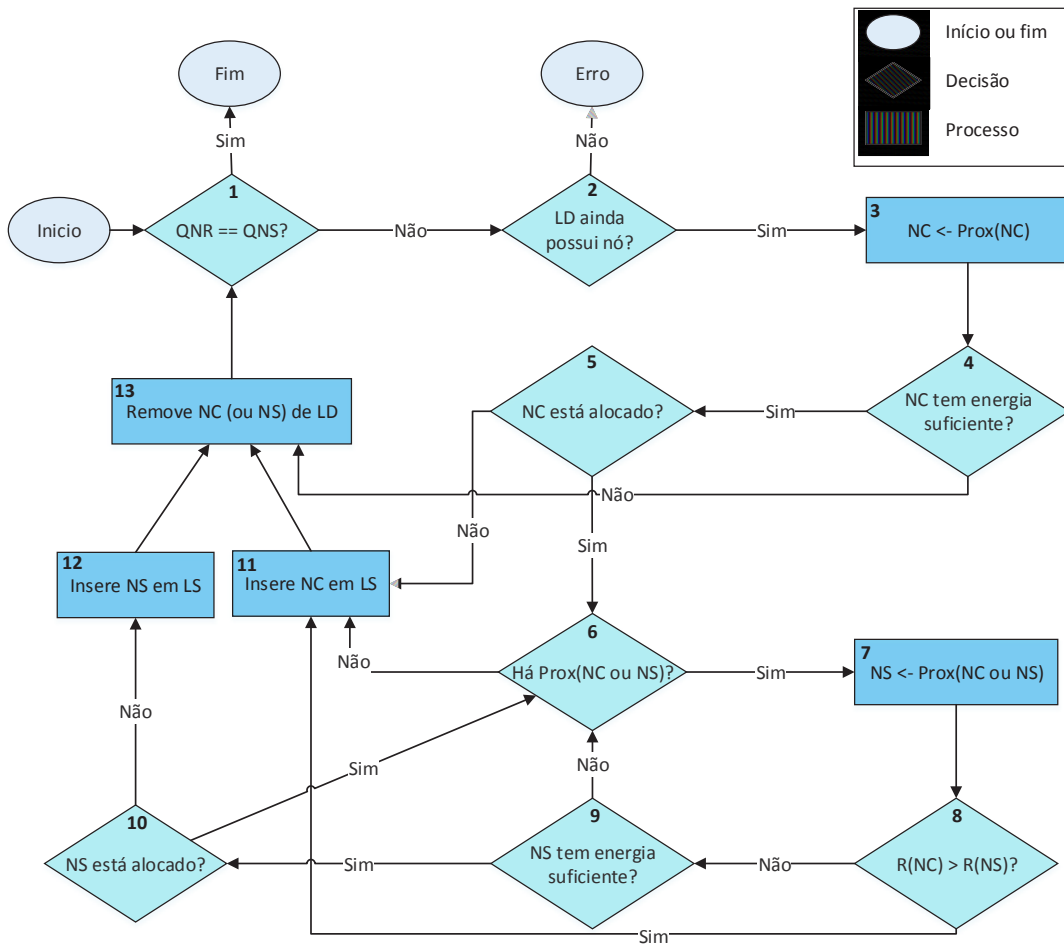


Figura 4.6: Fluxograma da seleção dos nós sensores

No início do etapa de seleção, o conteúdo inicial de algumas das variáveis são: LD contém todos os nós sensores disponíveis, LS está vazia, o valor de QNR é especificado pela aplicação e o valor de QNS é zero. Ao iniciar a etapa de seleção, o CM verifica se a quantidade de nós sensores requeridos é igual à de nós sensores selecionados (1). Em caso positivo, o processo de seleção termina com sucesso, com LS contendo os nós selecionados. Senão, verifica-se se LD ainda possui algum nó a ser analisado (2). Caso não possua, o processo termina com erro, indicando que não há nós suficientes para atender à requisição. Se existir, então as informações do próximo nó são recuperadas (3). Ao executar este processo a primeira vez, o NC é o primeiro nó sensor em LD .

De posse das informações do nó corrente, é verificado se o mesmo possui energia suficiente para atender à requisição (4). Se não possuir, o nó é removido da lista de nós disponíveis

(13). Se possuir energia, é verificado se o nó já está atendendo alguma requisição (5). Caso não esteja, o nó é inserido na lista de nós selecionados para atender à requisição (11). Senão, verifica-se se ainda há nó disponível na LD (6). O objetivo deste teste é que caso o nó corrente já esteja sendo utilizado, verifica-se o próximo nó sensor na lista. Se não há nó disponível, então o nó corrente é adicionado à LS (11). Caso ainda haja nó disponível, sua informação é recuperada (7). Após este processo, verifica-se se o recurso do nó corrente é maior que o recurso do nó seguinte (8). Em caso afirmativo, então o nó corrente é adicionado em LS, mesmo que ele já esteja atendendo à requisições. Senão, é verificado se o nó seguinte possui energia. Caso não possua, verifica-se se há próximo nó sensor em LD, voltando ao passo 6. Se o nó seguinte possuir energia, verifica-se então se ele também já está alocado (10). Caso esteja, volta-se novamente ao passo 6. Se o nós seguinte não estiver alocado ainda, o mesmo é adicionado em LS (13).

Ao final do processo descrito acima, o controlador enviará ao *Sink Node* uma lista contendo todos os nós sensores selecionados. É importante ressaltar que este processo de seleção não busca escolher os nós ótimos para atender às requisições, mas sim escolher nós mais adequados dentre aqueles que estão disponíveis.

4.3 Considerações

Através das seções anteriores foi explanado como é o funcionamento de POSSE-HWSN ao receber uma requisição de uma aplicação até o momento em que os nós sensores são escolhidos. O objetivo de POSSE-HWSN é selecionar os nós sensores mais adequado para atender às aplicações requisitantes de acordo com as suas demandas. Esta seleção é guiada através de políticas, o que confere a dinamicidade e flexibilidade do *framework*. No próximo capítulo será apresentada a metodologia utilizada para a implementação do *framework*, além dos cenários de testes e a análise dos resultados obtidos.

Capítulo 5

Protótipo e Resultados Experimentais

Neste capítulo, são apresentados detalhes da implementação do protótipo para validar o *framework* POSSE-HWSN. Este protótipo implementa as funções de POSSE-HWSN descritas no capítulo anterior. Além disso, é demonstrado o ambiente de simulação bem como os cenários de testes executados neste ambiente. Finalmente, os resultados destes testes são apresentados e analisados.

5.1 Implementação do protótipo

Para validar o *framework* POSSE-HWSN, foi implementado um protótipo utilizando o simulador Castalia versão 3.2 (BOULIS, 2011), o qual tem como plataforma base o Omnet++ (VARGA; OPENSIM, 2011). O Castalia é um simulador de RSSF e de Redes de Área Corporal *Body Area Networks* (BAN) muito utilizado para testes de algoritmos e protocolos. A escolha do Castalia baseou-se principalmente no fato deste possuir código-fonte aberto, possibilitando assim a implementação de eventuais funcionalidades. Ademais, é uma ferramenta bastante utilizada pela comunidade acadêmica, além de possuir documentação e materiais instrucionais em quantidade satisfatória que servem de base para o seu aprendizado (RASTEGARNIA; SOLOUK, 2011).

Para simular os nós sensores, o Castalia utiliza módulos comunicantes entre si, conforme ilustrado na Figura 5.1. Existem basicamente três módulos: o primeiro módulo é denominado Módulo físico, este tem por funcionalidade representar o ambiente físico que os nós sensores devem monitorar; o segundo módulo (Módulo de Hardware) representa o hardware

dos nós sensores e; o terceiro módulo (Módulo Wireless) representa o canal sem fio, pelo qual os nós sensores conseguem comunicar-se entre si. As setas que interligam as entidades significam a troca de mensagens entre elas. Com isso, por exemplo, quando um nó deseja enviar uma informação ele utiliza o módulo de comunicação sem fio que então decide quais nós sensores devem receber os pacotes.

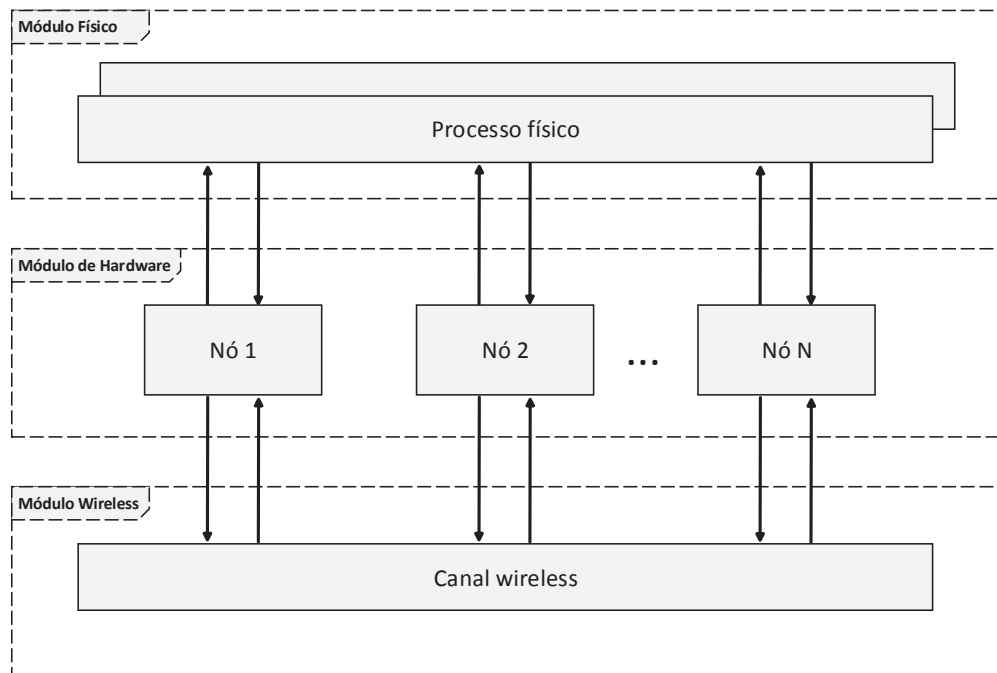


Figura 5.1: Módulos do Castalia. Adaptado de (BOULIS, 2011)

A Figura 5.2 apresenta o modelo lógico utilizado no Castalia para implementar os componentes dos nós sensores. As setas sólidas representam troca de mensagens entre as entidades, enquanto que as setas tracejadas significam simples chamadas de métodos entre módulos. Por exemplo, alguns módulos quando utilizados, chamam uma função do gerente de recursos para sinalizar que a energia foi consumida, ou que outros recursos de *hardware* estão sendo utilizados para determinada funcionalidade. O módulo de Aplicações é frequentemente utilizado para o gerenciamento de *hardware* ou criação de novas funcionalidades a serem simuladas na RSSF. Os módulos de *Medium Access Control* (MAC), Roteamento e Rádio têm por objetivo gerenciar o fluxo de tráfego e a comunicação entre os nós sensores da RSSF.

Além disso, o Castalia utiliza ainda dois arquivos de inicialização da RSSF, que são: (i) **.ned**, em que são definidos os parâmetros gerais de funcionamento da rede, como por exem-

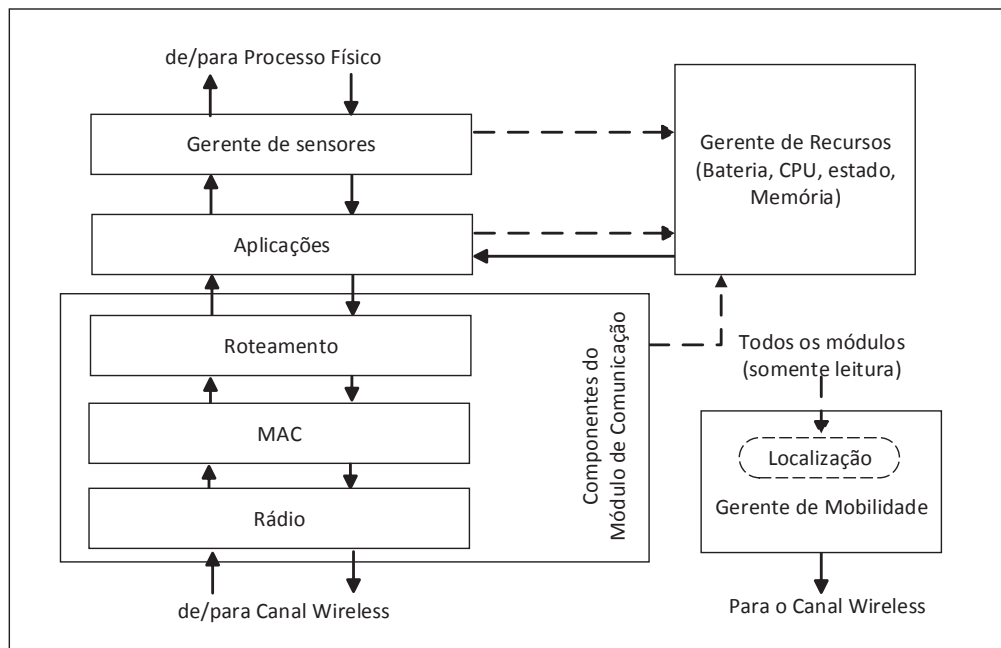


Figura 5.2: Modelo lógico dos componentes de nós sensores (BOULIS, 2011)

plo identificadores, prioridades para encaminhamento de tráfego, tamanhos de pacotes que serão permitidos ou descartados, dentre outros e; (ii) **.ini** no qual são definidos os parâmetros necessários para funcionamento do cenário de simulação, como por exemplo o recurso de *hardware* em cada sensor (CPU, memória, capacidade de transmissão e potência do rádio, etc.), o tipo de roteamento utilizado, topologia da rede, dentre outros.

POSSE-HWSN foi implementado em C++ como um módulo externo ao Castalia. Apesar de ser uma entidade externa, os componentes de POSSE-HWSN realizam chamadas aos métodos do módulo de Aplicações do Castalia para que seja possível, por exemplo, informar ao *Sink Node* sobre quais os nós sensores escolhidos. Para possibilitar essa comunicação, POSSE-HWSN cria uma instância do módulo de Aplicações, permitindo assim realizar chamadas a métodos do Castalia e, conseqüentemente, enviar e receber informações da RSSFH simulada. Para persistir as informações dos nós sensores e das requisições, foi utilizado o banco de dados MySQL.

5.2 Ambiente de simulação

Para a validação de POSSE-HWSN foram realizados testes utilizando um ambiente simulado que possui dimensões 150m x 150m. O estudo sobre a quantidade de nós sensores e seus melhores posicionamentos de forma a melhorar a área de cobertura é um tema de pesquisa por si só (SANGWAN; SINGH, 2015), (AMMARI, 2010), fugindo ao escopo desta dissertação. Assim sendo, o trabalho de (ZHI-YAN; JIAN-ZHEN, 2012) serviu como base para definir a quantidade mínima de nós sensores e o seus posicionamentos para propiciar a cobertura ideal do ambiente. O cálculo é realizado através da fórmula $N = (L * P) / (2r^2)$, no qual N representa a quantidade de nós sensores necessária, L e P representam a largura e a profundidade do ambiente, respectivamente, e r indica o raio de alcance dos nós sensores. Considerando um raio máximo de alcance para cada nó sensor de 12m, foi obtida a quantidade de aproximadamente 78 nós sensores, arredondados para 81 (80 nós sensores e 1 *Sink Node*), que foram uniformemente dispostos no ambiente de simulação, sendo o *Sink Node* posicionado no centro do ambiente para diminuir a quantidade de dados trafegados na RSSFH. Para as funções de roteamento é utilizado o algoritmo padrão do Castalia, o qual se baseia na estratégia de *Multipath routing* ((RADI et al., 2012)).

Todos os nós sensores que compõem a RSSFH possuem parâmetros de CPU, memória, capacidade de transmissão e energia. Cada um destes parâmetros foram divididos em dois níveis de disponibilidade (um mais alto e outro mais baixo). Devido à quantidade de parâmetros e suas variações, obtemos 2^4 possibilidades de diferentes configurações para os nós sensores. Assim, para efeitos de testes, foram criados 16 grupos cada um contendo 5 nós sensores em que cada grupo possui uma das possíveis combinações de parâmetros. A Tabela 5.1 sumariza os parâmetros destes 16 grupos. Além disso, todos os nós sensores foram configurados para atender até 4 requisições simultâneas. Ou seja, um mesmo nó sensor pode estar alocado para até 4 aplicações diferentes

Neste ambiente simulado, aplicações realizam requisições enviando mensagens contendo os parâmetros listados na Tabela 4.1. Todos os parâmetros descritos anteriormente foram implementados, porém apenas os parâmetros *numSensors* e *AppDemand* são essenciais para a tomada de decisão sobre quais nós sensores escolher. Os demais parâmetros serão tratados em extensões deste estudo.

Tabela 5.1: Recursos de *Hardware* dos nós sensores utilizados na simulação

Grupo	Energia (j)	CPU (Mhz)	Memória (Kb)	Transmissão (Kbps)
G1	9360	200	128	256
G2	9360	200	128	512
G3	9360	400	128	256
G4	9360	400	128	512
G5	9360	200	256	256
G6	9360	200	256	512
G7	9360	400	256	256
G8	9360	400	256	512
G9	18720	200	128	256
G10	18720	200	128	512
G11	18720	400	128	256
G12	18720	400	128	512
G13	18720	200	256	256
G14	18720	200	256	512
G15	18720	400	256	256
G16	18720	400	256	512

Conforme mencionado no Capítulo 3, a grande maioria das abordagens encontradas na literatura tem por objetivo selecionar nós sensores com o intuito de prolongar o tempo de vida da RSSF. Nestas abordagens, geralmente são selecionados os nós sensores com maior energia e/ou com menor consumo, como por exemplo os trabalhos de (LI et al., 2014), (LIANG; WANG, 2010) e (AZIM; ISLAM, 2010). Desta forma, foi implementado o mecanismo de seleção de nós sensores baseado em (LI et al., 2014) chamado de SACHSEN, o qual escolhe os nós sensores com a maior energia. Esta implementação é utilizada para servir de base de comparação entre POSSE-HWSN e as propostas da literatura. Como nesta abordagem, para um nó sensor atender a uma requisição, ele precisa apenas não ter chegado ao seu limite de requisições atendidas, é feito uma consulta no banco de dados que retorna de forma direta a quantidade exata de nós sensores ordenados pela energia e que ainda possam atender requisições.

5.3 Cenários de testes e resultados

Esta seção apresenta os cenários de testes utilizados para validar o *framework* POSSE-HWSN e as análises dos resultados alcançados. Estes testes foram realizados no ambiente

de simulação descrito anteriormente e têm por objetivo verificar o *overhead* imposto por POSSE-HWSN em uma RSSFH, bem como analisar o seu desempenho.

As simulações foram divididos em três grupos. O objetivo do primeiro grupo de simulações é avaliar o *overhead* de POSSE-HWSN. Uma vez que o *framework* introduz um novo componente, é necessário avaliar o tráfego adicional imposto pela proposta. O segundo grupo de simulações tem por objetivo validar o funcionamento de POSSE-HWSN ao utilizá-lo para escolher os nós sensores para um dado conjunto de aplicações. Finalmente, o terceiro grupo de simulações objetiva medir o desempenho do *framework* proposto ao atender diferentes números de aplicações.

5.3.1 Overhead de POSSE-HWSN

Estes testes têm por objetivo mensurar o *overhead* imposto por POSSE-HWSN durante a seleção de nós sensores. Isto foi realizado comparando a quantidade de mensagens trocados entre a aplicação e a RSSF. De acordo com abordagens existentes na literatura (ROWAIHY et al., 2007), para que haja seleção dos nós sensores faz-se necessária apenas a comunicação da aplicação requisitante com o *Sink Node*, o qual seleciona os nós.

Ao utilizar POSSE-HWSN, esta comunicação é realizada através do controlador. Além disso, toda vez que uma requisição chega ao controlador, este precisa requisitar, através do *Sink Node*, os estados de todos os nós sensores. Isto é feito para que o controlador possua as informações mais atualizadas da rede. Este processo tem um custo, já que é necessário o envio de uma mensagem *broadcast* para todos os nós sensores por parte do *Sink Node*, o retorno de suas respectivas respostas e a atualização destas informações na base de dados. Já na literatura, é mais usual os nós sensores enviarem estas informações para o *Sink Node* de forma periódica, independente se há uma requisição ou não (HAMIDA,).

Foi mensurada a quantidade de mensagens de apenas uma aplicação que necessita selecionar 10 nós sensores. Devido o objetivo do teste, o tipo de demanda torna-se irrelevante, pois para quaisquer uma delas faz-se necessária a mesma quantidade de informações trocadas entre POSSE-HWSN e a RSSFH. Além disso, foram realizados testes apenas com uma aplicação, pois quando há aumento no número de aplicações, o número de mensagens torna-se proporcional a quantidade de aplicações. A Tabela 5.2 demonstra um comparativo entre a quantidade de mensagens trocadas durante o procedimento de seleção de nós sensores. A

abordagem da literatura envia apenas uma mensagem para o *Sink Node*, o qual realiza todo o trabalho de seleção. Já utilizando POSSE-HWSN, há uma mensagem de requisição entre a aplicação e o *framework*, uma mensagem de POSSE-HWSN para o *Sink Node* requisitando os estados de todos os nós sensores, 80 mensagens do *Sink Node* para POSSE-HWSN, o que equivale às mensagens de informações de estado dos nós sensores e finalmente uma mensagem novamente de POSSE-HWSN para o *Sink Node* informando os nós sensores escolhidos, totalizando assim 83 mensagens. Apesar desta quantidade de mensagens adicional, a requisição que POSSE-HWSN faz ao *Sink Node* para obter os estados dos nós sensores confere uma dinamicidade maior ao processo. Além disso, caso duas requisições sejam realizadas em um curto intervalo de tempo, a quantidade de mensagens diminui, pois POSSE-HWSN possui informações atualizadas e, consequentemente, está apto a realizar a seleção de nós sensores sem necessidade de disparar as mensagens de solicitação de estados.

Tabela 5.2: Comparativo de mensagens trocadas

Mensagens	App / POSSE-HWSN	<i>Sink</i> / POSSE-HWSN	App / <i>Sink</i>	Total
POSSE-HWSN	1	82	-	83
Literatura	-	-	1	1

Vale salientar que tanto os nós sensores da rede quanto o *Sink Node* utilizados na simulação possuem funcionalidades simples. Apesar de ser possível utilizar esquemas de agregação de dados para diminuir o tráfego na rede, este não é o foco da dissertação.

5.3.2 Testes de validação

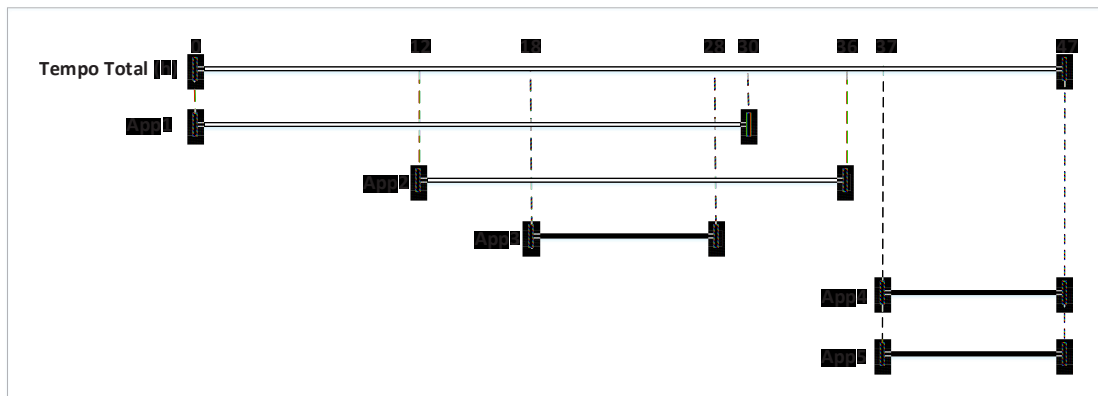
Para a validação de POSSE-HWSN foi criado um cenário experimental simples no qual um módulo de software dispara cinco requisições representando cinco aplicações. A Tabela 5.3 sumariza os parâmetros utilizados para representar cada requisição. Assim, por exemplo, a aplicação *App1* necessita de 10 nós sensores durante 30 horas e estes nós devem fazer algum tipo de pré-processamento (e.g. criptografia).

A Figura 5.3 ilustra a linha do tempo das aplicações requisitantes utilizadas no presente cenário de simulação, mostrando o instante que a aplicação realizou a requisição e a sua duração. Por exemplo, a aplicação *App2* foi iniciada 12 horas após o início da simulação. De acordo com a Tabela 5.3, o tempo de duração desta aplicação será de 24 horas, logo a

Tabela 5.3: Características das aplicações requisitantes

ID	Duração (h)	Demanda	Nº de Sensores
App1	30	Pré-processamento	10
App2	24	Tratamento de imagens	10
App3	10	Armazenamento	10
App4	10	Alta taxa de transmissão	10
App5	10	Economia de recursos	10

aplicação será finalizada 36 horas após o início da simulação.

**Figura 5.3:** Linha do tempo para o cenário utilizado

Nos testes realizados, foram utilizadas as métricas de atraso de configuração e eficácia na seleção de nós sensores. O atraso de configuração refere-se ao intervalo de tempo desde que o controlador recebe a requisição de uma aplicação até o momento em que o controlador seleciona os nós sensores mais apropriados para aquela aplicação. Esta métrica foi utilizada devido a necessidade de demonstrar que, apesar da criação de um novo componente, o tempo necessário para a seleção de nós sensores não inviabiliza o atendimento da aplicação. A métrica de eficácia refere-se à porcentagem dos nós selecionados conforme os requisitos necessários à aplicação. Por exemplo, em uma aplicação que requisita 10 nós sensores e necessita de pré-processamento dos dados coletados, é desejável que sejam selecionados os nós sensores com as maiores capacidades de processamento. Se dentre os 10 selecionados, 3 não possuem as melhores capacidades de processamento, então a eficácia da seleção de nós sensores para esta requisição foi de 70%.

A Figura 5.4 mostra o resultado do teste comparativo entre POSSE-HWSN e a abordagem da literatura para avaliar o atraso na seleção de nós sensores, em que cada teste foi

realizado 10 vezes, obtendo-se a média dos resultados. Como é possível observar, ao utilizar POSSE-HWSN, o atraso de configuração foi cerca de 60ms maior em todos os casos. Este resultado já era esperado, uma vez que POSSE-HWSN funciona como um componente extra entre a aplicação e a rede, interceptando mensagens e realizando o processo de seleção. Mas apesar disto, este tempo adicional não inviabiliza o atendimento das aplicações.

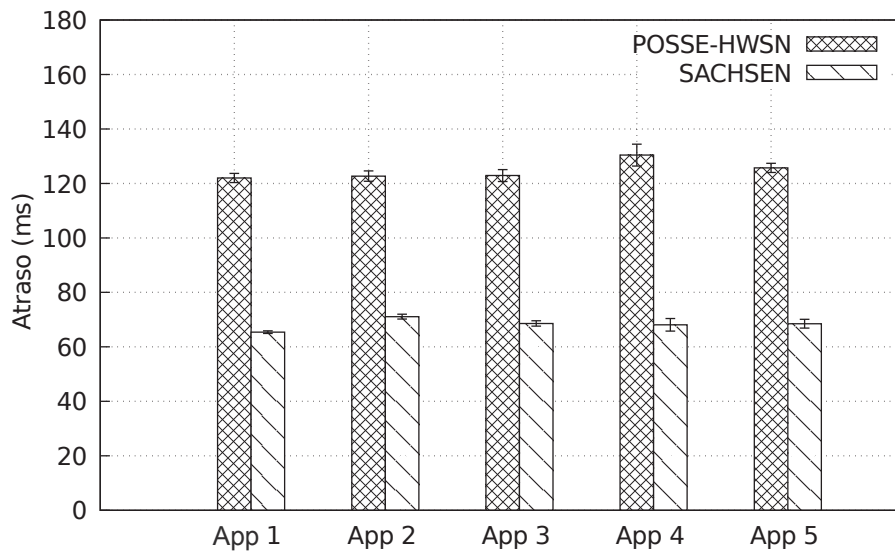


Figura 5.4: Atraso médio na seleção de nós sensores

Já a Figura 5.5 mostra a eficácia do processo de seleção ao utilizar POSSE-HWSN e a abordagem da literatura. Ao observar o gráfico, percebe-se que a eficácia de POSSE-HWSN chegou a 100%. Ou seja, todos os nós sensores escolhidos têm os recursos de hardware mais adequados para cada aplicação. Por exemplo, os 10 nós sensores escolhidos para atender à aplicação *App1*, a qual necessita de pré-processamento, possuem o parâmetro de poder de processamento de 400 Mhz (o mais alto). Já a abordagem da literatura alcançou uma eficácia média de 45%. É importante ressaltar que a eficácia de 100% alcançada por POSSE-HWSN se deve ao fato de existirem poucas aplicações resultantes e ao mesmo tempo vários nós sensores disponíveis. Nos testes de desempenho apresentados a seguir, é mostrado que a eficácia pode diminuir ao se aumentar o número de requisições.

5.3.3 Testes de desempenho

Os testes de desempenho foram realizados para mensurar o comportamento de POSSE-HWSN quando há requisições simultâneas. Para estes testes foram consideradas as mesmas

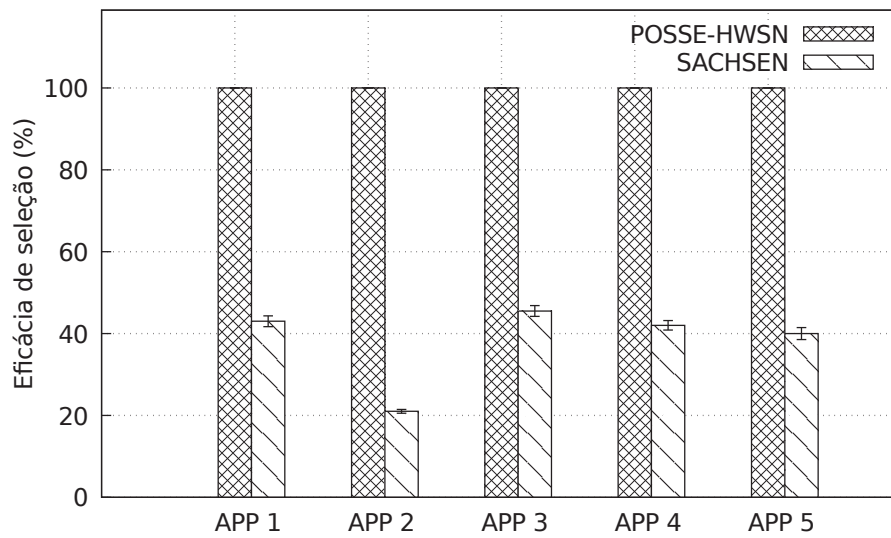


Figura 5.5: Eficácia na seleção de nós sensores

métricas de atraso de configuração e eficácia na seleção de nós sensores. Foram utilizadas requisições simultâneas para verificar o caso em que duas ou mais requisições cheguem em um intervalo de tempo pequeno o suficiente entre elas de forma que a informação que o controlador tem da rede ainda está atualizada. Isto evita que o processo de requisitar os estados dos nós sensores seja disparado.

Como mencionado anteriormente, os parâmetros *appDemand* e *numSensors* enviados pela requisição de cada aplicação são essenciais para o processo de escolha dos nós sensores. Desta forma, os testes de desempenho foram realizados em dois cenários considerando a variabilidade da quantidade de nós sensores requisitados e/ou da probabilidade de uma ou mais demandas serem solicitadas, conforme será explicado em cada cenário.

5.3.3.1 Cenário 1: aplicações que requisitam quantidades iguais de nós sensores

Neste cenário, todas as aplicações requisitam 10 nós sensores. Cada teste mostra a média de dez rodadas e em cada rodada foram executadas 10, 20 e 30 requisições simultâneas, tanto utilizando o *framework* POSSE-HWSN quanto utilizando o SACHSEN. Como toda aplicação necessita de 10 nós sensores, cada nó sensor consegue atender a 4 aplicações diferentes e há 80 nós sensores, então a rede consegue atender um total de 32 aplicações ($80 * 4/10$). Por este motivo foram utilizados no máximo 30 requisições simultâneas.

Além disso, é definida uma probabilidade (*Prob*) de uma das demandas ser utilizada por cada requisição feita ao *framework*, sendo que *Prob* varia de 10% a 90%, com passo

20. Sua utilização funciona da seguinte maneira: ao aplicar *Prob* para uma das demandas, significa que quando houver 10% de probabilidade de que uma aplicação necessite de pré-processamento, a probabilidade de que esta mesma aplicação necessite de alguma outra das 4 demandas é de 90% (22,5% para cada demanda). Quando a probabilidade de que uma aplicação necessite de pré-processamento é de 30%, então a probabilidade de que esta mesma aplicação necessite de alguma outra das 4 demandas é de 70% (17,5% para cada demanda). Esta abordagem de teste foi utilizada para gerar cenários aleatórios de requisições, uma vez que é difícil prever o tipo de demanda que um ambiente de Cidades Inteligentes pode ter que atender e para verificar o quanto a variabilidade de uma demanda pode afetar no desempenho de POSSE-HWSN. Para sortear qual demanda a aplicação irá requisitar (de acordo com *Prob*), foi utilizado um *script* Python, o qual possui as funções randômicas com as suas respectivas personalizações.

A Figura 5.6 apresenta o atraso de configuração ocasionado por POSSE-HWSN ao considerar *Prob* para apenas uma das demandas, variando de 10% a 90% com passo 20. O eixo x dos gráficos representa o número de aplicações simultâneas requisitantes e o eixo y representa o tempo para realizar a seleção em segundos (s).

Ao analisar todos os gráficos que compõem a Figura 5.6, nota-se que o atraso de configuração não aumentou proporcionalmente à quantidade de requisições simultâneas. Isso deve-se às operações de solicitação dos estados dos nós sensores terem sido realizadas apenas para a primeira requisição. Como as requisições são simultâneas, o controlador já possuía as informações atualizadas da rede ao tratar das demais requisições. Além disso, como era de se esperar, o atraso de configuração de POSSE-HWSN é maior quando comparado ao da literatura.

Ao comparar os gráficos em que *Prob* é de 10%, 30% e 50% (Figuras 5.6 (a), (b) e (c)) com os gráficos em que *Prob* é de 70% e 90% (Figuras 5.6 (d) e (e)), também é possível verificar uma leve diminuição no atraso de configuração. Isto deve-se ao fato de que, como a probabilidade de uma requisição escolher a demanda de pré-processamento aumenta, então aumenta a probabilidade de que os nós sensores com os maiores níveis de processamento atinjam mais rápido seu limite máximo de requisições que podem atender. Consequentemente, a lista de nós sensores disponíveis retornada do banco de dados diminui, o que faz diminuir o tempo de escolha, já que há menos elementos para avaliar durante o processo de

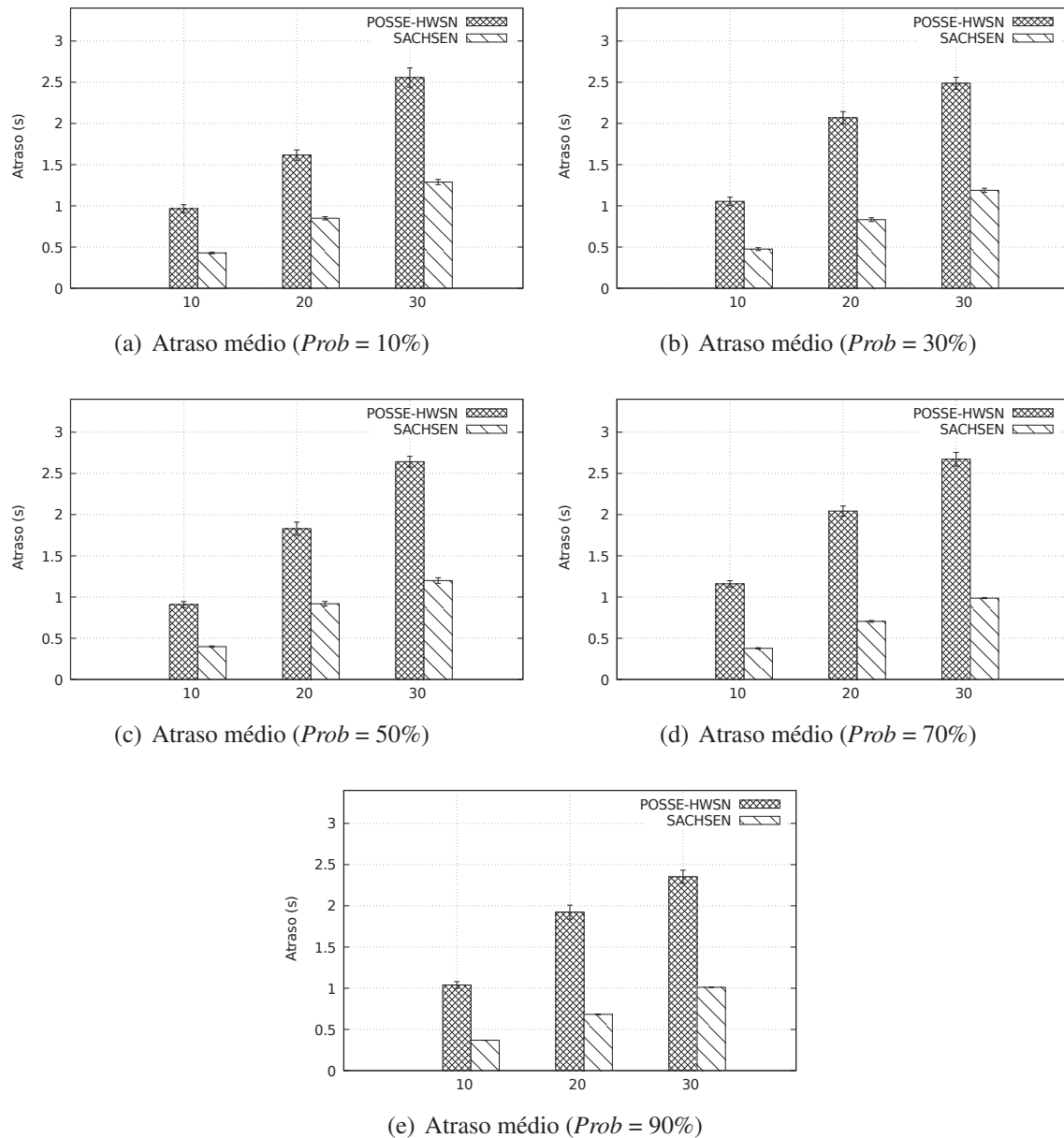


Figura 5.6: Atraso médio do cenário 1 com $Prob$ variável para uma demanda

seleção.

Outra rodada de testes foi realizada neste mesmo cenário, contudo aplicando $Prob$ para duas demandas, no caso pré-processamento e capacidade de armazenamento. A ideia de funcionamento é a mesma dos testes anteriores. Contudo, agora quando a probabilidade de que uma aplicação necessite uma de duas demandas específicas é de 10% (5% para cada demanda), então a probabilidade de que esta mesma aplicação necessite de alguma outra das 3 demandas é de 90% (30% para cada demanda) e assim por diante. A Figura 5.7 mostra o atraso de configuração ocasionado quando $Prob$ é aplicada para as duas demandas.

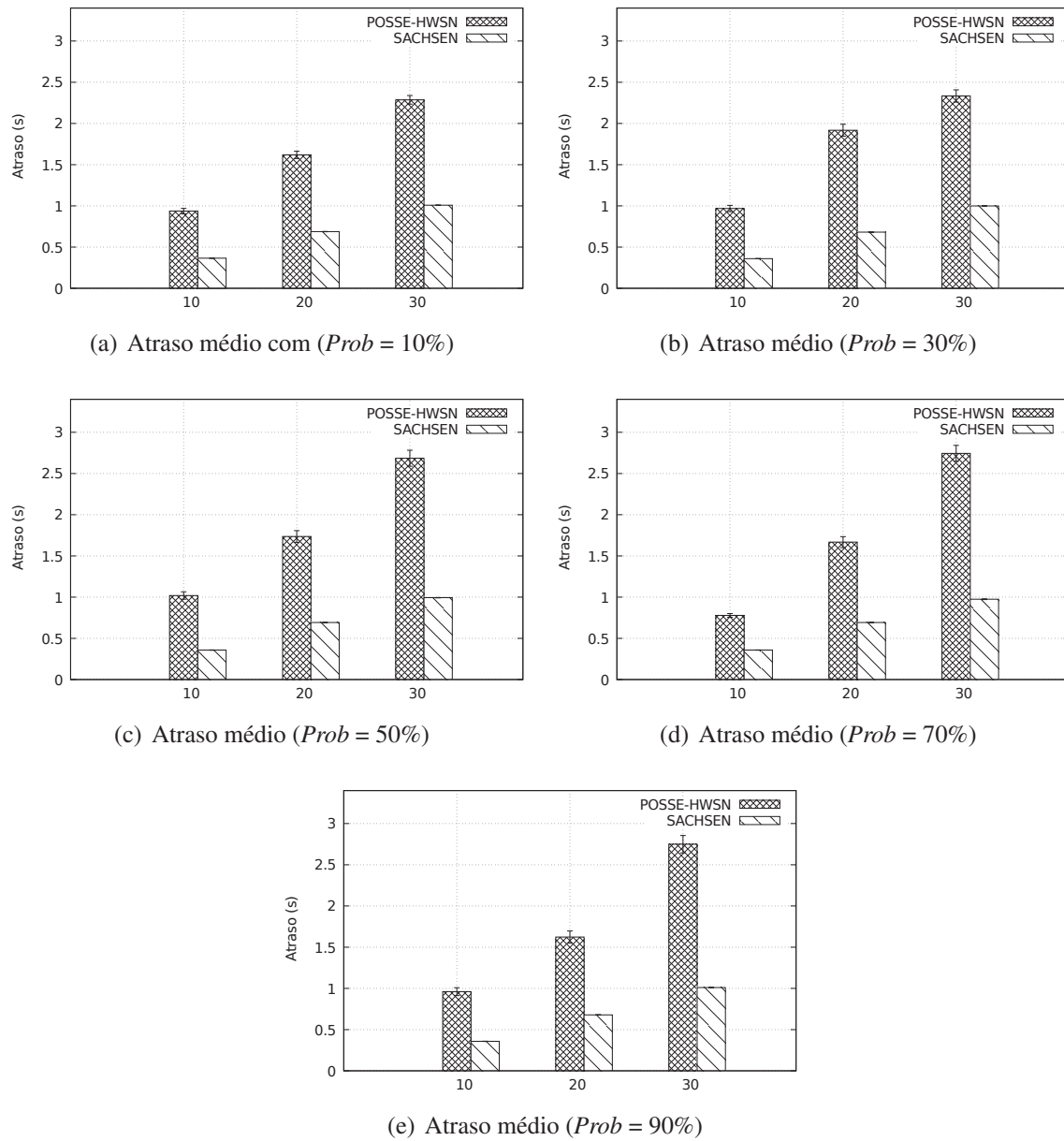


Figura 5.7: Atraso médio do cenário 1 com $Prob$ variável para duas demandas

De forma semelhante aos resultados mostrados na Figura 5.6, aqui nota-se que, apesar de $Prob$ ser aplicado para duas demandas, não houve aumento proporcional do atraso de configuração ao aumentar o número de requisições simultâneas, o que pode ser explicado pelo mesmo motivo. Ou seja, a solicitação do estado dos nós sensores é feita apenas na primeira requisição sendo que nas demais o controlador já possui as informações atualizadas. Repetiu-se também o resultado esperado do atraso de configuração de POSSE-HWSN ser maior quando comparado ao da literatura.

Também verificou-se o mesmo comportamento de ligeira queda no atraso de configura-

ção quando *Prob* é de 70% e 90%, justificado pelo mesmo motivo, ou seja, devido ao grande número de aplicações que requisitam as mesmas demandas os nós sensores atingem sua capacidade máxima e são descartados do processo de análise, gerando assim um atraso menor. Finalmente, ao comparar os gráficos da Figura 5.6 com os gráficos da Figura 5.7, é possível verificar que a variabilidade de probabilidade entre uma ou duas demandas no ambiente ocasiona uma variação de poucos milissegundos, não impactando assim de forma significativa o resultado alcançado.

Além dos testes referentes ao atraso de configuração também foram realizados testes para aferir a eficácia na seleção de nós sensores ao considerar *Prob* para uma e duas demandas. A Figura 5.8 apresenta os resultados de eficácia ao aplicar *Prob* para apenas uma demanda.

Ao analisar os gráficos, é possível verificar que POSSE-HWSN consegue atender até 20 requisições com níveis de eficácia de 100% ou próximo disto, em todos os casos. Ou seja, todos os nós sensores escolhidos (ou quase todos) possuem os melhores recursos de hardware para atender a demanda de cada aplicação. Já quando há 30 requisições simultâneas, POSSE-HWSN tem uma eficácia decrescente conforme *Prob* aumenta. Isto pode ser explicado pelo fato de que quanto maior a probabilidade de uma demanda ser escolhida, maiores as chances dos nós sensores mais indicados para tal demanda atinjam seu número máximo de aplicações que podem atender. Assim sendo, outros nós sensores que não são os mais adequados acabam por serem selecionados. Mas ainda sim, todas as eficácias possuem índices próximo de 80%. Além disso, quando *Prob* é 90% (pior caso) POSSE-HWSN consegue ser cerca de 20% mais eficaz, se comparado à abordagem tradicional, assim é evidente o ganho que POSSE-HWSN proporciona.

A Figura 5.9 apresenta os resultados de eficácia ao aplicar *Prob* para duas demandas. Ao analisar os gráficos da figura, é possível perceber a mesma tendência até 20 requisições verificada no teste anterior, ou seja, índices próximos de 100% de eficácia. Já com 30 requisições, pode-se perceber que, apesar *Prob* ser aplicada para duas demandas, os índices de eficácia permanecem muito próximos dos que foram apresentados na Figura 5.6. Este fato é propiciado devido a capacidade de organização de hardware proporcionado por POSSE-HWSN, ou seja, devido a seleção dos nós ser realizada com base no recurso de hardware necessário a demanda, POSSE-HWSN proporciona uma melhor organização da seleção dos nós disponíveis e, conseqüentemente, melhora a eficácia na seleção de nós sensores. Com

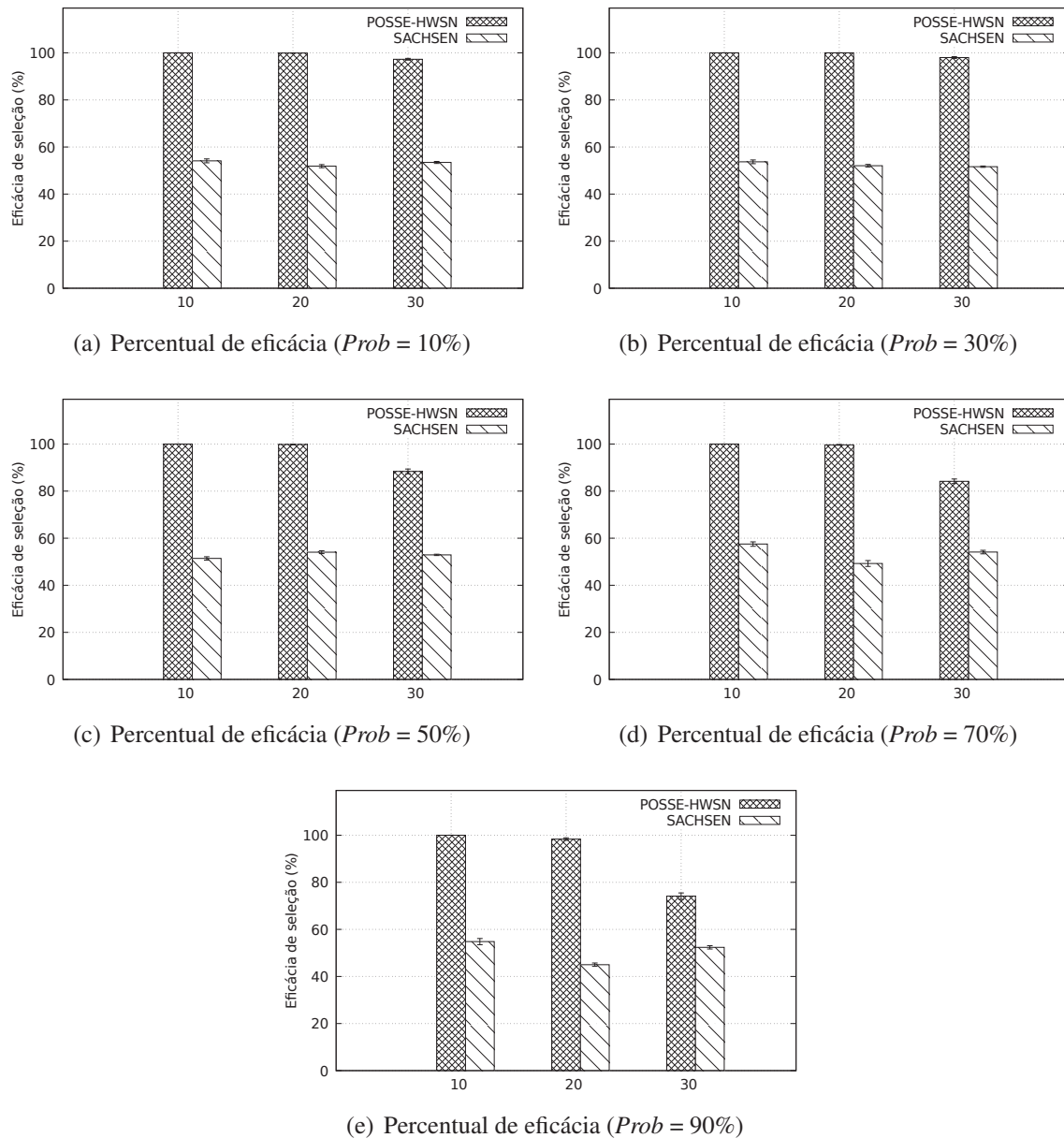


Figura 5.8: Percentual de eficácia do cenário 1 com *Prob* variável para uma demanda

isso, mesmo quando *Prob* é 90% (pior caso) POSSE-HWSN consegue ser cerca de 20% mais eficaz em comparação a SACHSEN.

5.3.3.2 Cenário 2: aplicações que requisitam quantidades aleatórias de nós sensores

Neste cenário, as aplicações podem requisitar entre 5 e 15 nós sensores, com a mesma probabilidade para qualquer valor. Desta forma, pode-se considerar o melhor caso que é quando a rede consegue atender o maior número de aplicações. Considerando que há 80 nós sensores e que cada um pode atender até 4 aplicações, se todas as aplicações requisitarem apenas 5 nós

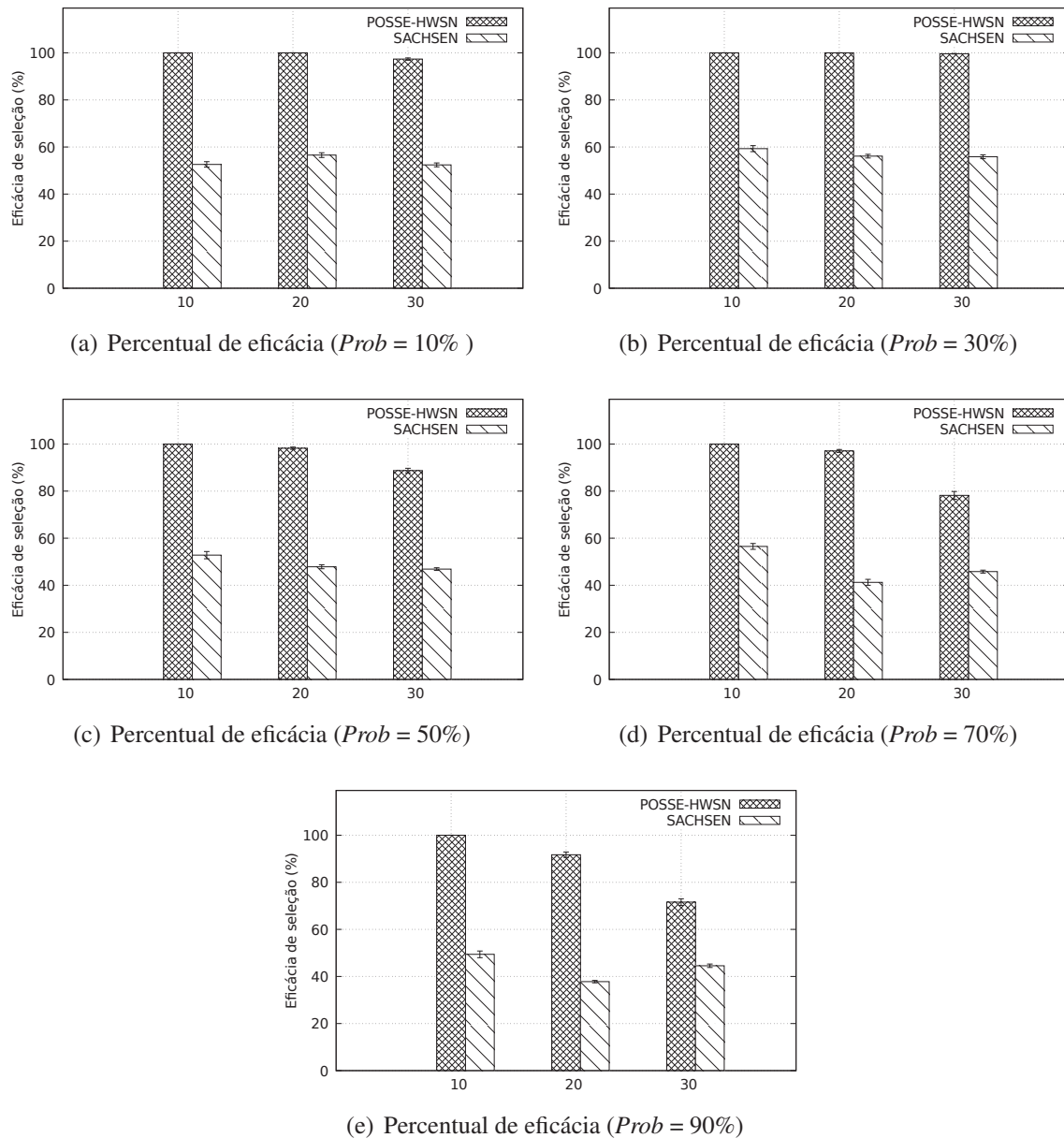


Figura 5.9: Percentual de eficácia do cenário 1 com $Prob$ variável para duas demandas

sensores, então é possível atender até 64 aplicações (60 ao considerar múltiplos de 10 mais próximos do valor). Assim sendo, cada teste neste cenário mostra a média de dez rodadas e em cada rodada foram executadas 10, 20, 30, 40, 50 e 60 requisições simultâneas, tanto utilizando o *framework* POSSE-HWSN quanto utilizando o SACHSEN. Da mesma forma que no cenário 1, os testes deste cenário foram realizados variando a probabilidade de que uma demanda (pré-processamento) seja escolhida e de que uma entre duas demandas específicas (pré-processamento ou armazenamento) seja escolhida.

Devido a variabilidade na quantidade de nós sensores requerida por cada aplicação, pode

acontecer de que algumas aplicações não sejam atendidas. Por causa disto, ao gerar os resultados dos testes, foram consideradas apenas aquelas aplicações que puderam ser atendidas, ou seja, todos os nós sensores requeridos por ela foram selecionados. Quando pelo menos um nó sensor não é selecionado, então a aplicação é considerada como não atendida. A Tabela 5.4 sumariza o percentual de aplicações atendidas ao utilizar *Prob* para uma ou duas demandas. Pela tabela, é possível perceber que a quantidade de aplicações atendidas não variou significativamente entre os dois casos.

Tabela 5.4: Percentual de aplicações atendidas

<i>Prob</i>	Qtd Apps	% de Atendimento (<i>Prob</i> para uma demanda)	% de Atendimento (<i>Prob</i> para duas demandas)
10%	10	100%	100%
	20	100%	100%
	30	98,34%	99%
	40	78,75%	78,5%
	50	60%	61,2%
	60	52,5%	52,05%
30%	10	100%	100%
	20	100%	100%
	30	99%	99%
	40	79,5%	78,5%
	50	62,4%	61,2%
	60	51,66%	52%
50%	10	100%	100%
	20	100%	100%
	30	99,66%	99,67%
	40	79,25%	75,25%
	50	62%	63,78%
	60	52,6%	52,84%
70%	10	100%	100%
	20	100%	100%
	30	100%	99,66%
	40	76,25%	77,5%
	50	63%	62,6%
	60	50,33%	52,17%
90%	10	100%	100%
	20	100%	100%
	30	98,66%	99,67%
	40	77%	76,5%
	50	63,6%	64,6%
	60	52,66%	51%

A Figura 5.10 e a Figura 5.11 mostram os atrasos de configuração ocasionados quando aplicado *Prob* para uma demanda e para duas demandas, respectivamente. Pode-se perceber nos dois casos comportamentos semelhantes aos testes da Subseção 5.3.3.1 i) Os valores de atraso obtidos com POSSE-HWSN são maiores do que os obtidos ao utilizar a abordagem da literatura, o que também já era esperado; ii) Observou-se uma leve diminuição no atraso de configuração conforme aumenta-se *Prob*, o que também é explicado pela diminuição do número de nós sensores disponíveis a se analisar, uma vez que a maioria já atingiu seu número máximo de aplicações que podem atender.

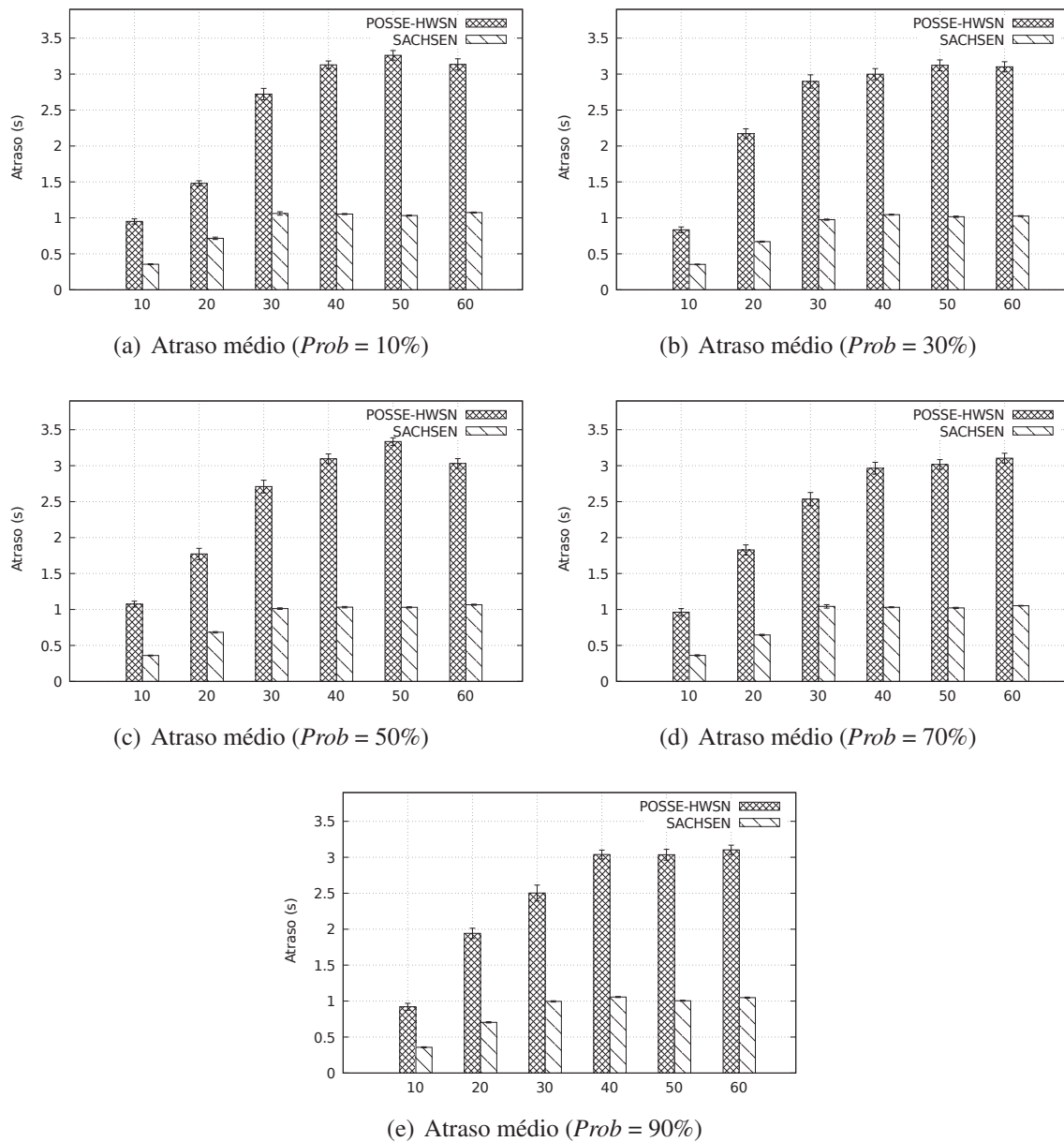


Figura 5.10: Atraso médio do cenário 2 com *Prob* variável para uma demanda

Além disso, ao comparar os resultados das duas figuras, é possível perceber que o fato do ambiente atender aplicações que possuam uma variabilidade considerável de parâmetros (número de nós sensores e demandas), o processo de escolha executado por POSSE-HWSN é afetado de forma mínima.

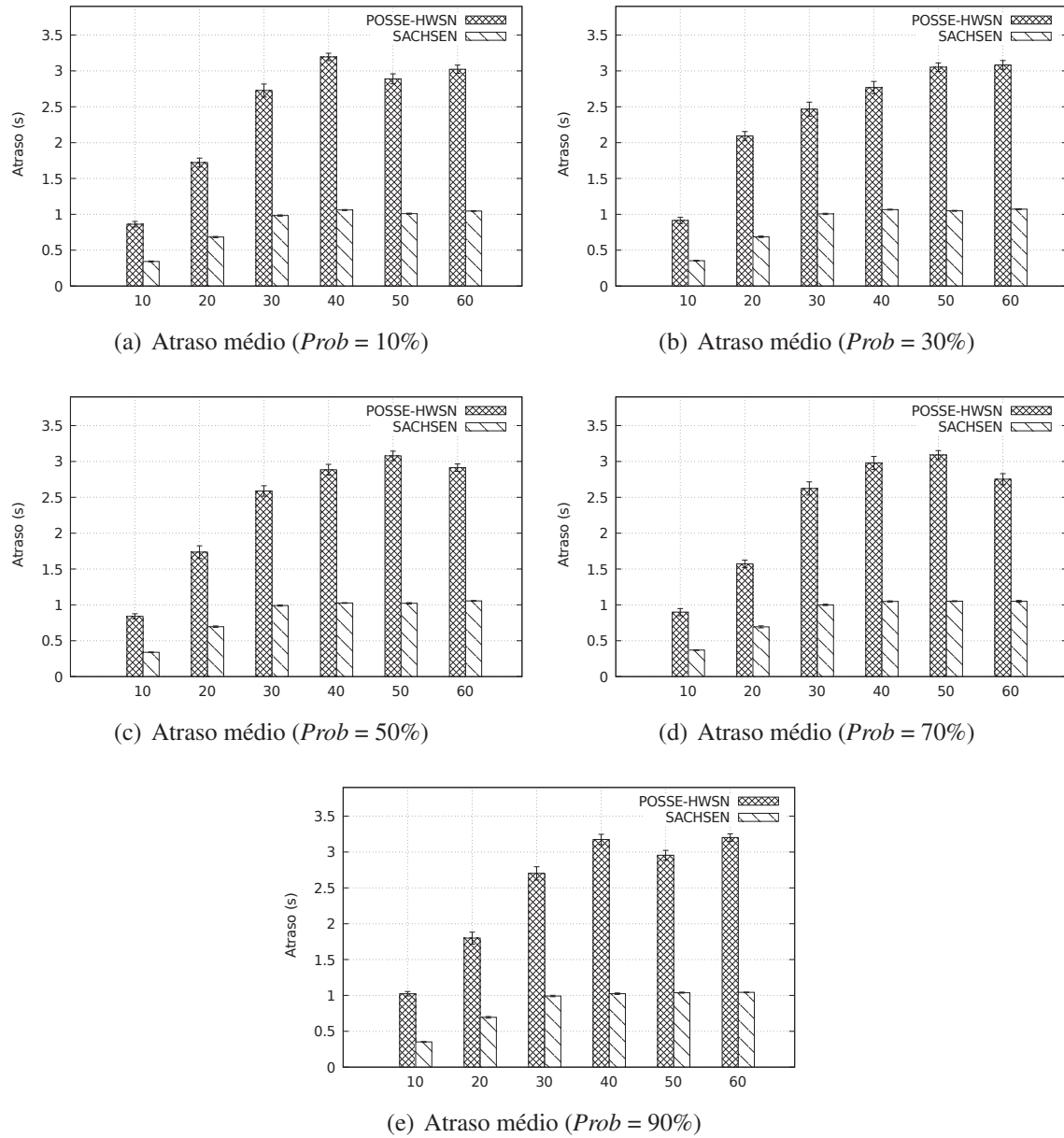


Figura 5.11: Atraso médio do cenário 2 com *Prob* variável para duas demandas

Finalmente, as Figuras 5.12 e 5.13 mostram os resultados de eficácia da seleção ao variar a probabilidade que a rede tem de receber uma demanda específica (pré-processamento) e uma entre duas demandas específicas (pré-processamento ou armazenamento), respectivamente. Da mesma forma que nos testes de atraso de configuração, os resultados destes testes

possuem um comportamento semelhante aos testes de eficácia da Subseção 5.3.3.1. Até 20 requisições simultâneas, a eficácia é de 100% ou próximo disto em todos os casos. Isto é explicado pelo fato de como há um número relativamente pequeno de requisições, então ainda há vários nós sensores disponíveis que são mais adequados a cada aplicação. Conforme aumenta-se o número de requisições, a eficácia tende a diminuir. Contudo, ainda sim a abordagem de POSSE-HWSN tem resultados melhores do que a da literatura.

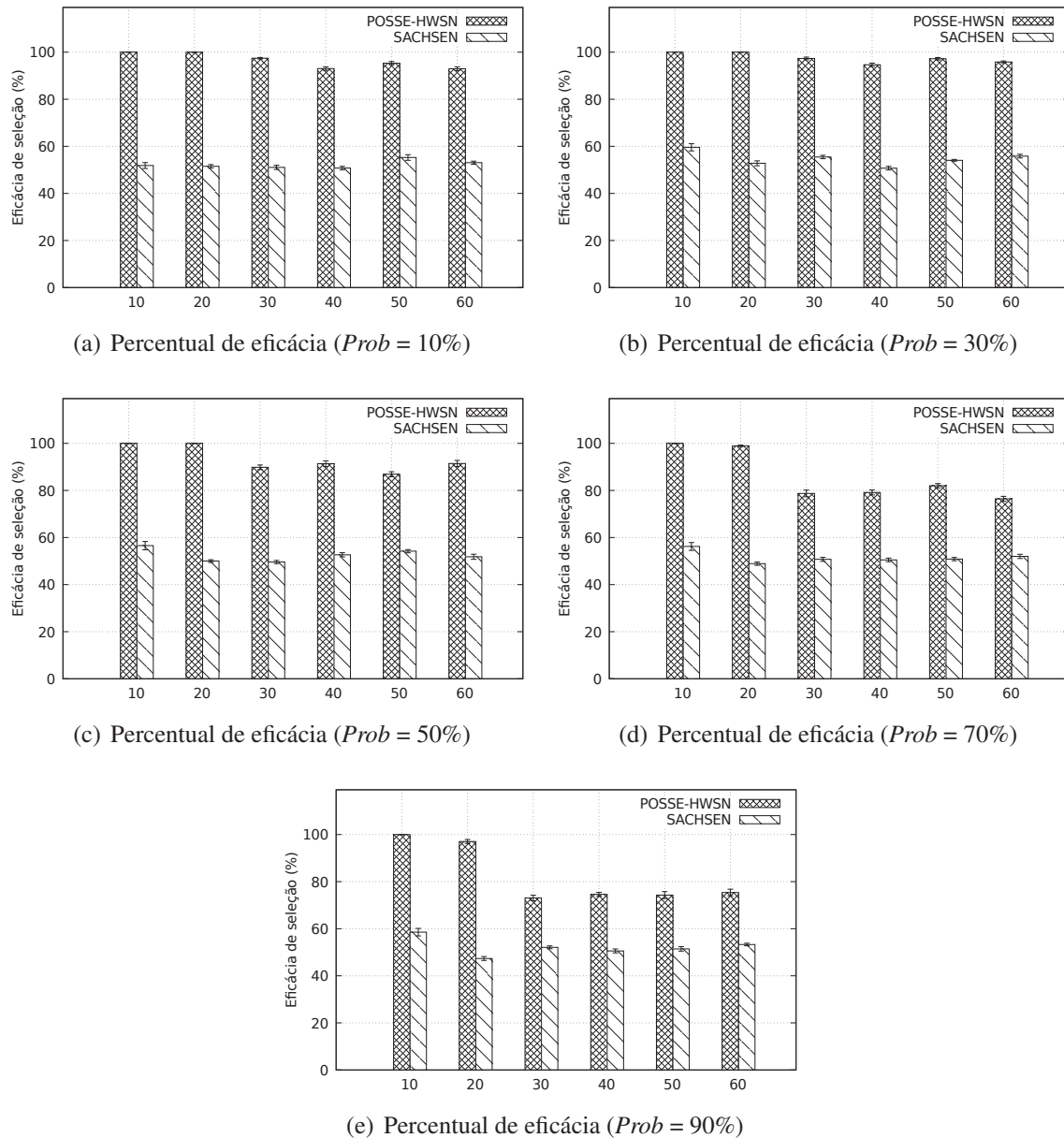


Figura 5.12: Percentual de eficácia do cenário 2 com *Prob* variável para uma demanda

Uma última conclusão que pode-se delinear dos gráficos é que mesmo variando o número de nós sensores requeridos pelas aplicações, os resultados do processo de seleção realizado

por POSSE-HWSN permaceram estáveis e melhores do que a da literatura em todos os casos, confirmando assim o benefício de sua utilização.

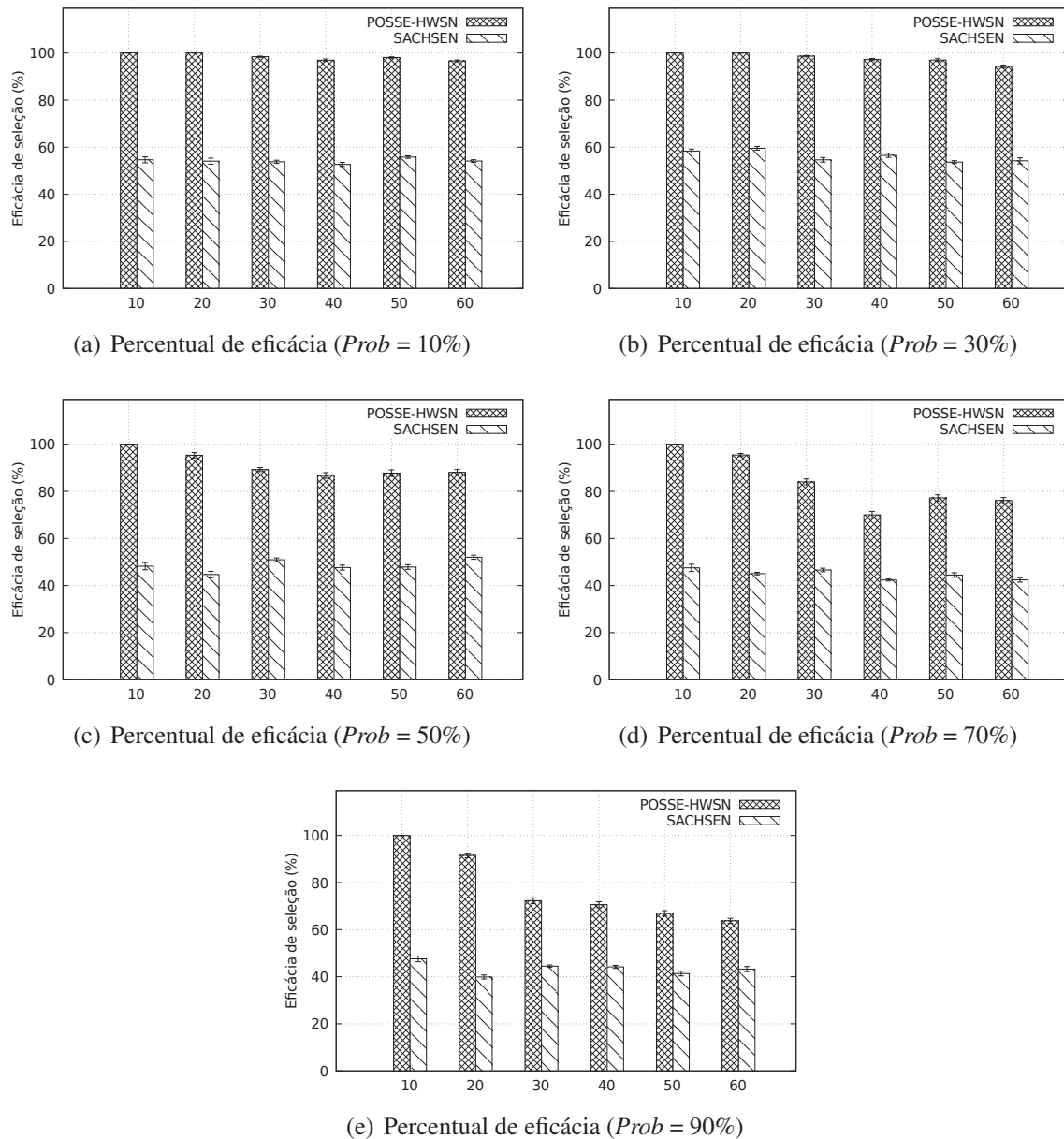


Figura 5.13: Percentual de eficácia do cenário 2 com $Prob$ variável para duas demandas

Ao analisar os gráficos presentes na Figura 5.13 percebe-se que POSSE-HWSN conseguiu ser mais eficaz na seleção de nós sensores em todas as simulações. Além disso, quando tem-se probabilidade de até 50%, POSSE-HWSN conseguiu índices de eficácia próximos de 100%, devido a sua capacidade de organização do *hardware* disponível de acordo com as demandas. Até mesmo no pior caso (quando há probabilidade de 90%), POSSE-HWSN foi capaz de realizar cerca de 15% mais seleções eficazes em comparação a SACHSEN.

Ao analisar todos os gráficos que demonstram a avaliação de POSSE-HWSN utilizando a métrica de eficácia, pode-se concluir que a seleção de nós sensores com base nas demandas das requisições possibilitou maior eficácia no gerenciamento dos recursos de Hardware dos nós sensores, além de possibilitar que as demandas possuíssem recursos de hardware necessários ao seu funcionamento.

5.4 Considerações

Este capítulo apresentou o protótipo de POSSE-HWSN, o ambiente de simulação e os cenários de testes realizados. Além disso, foram discutidos os resultados alcançados, os quais mostraram que, apesar de POSSE-HWSN adicionar atraso extra no processo de seleção de nós sensores, a eficácia da seleção foi melhorada.

Capítulo 6

Conclusão e Trabalhos Futuros

Esta dissertação dedicou-se ao problema de gerência de RSSFH em ambientes de Cidades Inteligentes ao propor um *framework* para seleção dinâmica e sob-demanda de nós sensores. Este *framework*, chamado de POSSE-HWSN, baseia-se no paradigma SDN ao utilizar um controlador com uma visão geral e centralizada da rede. Também utiliza políticas para auxiliar no processo de seleção, conferindo uma maior dinamicidade e flexibilização na proposta, ao permitir alterações das regras de seleção sem que haja necessidade de re-inicialização do sistema.

Ao propor esta solução, foi considerado o potencialmente complexo ambiente de Cidades Inteligentes no qual aplicações dos mais variados tipos com diversas demandas podem realizar requisições. Foi levada em consideração também a heterogeneidade de nós sensores que podem fazer parte destes ambientes. Diferente da grande maioria dos trabalhos, ao desenvolver o mecanismo de seleção, foram utilizados outros parâmetros de hardware, como capacidade de armazenamento, capacidade de transmissão e poder de processamento. Também foi levado em consideração quais os tipos de demandas de cada requisição, permitindo assim o direcionamento do processo no intuito de selecionar os nós sensores com os recursos de hardware mais indicados para cada aplicação.

O protótipo do *framework* POSSE-HWSN foi implementado na linguagem de programação C++ de forma a interagir com o simulador Castalia, o qual foi utilizado para simular a RSSFH. Assim, POSSE-HWSN interceptava as requisições das aplicações à rede, funcionando como um *proxy*. Experimentos foram realizados para validar e avaliar o desempenho do *framework*. Nos testes de desempenho, foram delineados 2 cenários, considerando a

quantidade de nós sensores requeridos por cada aplicação e a probabilidade de atender a uma ou duas demandas específicas.

Como métricas de avaliação, foram utilizadas o atraso e a eficácia da seleção dos nós sensores. Como resultados, pode-se observar que apesar do atraso extra imposto por POSSE-HWSN, quando comparado às técnicas tradicionais, a eficácia na seleção melhorou, permitindo assim obter os nós sensores mais adequados para cada tipo de aplicação.

Como resultado do trabalho realizado na dissertação, foi publicado um artigo no *2015 Brazilian Symposium on Computing Systems Engineering (SBESC)* (JÚNIOR; MATOS, 2015). Além disso, a escrita de um artigo para submissão ao periódico *Sensors* está em andamento.

Como contribuições desta dissertação, pode-se citar: a abordagem de desenvolvimento da proposta, ao utilizar outros parâmetros de hardware, que não energia, considerar as demandas das aplicações durante o processo de seleção de nós sensores, a utilização da métrica de eficácia para avaliar o quão adequada a seleção e criar um processo dinâmico e sob-demanda de seleção; a proposta do *framework* POSSE-HWSN, utilizando políticas e um controlador com uma visão geral e centralizada da rede; e a implementação do *framework* POSSE-HWSN e realização de testes em ambientes simulados.

Como trabalhos futuros, pretende-se:

- Utilizar parâmetros adicionais da aplicação como fator de decisão na seleção de nós sensores, como por exemplo, tempo de execução, frequência de coleta e frequência de envio;
- Utilizar outros parâmetros dos nós sensores como fator de decisão, como por exemplo, o(s) fenômeno(s) físico(s) que o nó sensor consegue monitorar;
- Utilizar políticas mais complexas, levando em consideração outros parâmetros da rede, como por exemplo, posicionamento dos nós sensores;
- Implementar funcionalidade para envio periódico dos níveis de energia dos nós sensores que estão em utilização. Esta funcionalidade permitirá diminuir a quantidade de mensagens trocadas entre POSSE-HWSN e a RSSFH durante o processo de seleção dos nós sensores.

- Utilizar abordagem em que haja redundância do controlador de modo que, caso hajam falhas, a seleção dos nós sensores permaneça ininterrupta.
- Realizar testes do *framework* em ambientes reais.

Bibliografia

- AKBAR, A. H.; IQBAL, A. A.; KIM, K.-H. Binding multiple applications on wireless sensor networks. In: *Advances in Grid and Pervasive Computing*. Springer, 2006. p. 250–258. Disponível em: <http://link.springer.com/chapter/10.1007/11745693_25>.
- AKYILDIZ, I. F. et al. Wireless sensor networks: a survey. *Computer networks*, v. 38, n. 4, p. 393–422, 2002. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S1389128601003024>>.
- ALAJLAN, A. M.; ELLEITHY, K. M. High-level abstractions in wireless sensor networks: Status, taxonomy, challenges, and future directions. In: *American Society for Engineering Education (ASEE Zone 1), 2014 Zone 1 Conference of the*. IEEE, 2014. p. 1–7. Disponível em: <<http://ieeexplore.ieee.org/document/6820645/>>.
- AMMARI, H. M. Coverage in Wireless Sensor Networks: A Survey. *Network Protocols and Algorithms*, v. 2, n. 2, jun. 2010. ISSN 1943-3581. Disponível em: <<http://www.macrothink.org/journal/index.php/npa/article/view/276>>.
- ATZORI, L.; IERA, A.; MORABITO, G. The Internet of Things: A survey. *Computer Networks*, v. 54, n. 15, p. 2787–2805, out. 2010. ISSN 13891286. Disponível em: <<http://dl.acm.org/citation.cfm?id=1862541>>.
- AZIM, A.; ISLAM, M. M. Dynamic service policy-based clustered wireless sensor networks. In: *Wireless and Mobile Computing, Networking and Communications (WiMob), 2010 IEEE 6th International Conference on*. IEEE, 2010. p. 196–202. ISBN 978-1-4244-7742-5. Disponível em: <<http://ieeexplore.ieee.org/document/5644992/>>.
- BELLAVISTA, P. et al. Convergence of MANET and WSN in IoT Urban Scenarios. *IEEE Sensors Journal*, v. 13, n. 10, p. 3558–3567, out. 2013. ISSN 1530-437X, 1558-1748. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6552998>>.
- BOULIS, A. Castalia - A simulator for Wireless Sensor Networks and Body Area Networks - User Manual. In: . [S.l.: s.n.], 2011.
- CARDEI, M.; DU, D.-Z. Improving wireless sensor network lifetime through power aware organization. *Improving wireless sensor network lifetime through power aware organization*, v. 11, n. 3, p. 333–340, 2005. Disponível em: <<http://dl.acm.org/citation.cfm?id=1160098>>.

- CHEN, Z.; KUEHNE, A.; KLEIN, A. Delay constraints for multiple applications in wireless sensor networks. In: *Wireless Communication Systems (ISWCS), 2012 International Symposium on*. IEEE, 2012. p. 126–130. Disponível em: <http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6328343>.
- DAMIANOU, N. et al. The ponder policy specification language. In: *Policies for Distributed Systems and Networks*. Springer, 2001. p. 18–38. Disponível em: <http://link.springer.com/chapter/10.1007/3-540-44569-2_2>.
- DOHLER, M. et al. Smart cities: An action plan. In: *Barcelona Smart Cities Congress*. [s.n.], 2011. Disponível em: <<http://www.fi-ppp-outsmart.eu/en-uk/publications/Publications/Publications/SmartCityPaper.pdf>>.
- FIGUEIREDO, C. M. et al. Policy-based adaptive routing in autonomous WSNs. In: *Ambient Networks*. Springer, 2005. p. 206–219. Disponível em: <link.springer.com/chapter/10.1007%2F11568285_18>.
- GANTE, A. D.; ASLAN, M.; MATRAWY, A. Smart wireless sensor network management based on software-defined networking. In: *Communications (QBSC), 2014 27th Biennial Symposium on*. IEEE, 2014. p. 71–75. Disponível em: <http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6841187>.
- GONÇALVES, N. M.; SANTOS, A. L. D.; HARA, C. S. A Policy-based storage model for sensor networks. In: *2014 IEEE Network Operations and Management Symposium (NOMS)*. IEEE, 2014. p. 1–8. Disponível em: <http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6838317>.
- HAGHIGHI, M. Market-based resource allocation for energy-efficient execution of multiple concurrent applications in wireless sensor networks. In: *Mobile, ubiquitous, and intelligent computing*. Springer, 2014. p. 173–178. Disponível em: <http://link.springer.com/chapter/10.1007%2F978-3-642-40675-1_27>.
- HAMIDA, E. B. Neighbor Discovery Analysis in Wireless Sensor Networks. In: *CoNEXT '06 Proceedings of the 2006 ACM CoNEXT conference*. [s.n.]. ISBN 1-59593-456-1. Disponível em: <dl.acm.org/ft_gateway.cfm?id=1368470>.
- HAN, Z.-j.; REN, W. A Novel Wireless Sensor Networks Structure Based on the SDN. *International Journal of Distributed Sensor Networks*, v. 2014, p. 1–7, 2014. ISSN 1550-1329, 1550-1477. Disponível em: <<http://www.hindawi.com/journals/ijdsn/2014/874047/>>.
- JAMMAL, M. et al. Software defined networking: State of the art and research challenges. *Computer Networks*, v. 72, p. 74–98, out. 2014. ISSN 13891286. Disponível em: <<http://linkinghub.elsevier.com/retrieve/pii/S1389128614002588>>.
- JAVOID, N. et al. EDDEEC: Enhanced Developed Distributed Energy-efficient Clustering for Heterogeneous Wireless Sensor Networks. *Procedia Computer Science*, v. 19, p. 914–919, 2013. ISSN 18770509. Disponível em: <<http://linkinghub.elsevier.com/retrieve/pii/S1877050913007333>>.

- JIANG, F.-C. Mitigation techniques for the energy hole problem in sensor networks using N-policy M/G/1 queuing models. In: *IET International Conference*. [s.n.]. Disponível em: <<http://ieeexplore.ieee.org/document/5632268/?reload=true&arnumber=5632268>>.
- JÚNIOR, F.; MATOS, F. SDN-based Approach to Select Allocation Strategies in Heterogeneous Wireless Sensor Networks. In: *Brazilian Symposium on Computing Systems Engineering*. [s.n.], 2015. ISBN 978-1-5090-0182-8. Disponível em: <<http://ieeexplore.ieee.org/document/7423207/?reload=true&arnumber=7423207>>.
- KAPOOR, N. K.; NANDY, B.; MAJUMDAR, S. Dynamic allocation of sensor nodes in wireless sensor networks hosting multiple applications. In: *Performance Evaluation of Computer and Telecommunication Systems (SPECTS), 2013 International Symposium on*. IEEE, 2013. p. 51–57. ISBN 978-1-56555-351-4. Disponível em: <<http://ieeexplore.ieee.org/document/6595741/>>.
- LARA, A.; KOLASANI, A.; RAMAMURTHY, B. Network Innovation using OpenFlow: A Survey. *IEEE Communications Surveys & Tutorials*, v. 16, n. 1, p. 493–512, 2014. ISSN 1553-877X. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6587999>>.
- LI, W. et al. Efficient allocation of resources in multiple heterogeneous Wireless Sensor Networks. *Journal of Parallel and Distributed Computing*, v. 74, n. 1, p. 1775–1788, jan. 2014. ISSN 07437315. Disponível em: <<http://linkinghub.elsevier.com/retrieve/pii/S0743731513002104>>.
- LIANG, J.; WANG, Z. Adaptive sensor selection for multitarget detection in heterogeneous sensor networks. In: *Fuzzy Systems and Knowledge Discovery (FSKD), 2010 Seventh International Conference on*. IEEE, 2010. v. 2, p. 972–976. Disponível em: <http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5569156>.
- LUO, T.; TAN, H.-P.; QUEK, T. Q. S. Sensor OpenFlow: Enabling Software-Defined Wireless Sensor Networks. *IEEE Communications Letters*, v. 16, n. 11, p. 1896–1899, nov. 2012. ISSN 1089-7798. Disponível em: <<http://ieeexplore.ieee.org/document/6324377/>>.
- MAMUN, M. S. I. et al. Policy based intrusion detection and response system in hierarchical WSN architecture. 2012. Disponível em: <<https://arxiv.org/pdf/1209.1678>>.
- MATTHYS, N.; JOOSEN, W. Towards policy-based management of sensor networks. In: *Proceedings of the 3rd international workshop on Middleware for sensor networks*. ACM, 2008. p. 13–18. Disponível em: <<http://dl.acm.org/citation.cfm?id=1462701>>.
- MISHRA, M. et al. GECSA: A game theory based energy efficient cluster-head selection approach in Wireless Sensor Networks. In: *2015 International Conference on Man and Machine Interfacing (MAMI)*. IEEE, 2015. p. 1–5. Disponível em: <http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=7456580>.
- MIT. 10 Emerging Technologies That Will Change the World. *10 Emerging Technologies That Will Change the World*, 2003. Disponível em: <<http://www2.technologyreview.com/news/401775/10-emerging-technologies-that-will-change-the/>>.

- PAWGASAME, W. A survey in adaptive hybrid wireless Sensor Network for military operations. In: *2016 Second Asian Conference on Defence Technology (ACDT)*. IEEE, 2016. p. 78–83. Disponível em: <http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=7437647>.
- PERILLO, M. A.; HEINZELMAN, W. B. Optimal sensor management under energy and reliability constraints. In: *Wireless Communications and Networking, 2003. WCNC 2003. 2003 IEEE*. IEEE, 2003. v. 3, p. 1621–1626. Disponível em: <http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1200629>.
- QWASMI, N.; LISCANO, R. TinyPolicy: A distributed policy based management framework for Wireless Sensor Networks. In: *Integrated Network Management (IM), 2015 IFIP/IEEE International Symposium on*. IEEE, 2015. p. 918–921. Disponível em: <http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=7140409>.
- RADI, M. et al. Multipath Routing in Wireless Sensor Networks: Survey and Research Challenges. *Multipath Routing in Wireless Sensor Networks: Survey and Research Challenges*, p. 181–192, 2012. Disponível em: <<http://dx.doi.org/10.14257/ijfgcn.2013.6.6.19>>.
- RASTEGARNIA, A.; SOLOUK, V. Performance evaluation of castalia wireless sensor network simulator. In: *Telecommunications and Signal Processing (TSP), 2011 34th International Conference on*. IEEE, 2011. p. 111–115. Disponível em: <http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6043761>.
- ROSS, T. J. *Fuzzy logic with engineering applications*. 3rd ed. ed. Chichester, U.K: John Wiley, 2010. OCLC: ocn430736639. ISBN 978-0-470-74376-8.
- ROWAIHY, H. et al. A survey of sensor selection schemes in wireless sensor networks. In: *Defense and Security Symposium*. International Society for Optics and Photonics, 2007. p. 65621A–65621A. Disponível em: <<http://proceedings.spiedigitallibrary.org/proceeding.aspx?articleid=1302622>>.
- ROY, N.; RAJAMANI, V.; JULIEN, C. Supporting multi-fidelity-aware concurrent applications in dynamic sensor networks. In: *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2010 8th IEEE International Conference on*. IEEE, 2010. p. 43–49. Disponível em: <http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5470601>.
- SANGWAN, A.; SINGH, R. P. Survey on Coverage Problems in Wireless Sensor Networks. *Wireless Personal Communications*, v. 80, n. 4, p. 1475–1500, fev. 2015. ISSN 0929-6212, 1572-834X. Disponível em: <<http://link.springer.com/10.1007/s11277-014-2094-3>>.
- SCHAFFERS, H. et al. Smart Cities and the Future Internet: Towards Cooperation Frameworks for Open Innovation. *Future internet assembly*, v. 6656, n. 31, p. 431–446, 2011. Disponível em: <<http://link.springer.com/content/pdf/10.1007/978-3-642-20898-0.pdf#page=423>>.

- SILVA, W. M. da et al. Smart cities software architectures: a survey. In: *Proceedings of the 28th Annual ACM Symposium on Applied Computing*. ACM, 2013. p. 1722–1727. ISBN 978-1-4503-1656-9. Disponível em: <dl.acm.org/citation.cfm?id=2480688>.
- TAO, Y.; ZHANG, S. Dormancy Scheduling Algorithm Based on Node's Self-Adaptive Density in WSN. In: *NCM 2009: 5th International Joint Conference on INC, IMS and IDC*. IEEE, 2009. p. 494–500. ISBN 978-1-4244-5209-5. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5331672>>.
- VARGA, A.; OPENSIM. OMNeT++ - User Manual. In: . [s.n.], 2011. Disponível em: <<https://omnetpp.org/doc/omnetpp/manual/>>.
- XIAO, M.; CHEN, G.; XIAO, D. A Policy-Based Energy Efficient Clustering Scheme for Wireless Sensor Networks. In: *Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing*. IEEE, 2007. p. 689–694. ISBN 978-0-7695-2909-7. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4287771>>.
- ZHANG, W.; XU, H. A Policy Based Wireless Sensor Network Management Architecture. In: *Intelligent Networks and Intelligent Systems (ICINIS)*. IEEE, 2010. p. 552–555. ISBN 978-1-4244-8548-2. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5693764>>.
- ZHI-YAN, G.; JIAN-ZHEN, W. Research on coverage and connectivity for heterogeneous wireless sensor network. In: *Computer Science & Education (ICCSE), 2012 7th International Conference on*. IEEE, 2012. p. 1239–1242. Disponível em: <<http://ieeexplore.ieee.org/document/6295289/>>.