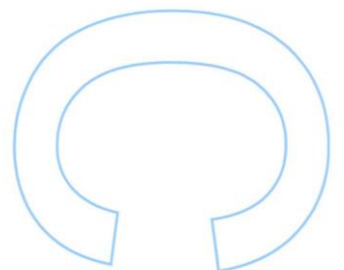
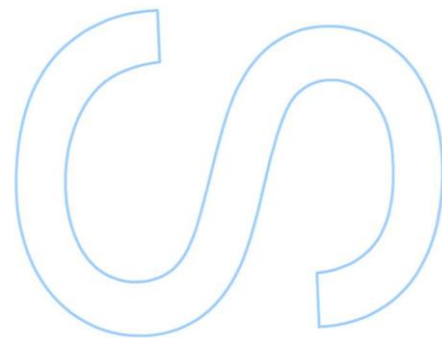
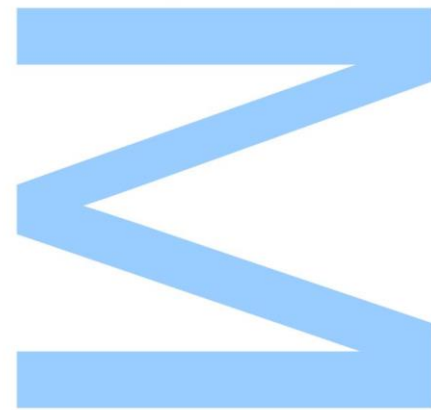


Cobertura, Conectividade e Requisitos de uma Rede de Sensores sem Fios



Vítor Daniel Peixoto de Sousa

Mestrado integrado em Engenharia de Redes e Sistemas Informáticos
Departamento de Ciência de Computadores
2013/2014

Orientador

Engº Gil Gonçalves, Grupo de Engenharia da Decisão e Controlo ISR - FEUP

Coorientador

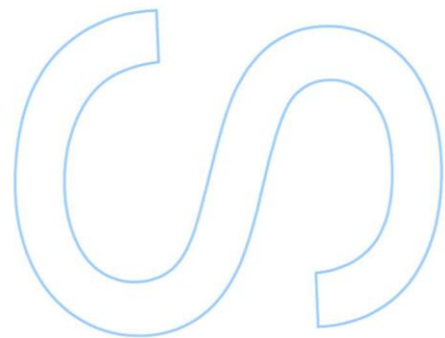
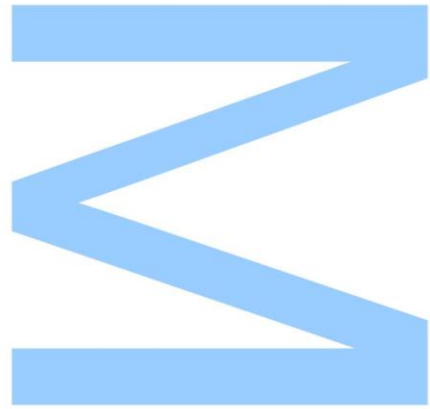
Engº João Reis, LSTS - FEUP



Todas as correções determinadas pelo júri, e só essas, foram efetuadas.

O Presidente do Júri,

Porto, ____ / ____ / ____



**Aos meus pais,
António Sousa e Olga Peixoto**

Agradecimentos

Agradeço a todos os que contribuíram e me incentivaram para o desenvolvimento deste projeto, a toda a equipa do laboratório LSTS, agradeço a boa receção e o bom ambiente prestado ao longo de todo o estágio.

Quero deixar um agradecimento especial ao Professor Engenheiro Gil Gonçalves pela orientação e por todos os conhecimentos partilhados, fica também um agradecimento especial ao Engenheiro João Reis e ao meu amigo Luis Neto pela disponibilidade e apoio no esclarecimento de todas dúvidas durante o desenrolar da dissertação.

Por fim, mas não menos importante, agradeço à minha família, namorada e amigos mais próximos o afeto e o apoio incondicional demonstrados.

Um muito obrigado.

Resumo

O tema redes de sensores sem fios tem sido frequentemente abordado por diversas entidades devido ao seu uso intensivo nos dias de hoje. O futuro da computação e da comunicação depende da evolução e inovação deste tipo de redes que atualmente são utilizadas com diferentes propósitos no nosso dia-a-dia com o conceito de *Internet of Things*, na área ambiental, militar e entre muitas outras, na área e setor Industrial.

Esta dissertação descreve o trabalho realizado ao longo de largos meses e consistiu na elaboração de um serviço no âmbito do setor industrial, com funcionalidades que permitem atuar em fases importantes de um sistema de produção reconfigurável nomeadamente a fase de controlo e de configuração, contribuindo ainda com uma ajuda importante na fase de planeamento de uma rede de sensores sem fios. A cobertura, a conectividade e os requisitos de uma rede de sensores sem fios são o foco principal deste serviço, sendo o seu objetivo primário obter uma configuração de uma RSSF para uma determinada área, de maneira a alcançar uma cobertura ótima da mesma e garantir a conectividade da rede, distribuindo os dispositivos em locais específicos. Para alcançar este objetivo, contribuiu o fato do serviço estar encapsulado como um *bundle* na arquitetura *SmartComponent*, que lhe permite ter acesso e interagir diretamente com os serviços que estão disponíveis na rede, faz com que possa calcular os seus resultados utilizando parâmetros obtidos diretamente dos dispositivos registados no sistema.

O resultado foi um serviço capaz de tratar todos os dados importantes e gerar uma solução com a garantia de obtenção de informação fidedigna e em tempo real de todos os atores da RSSF, servirá não só para obter resultados rápidos e mais fiáveis, como também ajudará na análise dos requisitos pretendidos para a aplicação, com a verificação da precisão dos serviços prestados bem como o raio de alcance dos sensores utilizados. No contexto de produção de uma linha industrial, determinismo em relação a decisões de controlo sobre uma célula de produção, é uma característica fundamental. Este trabalho pretende contribuir nesse âmbito, reforçando a confiança nas decisões de controlo, por sua vez tornando ainda mais próximo o objetivo de conseguir uma *SmartFactory*.

Palavras-chave: Rede de Sensores sem Fios, Cobertura, Conectividade.

Abstract

The wireless sensor networks (WSN) theme has been frequently addressed by several entities due to its intensive use nowadays. The usefulness and functionality provided by these network systems, in such diverse and wide areas and sectors, is what made them so popular. The future of both computing and communication depend on the evolution and innovation of these types of networks which are currently used with different purposes, namely in the environmental and military areas and also in industrial applications.

This dissertation describes the work of several months, which consisted in the development of an industrial sector service with functionalities that allow the user to reconfigure the production system during important phases, namely the configuration and control phase, while also providing significant help in the planning phase of a wireless sensor network. The WSN's coverage, connectivity and its requirements are the main focus of this service and its prime objective is to obtain an RSSF configuration for a given area, so that its coverage is optimal, and network connectivity is guaranteed by placing the devices in specific locations. The fact that the service was encapsulated as a bundle in the SmartComponent architecture allowed it to have access to, and interact directly with, the services available in the network, thus making it possible for the results to be calculated using parameters obtained directly from the devices registered in the system.

The result was an app/service, capable of processing data, and the development of a solution guaranteed to gather reliable real time information from every component of the WSN, not only obtaining faster and more reliable results but also providing help in the analysis of the app's prerequisites through the verification of the service's accuracy as well as the range of the sensors used. In the industrial line context, accuracy is essential in regards to control decisions of a production cell. This paper aims to contribute in that scope, reinforcing trust in control decisions, which in turn aims to enhance a SmartFactory development.

Keywords: Wireless Sensor Network, Coverage, Connectivity.

“Coming together is a beginning; keeping together is progress; working together is success”

Henry Ford

Conteúdo

Agradecimentos	5
Resumo	6
Abstract	7
Lista de Tabelas	12
Lista de Figuras	14
1 Introdução	15
1.1 Contexto	16
1.2 Motivação e Objetivos	17
1.3 Estrutura da dissertação	19
2 Problema	20
2.1 Contexto Industrial	20
2.1.1 Reconfigurable Manufacturing System (RMS)	20
2.1.2 I-RAMP3	21
2.1.3 SelSus	21
2.2 Requisitos	22
3 Estado da Arte	24
3.1 Redes de sensores sem fio (RSSF)	24
3.1.1 Áreas e setores de aplicação	25
3.1.2 Características das RSSF	26
3.1.2.1 Configuração	26
3.1.2.2 Monitorização	27
3.1.2.3 Comunicação	27
3.1.2.4 Processamento	29
3.1.2.5 Energia	29
3.1.3 Componentes, padrões e tecnologias	30

3.1.3.1	Dispositivos	30
3.1.3.2	Padrões	32
3.1.3.3	Sistemas Operativos	33
3.2	Cobertura e conectividade de uma RSSF	34
3.2.1	Modelos de Cobertura	35
3.2.2	Modelos de Conectividade	38
3.2.3	Controlo de cobertura e conectividade	39
3.2.4	Mobilidade	40
3.2.5	Distribuição e organização dos dispositivos	41
4	Desenho e Implementação	44
4.1	Tecnologias	45
4.1.1	JAVA	45
4.1.2	Universal Plug and Play - UPnP	45
4.1.3	Open Service Gateway initiative - OSGi	46
4.1.4	JAXB	46
4.2	Implementação	47
4.2.1	Fase de Planeamento	47
4.2.1.1	Vizinhos	52
4.2.2	Fase de distribuição	53
4.2.2.1	Tipos de Movimento	53
4.2.2.2	Prioridades	54
4.2.2.3	Regras para resolução de problemas com fronteiras	55
4.2.2.4	Código do Algoritmo	60
4.3	Enquadramento do serviço na arquitetura - Comunicação	63
4.3.1	Arquitetura SmartComponent	63
4.3.2	Universal Plug and Play	64
4.3.3	Device Integration Language	64
4.3.3.1	NetDev Service Description	65
4.3.3.2	Task Description Document	66
4.3.3.3	Quality Result Document	67
4.3.3.4	Task Fullfilmente Document	68
4.4	Métodos, funções, output	69
4.4.1	Cobertura de área	69
4.4.2	Cobertura de área com Obstáculos	70
4.4.3	Verificação de Requisitos	70
4.4.4	Comunicação DIL	71
5	Testes e Resultados	72

6 Conclusões e trabalho futuro	80
A Acrónimos	82
B Ficheiro NSD	83
Referências	86

Lista de Tabelas

3.1	Fluxo de informação. Comunicação em redes de sensores sem fios. [1]	28
4.1	Posições normais dos seis vizinhos de um dispositivo	52

Lista de Figuras

3.1	Componentes do dispositivo	30
4.1	Planeamento da RSSF quando $R_s > R_c$	48
4.2	Planeamento da RSSF quando $R_s = R_c$	49
4.3	Planeamento da RSSF quando $R_s < R_c \leq \sqrt{3}R_s$	50
4.4	Planeamento da RSSF quando $R_c > \sqrt{3}R_s$	51
4.5	Vizinhos de um dispositivo, Distribuição em forma de hexágono	52
4.6	Tipos de movimento usados	53
4.7	Distribuição de dispositivos sobre uma área	54
4.8	Prioridades	55
4.9	Representação de obstáculos	56
4.10	Resolução de problema na fronteira lateral	56
4.11	Resolução de problema na fronteira lateral de obstáculos	57
4.12	Regra fronteira da área para movimentos tipo 3 e 4	58
4.13	Regra fronteira do obstáculo para movimentos tipo 3 e 4	59
4.14	Regra fronteira da área para movimentos tipo 5 e 6	59
4.15	Regra fronteira do obstáculo para movimentos tipo 5 e 6	60
4.16	Prioridades de movimentos efetuados na distribuição de dispositivos	60
4.17	Comunicação via UPnP - Device Spy	64
4.18	Parte do Ficheiro NetDev Service Description	65
4.19	Parte do Ficheiro NetDev Service Description	66
4.20	Ficheiro Task Description Document	67
4.21	Ficheiro Quality Result Document	67
4.22	Ficheiro Task Fullfilmente Document	68
5.1	Distribuição quando $R_s = R_c$	74
5.2	Distribuição quando $R_s < R_c \leq \sqrt{3}R_s$	75
5.3	Distribuição quando $R_c > \sqrt{3}R_s$	76
5.4	Distribuição quando $R_s > R_c$	77
5.5	Distribuição sobre obstáculos	78
5.6	Distribuição sobre obstáculos simples	79

5.7	Distribuição sobre obstáculo composto	79
-----	---	----

Capítulo 1

Introdução

As redes de sensores sem fios (RSSF) são uma das tecnologias mais importantes de momento e estão cada vez mais presentes no nosso quotidiano. A sua aplicabilidade está ligada ao limite da nossa imaginação e, graças a sua escalabilidade, modularidade e possibilidade de uma panóplia de funcionalidades, estas redes são hoje utilizadas para desempenhar um enorme leque de tarefas e funções.

O fantástico poder que este tipo de redes é capaz de fornecer a qualquer tipo de serviço ou aplicação faz com que seja um dos temas mais investigados na atualidade, e uma tecnologia bastante procurada por imensas entidades nas mais variadas áreas e setores. A investigação e a indústria de vários países atribuem uma grande importância às redes de sensores sem fios, que é considerada como uma das mais influentes tecnologias do séc. XXI, no entanto, por vezes é ainda necessário ultrapassar uma certa barreira mental que poderá existir no que toca à utilização de redes de sensores sem fios aliada à indústria. Apesar de tudo, as RSSF ainda não são uma abordagem direta quando falamos no setor industrial, ou seja, estes sistemas ainda não são a primeira escolha para a resolução dos problemas associados ao *shop-floor*.

Vários aspetos e atributos fazem este tipo de redes um pouco diferentes das redes tradicionais de computadores, sejam esses de métrica, de característica ou de finalidade. As redes de sensores sem fios têm a capacidade de operar de forma autónoma, sendo apenas necessário a intervenção humana para tarefas de manutenção específicas. Possuem grandes restrições quanto ao processamento, memória e energia, têm uma topologia dinâmica e por isso devem prover mecanismos automatizados de modo a gerir a rede garantindo a sua configuração, manutenção, organização e proteção.

Torna-se importante um estudo aprofundado destas redes em várias vertentes de forma a garantir a máxima robustez e fiabilidade em sistemas que utilizem este tipo de tecnologia. É importante que se analise e configure de maneira adequada serviços que integrem uma rede de sensores sem fios, fornecendo a possibilidade de uma integração rápida e direta em sistemas já implementados.

Este projeto tem como objetivo o estudo da cobertura e conectividade das RSSF, de forma a garantir os requisitos pretendidos por vários tipos de sistemas, pretendendo com isto, garantir um avanço na inovação da indústria, automatização e evolução tecnológica.

1.1 Contexto

Actualmente, na fase de desenho e planeamento de uma RSSF, são escolhidas ferramentas específicas com capacidade para simular uma rede, de modo a que se possa testar e configurar todo o sistema antes da sua implementação. Estas soluções normalmente não têm em consideração as condições do ambiente onde se pretende implementar a rede e todos os dados são simulados, não existindo uma comunicação direta com a própria rede, nem com os seus constituintes, de forma a se efetuar uma configuração dinâmica e automática do sistema.

Existem vários exemplos de implementações de simuladores de RSSF que têm em conta as características dos dispositivos normalmente presentes neste tipo de redes. As suas principais funcionalidades são simulações e avaliações a nível protocolar, e recaem sobre como a rede gere e distribui os dados ao longo do seu tempo de vida. Esta pesquisa visa a integração na arquitetura lógica do sistema de componentes de *software*, que gerem os serviços deste tipo de redes, focando-se principalmente na camada física da rede, na forma como os dispositivos estão distribuídos ao longo de determinada área e nas características dos mesmos. Por este motivo, este projeto difere um pouco dos simuladores existentes, com objetivos e métodos diferentes.

A implementação de uma rede de sensores pode ser efetuada de modo manual, onde os dispositivos são instalados em posições previamente definidas ou de forma aleatória sobre uma área de interesse. O problema da distribuição dos nós de uma rede, tem sido explorado por vários investigadores, que apresentam soluções que têm o objetivo de garantir a cobertura máxima de uma certa região a monitorizar[2]. Apresentamos ao longo do documento as formas de atuar de uma rede de sensores sem fios, bem como alguns algoritmos e métodos estudados para resolver o problema da maximização de cobertura

de uma rede devido ao posicionamento dos seus dispositivos e às características dos mesmos.

Apesar disso, ainda não existem serviços integrados na própria arquitetura da rede, que avaliem as características da área e dos dispositivos disponíveis com o objetivo de apresentar uma solução ótima para a distribuição de nós da rede, tendo em conta os requisitos pretendidos pela RSSF. Ou seja, a existência de serviços com capacidade de disponibilizar, através da rede, informação sobre a melhor topologia e disposição dos nós ao longo da área, de modo a maximizar não só a cobertura, mas também a conectividade.

A facilidade de projetar e planejar uma rede de forma automática e dinâmica pode levar à diminuição de custos e de tempo, requisitos de extrema relevância em todas as áreas e setores, principalmente no que concerne à Indústria. As características do serviço que aqui é proposto fazem com que esta tecnologia se enquadre em projetos que têm em vista a forte inovação da Indústria.

1.2 Motivação e Objetivos

O seguinte trabalho de pesquisa/investigação tem como objetivo, criar um serviço com capacidade de ser integrado diretamente na arquitetura de uma rede de sensores sem fios, de forma a efetuar avaliações em tempo real dos requisitos e dispositivos do sistema. O serviço engloba duas fases inerentes a uma rede de sensores sem fios; a fase de desenho e a fase de operação/manutenção, tem como foco acelerar e automatizar a construção e configuração de uma RSSF conseguindo minimizar os custos e o tempo despendido em todo o processo.

Na fase de planeamento e desenho deste tipo de redes, o serviço tem como objetivo resolver o problema da cobertura máxima de uma determinada área de forma automática, com recurso a algoritmos estudados que combatem este problema. Ou seja, possui a capacidade de formular e sugerir o posicionamento dos dispositivos pela área a monitorizar, de modo a garantir os requisitos de qualidade e fiabilidade pretendidos pelo sistema.

O serviço proposto pretende responder ainda a questões sobre os requisitos do sistema, como por exemplo; verificar se determinada área está totalmente coberta pelos dispositivos que nela estão presentes; verificar se determinado parâmetro (ex. medição de temperatura) está a ser garantido com a precisão requerida ou se é possível garantir determinado parâmetro numa determinada área. Estes são elementos que podem ser bastante úteis

na fase de operação/manutenção do sistema e que pode fornecer informações úteis aos responsáveis por essa tarefa.

Espera-se alcançar a integração do serviço no sistema com ajuda de tecnologias como o OSGi e o UPnP, que fornecem meios para uma comunicação livre e transparente entre todos os dispositivos. O serviço deve então ser capaz de responder adequadamente a pedidos recebidos por parte de outros agentes através da rede. Seguem-se alguns exemplos do tipo de informação que o serviço pode fornecer:

- Sugestão de uma distribuição ótima de dispositivos numa determinada área.
- Informação detalhada sobre todos os dispositivos presentes na rede.
- Verifica a disponibilidade e o estado de dispositivos.
- Informações sobre a QoS de uma rede, ou de parte da rede em particular.
- Verificar os serviços que estão disponíveis.
- Verificar a qualidade dos serviços que a rede dispõe.

Este projeto tem como objetivo simplificar o planeamento e implementação de uma rede de sensores sem fios, com soluções que tornam fácil e transparente a integração e manutenção dos dispositivos, quer seja num sistema já implementado ou numa área que se deseje monitorizar. Este serviço pode ser considerado uma ferramenta útil para diagnosticar, avaliar e configurar uma rede de sensores sem fios. Contudo, as informações que pode garantir, são também, bastante úteis para as ferramentas de diagnóstico das RSSF existentes.

Estas características referidas acima fazem com que o serviço se enquadre em projetos europeus no âmbito da automatização de sensores, monitorização de máquinas industriais, inovação da indústria e equipamento inteligente, nomeadamente os projetos I-RAMP3 [3] e SelSus [4].

1.3 Estrutura da dissertação

Esta dissertação apresenta na sua estrutura os seguintes capítulos organizados da seguinte forma:

Capítulo 1 Introdução: Capítulo que dá início a esta dissertação. Aborda o contexto do projeto e expõe a motivação e o objetivo desta dissertação.

Capítulo 2 Problema: Referimos o problema que nos propomos a resolver e abordamos o contexto industrial em que esta dissertação se enquadra, referenciando os projetos a ela associados.

Capítulo 3 Estado da Arte: Neste capítulo é explorado o tema das redes de sensores sem fios de uma forma geral, expondo as suas características e aplicabilidade. É abordado também o tema da cobertura e conectividade de uma RSSF e os métodos de distribuição de dispositivos numa área de interesse.

Capítulo 4 Desenho e Implementação: Ao longo deste capítulo explicamos o nosso projeto com mais pormenor, evidenciando as tecnologias usadas, o algoritmo contruído e o seu método de funcionamento assim como os resultados esperado.

Capítulo 5 Testes: Apresentados os testes efetuados e os resultados obtidos.

Capítulo 6 Conclusões e trabalho futuro: Capítulo final desta dissertação onde se discute os resultados obtidos, bem como o trabalho futuro a ter em consideração.

Capítulo 2

Problema

2.1 Contexto Industrial

Projetos europeus no âmbito da automatização das operações de manutenção e aceleração da fase de arranque operacional contextualizam os objetivos deste trabalho, tendo em conta o forte crescimento da área e a forte inovação da Indústria com recurso a equipamento inteligente. Os projetos cofinanciados pela Comissão Europeia no âmbito do 7º Programa Quadro (FP7), tratam de uma mudança no paradigma dos sistemas de produção industrial ao nível do *shop-floor*, que representa a parte operacional da empresa/fábrica.

2.1.1 Reconfigurable Manufacturing System (RMS)

Um sistema de produção industrial tem identificadas algumas variáveis que tem um impacto acrescido na sua eficiência e funcionamento. Dois parâmetros que têm particular influência num sistema deste tipo são a demanda de um produto e a sua variedade. Um RMS é uma classe de sistema de manufatura que tem como objetivo alcançar um elevado nível de produtividade em células de trabalho automatizadas na produção de produtos. Para se conseguir imunidade ao decréscimo de eficiência do sistema, uma vez influenciado pelas variáveis mencionadas, são adotadas um conjunto de estratégias que resultam num sistema de produção reconfigurável [31].

Uma das vantagens de um RMS é a redução do tempo de RAMP-UP, de modo a que seja reduzido o tempo de iniciação e conclusão de uma tarefa de produção. A variação do volume do produto deverá ter implicações mínimas na eficiência da linha de produção. O sistema deverá evitar paragens imprevistas, causadas por falhas nas máquinas. Deverá também ser conseguida a sobreposição dos processos das três fases de produção, dese-

nho, fabrico e montagem. O que se coaduna com os objetivos de manutenção inteligente e rápida delegação de processos de produção. Um RMS tem uma capacidade de reconfigurar os recursos de *hardware* e controlo em todos os níveis funcionais e organizacionais, a fim de se ajustar rapidamente a capacidade de produção e funcionalidade em resposta a mudanças bruscas de mercado ou em requisitos regulamentares.

2.1.2 I-RAMP3

Este projeto, que se entende por Intelligent Reconfigurable Machines for Smart Plug&Produce Production, explora a capacidade de virtualização dos equipamentos presentes na indústria, tipicamente, PLC's, PC's, Máquinas Complexas, sensores e atuadores. No âmbito do projeto, estes equipamentos necessitam de lógica adicional para que as tarefas delegadas aos equipamentos possam ser desempenhadas com um grau adicional de inteligência. Estes dispositivos serão então virtualmente encapsulados e representados por NETWORK enabled DEVICES, entidades virtuais que exploram a capacidade de comunicação transparente conseguida pelo protocolo *UPnP* e as capacidades dos dispositivos encapsulados, sejam capacidades de monitorização ou de atuação.

Tem como propósito, colmatar os requisitos de produção que são impostos à indústria Europeia nos dias de hoje de forma inteligente. Visa facilitar a integração de dispositivos em sistemas já implementados, explorando o conceito de *plug&produce*, minimizar os custos e o tempo de arranque, operação e de manutenção das linhas de produção.

2.1.3 SelSus

O projeto, com o nome de Health Monitoring and Life-Long Capability Management for SELF- SUSTaining Manufacturing Systems, tem como ambição criar um novo paradigma para obter máxima eficiência, autorreparação e sustentabilidade do sistema e dos recursos de produção, com o objetivo de maximizar o seu desempenho ao longo de todo o tempo de vida do sistema. A ideia é ter uma nova geração de máquinas, equipamentos e ferramentas industriais que em conjunto com dispositivos integrados com capacidade sensorial e *software* avançado, permitam criar um sistema com capacidade de autodiagnóstico e autorreparação que sirvam de suporte a todo o sistema de produção.

Este é um conceito diferente sobre a manufatura que incide numa autoanálise de toda a maquinaria. Criar um sistema de diagnóstico onde se pretende que as máquinas tenham uma forte componente de intervalidação, ou seja, que possam verificar de forma automática se tudo está a decorrer como pretendido, permitindo a detenção de avarias

e no caso de ser possível, permitir o ajuste automático de certos parâmetros com vista a melhorar ou prevenir que a máquina fique indisponível. O conceito de autoconhecimento e aprendizagem, resultante de dados processados por redes bayesianas, permite a marcação automática de revisões às máquinas no *shop-floor*, a previsão de possíveis falhas ou problemas e a análise de eficiência individual de cada máquina ou a eficiência total do sistema.

Tudo isto permite então obter um sistema inteligente e autossuficiente, economizador de recursos e que tira a eficiência máxima dos equipamentos fazendo com que eles durem e produzam o máximo. As informações resultantes do serviço proposto nesta dissertação, podem ser úteis para as ferramentas de diagnóstico utilizadas neste projeto.

2.2 Requisitos

A produção industrial é uma área onde a eficiência é uma estratégia chave, otimização dos recursos utilizados, aceleração das fases de desenho dos produtos finais, controlo de qualidade e problemas na linha de montagem. Para que um serviço de produção seja eficiente, existem várias garantias que têm que ser dadas, tais como:

- Informação detalhada sobre todos os dispositivos presentes na rede.
- Garantir conectividade e comunicação entre dispositivos.
- Garantir que a rede de sensores implementada alcança uma cobertura máxima.
- Dispositivos na rede com fiabilidade adequada ao serviço pretendido.
- Sugestões dinâmicas de planeamento, orientação e configuração da rede.

O presente trabalho irá incidir sobre estas mesmas garantias, requisitos estes que requerem uma integração com os sistemas de gestão de serviços das RSSF permitindo uma abordagem mais eficiente e rápida ao seu processo. Ao explorar os serviços anunciados na rede de sensores, podem-se implementar estratégias que resultem na solução de problemas concretos no cenário industrial.

O recurso às ontologias utilizadas por estes sistemas, nomeadamente troca de descrições de serviços e dispositivos em ficheiros de formato XML, permitirá obter os dados precisos sobre requisitos de serviços e características específicas de cada dispositivo, que servirão de fonte a estratégias de planeamento e gestão da rede. Este processo resulta num outro problema, visto que a integração com os sistemas deverá garantir avaliações dinâmicas,

fiáveis e em tempo real dos serviços e nós da rede de sensores. A compatibilidade entre atores tem de ser levada em consideração, visando facilitar a comunicação e interpretação de dados por todos os dispositivos associados ao sistema. É necessário o suporte de tecnologias eficazes e com capacidade de fornecer mecanismos de comunicação entre diferentes tipos de hardware e serviços de forma a criar um sistema robusto e integrável em diferentes plataformas.

O principal problema, que resulta no principal foco deste projeto, é garantir que a rede consegue monitorizar toda a área ou região, ou seja, garantir que a implementação da rede de sensores tem total cobertura sobre a área da sua instalação e que os dispositivos conseguem fornecer a fiabilidade e precisão adequada ao tipo de serviço e aplicação pretendida assim como fornecer um serviço integrado na arquitetura da rede de modo a estar disponível a qualquer entidade que dele pretenda usufruir.

Capítulo 3

Estado da Arte

3.1 Redes de sensores sem fio (RSSF)

O avanço tecnológico que se verifica nas áreas que estão intrinsecamente ligadas às redes de sensores sem fios, levaram a uma crescente adaptação e implementação de sistemas e aplicações com este tipo de rede. A evolução natural de microprocessadores, dispositivos sensores e/ou atuadores, MEMS (Micro Electro-Mecanical Systems) e da comunicação sem fios, garante que as redes de sensores sejam uma solução bastante prática e completa para as mais variadas aplicações em que pretendemos determinar de forma automática informação sobre determinado fenómeno físico.

Ao longo desta dissertação será adotado o termo de redes de sensores sem fios (RSSF) para designar a interligação de um determinado número de dispositivos com capacidade de observar o meio físico, podendo ou não conter dispositivos com capacidade de interagir com o ambiente, nomeadamente atuadores. Segundo os autores de [5], uma rede de sensores/atuadores pode ser definida como uma rede não linear de conexões pesadas entre um pequeno ou grande número de sensores/atuadores. Note-se também que a palavra sensor, ao longo deste documento, representará em concreto dispositivos de capacidade restrita que têm a capacidade de recolher dados do meio em que se encontram, processá-los ou não e disseminá-los através de uma rede. Dispositivos estes, que podem ser também denominados de nós de uma rede de sensores sem fios.

As RSSF fornecem então a possibilidade de determinar o valor de um parâmetro num local específico ou detetar a ocorrência de eventos com interesse, estabelecer valores de parâmetros de acordo com o tipo de evento ocorrido, classificar objetos detetados e rastrear um determinado objeto [6].

Os requisitos de *software* e hardware de uma RSSF diferem mediante o tipo de sistema que se deseja implementar, o que torna o processo de configuração e planeamento, assim como o método de funcionamento e a manutenção deste tipo de redes dependente do propósito da implementação, do local da implementação e dos requisitos exigidos aos serviços da própria rede.

3.1.1 Áreas e setores de aplicação

A enorme capacidade que uma rede de sensores pode conferir em qualquer área e setor torna-se cada vez mais atrativo para diversas entidades, sendo já um dos temas mais investigados pela comunidade educacional e investigadora. As RSSF são vistas como a peça que faltava no puzzle das redes informatizadas não só por fornecer a capacidade de monitorizar o meio, bem como permitem a interação com o mundo real, além de que, este tipo de redes pode ser bastante modular, adaptando-se ao tipo de aplicação e objetivo final.

Os seus dispositivos, apesar de apresentarem capacidades bastante reduzidas ao nível do processamento, bateria e memória, são bastante configuráveis. Um só dispositivo pode conter vários sensores que podem ser adaptados e escolhidos de acordo com o tipo de aplicação que se deseja, por exemplo para um sistema de controlo de gestão da iluminação de um edifício são usados sensores de luminosidade enquanto sensores de imagem e acústicos devem ser implementados nos dispositivos de um sistema de segurança. A construção de uma rede de sensores cablada, seja com par trançado, cabo coaxial ou de fibra ótica, pode ser a melhor resposta para sistemas críticos quando a rapidez de processo e troca de informação é essencial. Contudo, esta abordagem facilmente se torna inviável, não por questões de falta garantias de requisitos, mas porque pode fazer com que o custo de instalação e manutenção da rede suba para valores inoportáveis, como uma rede instalada numa floresta ou numa costa marítima.

As RSSF têm-se demonstrado capazes de responder a problemas em diversas áreas. Na área de controlo onde podem ser utilizadas para efetuar testes numa linha de montagem, na área do ambiente com a monitorização de florestas, costas marítimas, vulcões ou simplesmente usadas para monitorizar e gerir edifícios públicos ou privados. A área da segurança faz igualmente uso destas redes, como os sistemas de vídeo vigilância em centros comerciais ou parques de estacionamento que também pode ser aplicada ao tráfego na monitorização de veículos e gestão de trânsito rodoviário. Duas áreas que também têm despertado grande interesse em servir-se desta tecnologia são a medicina/biologia nas quais pode ser útil a monitorização de órgãos ou na deteção de substâncias perigosas, e a

área militar que aplica as RSSF em várias vertentes como a monitorização de militares no terreno, deteção de minas ou materiais perigosos em determinadas regiões ou na gestão de armamento e veículos.

As várias entidades de diferentes setores estão atentas às capacidades que as RSSF possuem e começam a surgir cada vez mais aplicações e sistemas que visam responder às necessidades de cada setor, facilitando e/ou melhorando serviços e permitir a automatização, monitorização, gestão e controle de vários parâmetros. Pode identificar-se alguns setores onde se encontram já implementados algumas soluções deste tipo como produção industrial, setor agrícola, distribuição de energia, gás e água, extração de petróleo e gás, indústria de aviação [6]. Verifica-se ainda um vasto número de situações onde as redes de sensores sem fios podem ser empregues, visto apresentarem soluções capazes para um conjunto extraordinariamente diversificado de problemas; monitorização em fábricas, edifícios habitacionais, monumentos, florestas, desertos, oceano, áreas de difícil acesso como vulcões através de parâmetros como pressão, temperatura, luminosidade, nível, movimento.

Razões mais do que suficientes para despertarem o interesse no setor da indústria, que visa elevar não só o nível de produção, bem como a organização, gestão, garantia e fiabilidade, diminuindo tempo e custos.

3.1.2 Características das RSSF

A avaliação de uma rede de sensores está subjacente à aplicação a que se destina, por conseguinte as suas características estão dependentes do objetivo e serviços requeridos. O planeamento da rede, a sua arquitetura, as funções e capacidades físicas e lógicas de cada dispositivo, a disposição de cada nó da rede pela área a monitorizar e a pilha protocolar a utilizar, deverão ter em conta o local de implementação da RSSF, os seus componentes e o objetivo final da aplicação pretendida. De acordo com os autores de [1] uma rede de sensores pode classificar-se segundo quatro parâmetros; configuração, capacidade de monitorização, tipo de comunicação utilizado e tipo de processamento efetuado.

3.1.2.1 Configuração

Os nós que representam cada componente da rede não têm de ser necessariamente iguais nas suas capacidades físicas e lógicas. Uma RSSF define-se como homogénea quando constituída por nós com os mesmos requisitos em termos de *hardware* e heterogénea caso contrário. O nível organizacional dos dispositivos presentes na rede pode ser efetuado de

forma hierárquica, onde os nós se agrupam conforme determinados parâmetros, ou de forma plana caso não exista nenhum grupo de dispositivos formado. Por sua vez, os grupos podem ter hierarquias entre si e conter um nó responsável por ser o líder, conhecido como *cluster-head*, desse conjunto de dispositivos.

Devido a existência de dispositivos com capacidades móveis, podemos separar este tipo de redes em dois tipos: a móvel e a estacionária. Numa RSSF móvel, ao longo do tempo de vida da rede, os dispositivos podem alterar as suas posições iniciais para cumprir certos requisitos estipulados sendo que, numa rede estacionária nenhum dos dispositivos altera a sua posição física desde a sua instalação.

A distribuição dos nós pela área de aplicação pode ser ou não uniforme e a densidade da rede é atribuída mediante o número de nós distribuídos por unidade de área. Em [1], o autor afirma que a rede é equilibrada quando existe uma distribuição ideal segundo o objetivo da rede, densa ou esparsa quando a concentração de nós por unidade de área é alta ou baixa respetivamente.

3.1.2.2 Monitorização

Os dispositivos sensores podem ser configurados de maneira diferente quanto à recolha de dados do meio ou fenómeno. A obtenção de dados pode ser efetuada de forma periódica ou contínua, em tempo real ou por reação a um impulso. Mais uma vez a escolha de cada um destes métodos recai sobre os requisitos da própria aplicação. Uma aplicação com o objetivo de recolher informações de temperatura numa floresta poderá dividir temporalmente o período de ativação e recolha de dados de cada dispositivo de maneira a economizar energia e consequentemente prolongar o tempo de vida da rede. Num sistema de vigilância a captação de dados é realizada de forma contínua ou configurada para reagir apenas se algum movimento for detetado. A comunicação em tempo real geralmente é escolhida em áreas com sistemas críticos, que envolvem riscos ou importantes tomadas de decisão como é o caso da medicina e área militar.

3.1.2.3 Comunicação

Neste tópico é abordada a forma como a informação é distribuída numa RSSF. Para uma mais fácil perceção apresentamos a tabela 3.1 que classifica este tipo de redes mediante a comunicação e ligações efetuadas.

Tabela 3.1: Fluxo de informação. Comunicação em redes de sensores sem fios. [1]

Envio de dados	Programada	Nós da rede enviam os dados em intervalos regulares
	Contínua	Nós da rede enviam os dados de forma contínua.
	Reativo	Nós apenas enviam dados mediante consultas ou por reação a um qualquer evento.
Tipo de Conexão	Simétrica	Todas as ligações entre os nós da rede têm o mesmo alcance, eventualmente poderá ser diferente no nó central.
	Assimétrica	Todas as ligações entre os nós da rede têm um alcance diferente.
Transmissão	Simplex	Nós apenas possuem a capacidade de transmitir dados
	Half-Duplex	Nós possuem a capacidade de transmitir e receber dados, contudo apenas o consegue fazer em exclusividade, ou seja, se transmite não recebe e vice-versa.
	Full-Duplex	Os nós têm capacidades para transmitir e receber dados em simultâneo.
Alocação de Canal	Estática	A largura de banda pode ser dividida por frequência, por tempo, por código ou ortogonalmente. (FDMA, TDMA, CDMA, OFDM)
	Dinâmica	Não existe nenhum critério de atribuição de banda e por isso os nós disputam pela utilização do canal.
Fluxo de informação	Flooding	Nós comunicam com os seus vizinhos através de <i>broadcast</i> até que os dados cheguem ao destino. Existe muitas mensagens a circular na rede o que pode provocar problemas na fiabilidade da mesma.
	<i>Multicast</i>	A rede está organizada hierarquicamente e é utilizado multicast para a comunicação entre os nós pertencentes a cada grupo.
	<i>Unicast</i>	Nós tomam partido de protocolos de roteamento multi-salto para comunicar diretamente com o nó pretendido.
	<i>Gossiping</i>	Os nós escolhem os destinatários a que querem enviar informação
	<i>Bargaining</i>	Apenas é transmitida informação por parte de um nó se receber uma manifestação por parte de outro nó. Em geral, os nós solicitam a transmissão de dados.

3.1.2.4 Processamento

O processamento dos dados numa RSSF está também dependente do tipo de aplicação pretendida, e ocorre de várias formas conforme os requisitos pretendidos pelo sistema. Os nós, que representam cada dispositivo presente na rede, realizam procedimentos relativos à infraestrutura como a eleição de líderes em redes hierárquicas, procura e descoberta de nós ou criptografia de dados. O processamento pode ainda ser realizado de forma localizada, quando os nós executam um pequeno processamento local antes de enviar informações através da rede, como por exemplo a tradução dos dados que recolhem, ou de forma correlacionada onde os dispositivos utilizam técnicas de fusão e agregação de dados, contagem e compressão à medida que são trocadas as informações pela rede.

3.1.2.5 Energia

Uma das características das RSSF mais importante para garantia de QoS do sistema é a sua energia. O facto de estas redes permitirem a sua instalação em locais remotos e por vezes de difícil acesso, dificulta a sua manutenção física, e torna importantíssima a questão da energia. O tempo de vida de uma rede é definido pela quantidade de energia de cada um dos seus dispositivos que são alimentados geralmente por baterias.

Torna-se então importante, gerir da melhor forma a quantidade de energia gasta por cada componente de uma RSSF, assim como garantir que a rede seja capaz de responder a falhas críticas de energia que podem ter consequência direta na falha de transmissão e receção de dados. Como exemplo prático, um dispositivo sensor pode ficar desativado por longos períodos de tempo e apenas transmitir quando lhe é efetuado um pedido por parte de outra entidade, poupando desta forma energia. Se os dispositivos forem capazes de trocar através da rede, informação sobre a quantidade de energia que possuem e desta forma fornecer um meio utilizado por certos protocolos de roteamento na gestão e construção de caminhos fiáveis entre os nós da RSSF. Com esta informação torna-se possível elaborar um mapa de energia da rede, ou seja, é possível representar graficamente a informação em tempo real de toda a energia presente na rede e desta forma prevenir falhas do sistema ou adaptar a rede da melhor maneira em função de maximizar o seu tempo de vida.

Os sensores representam os elementos do dispositivo que são responsáveis por escutar e sentir o meio ambiente e o seu consumo de energia está dependente da forma como se encontram a operar e do tipo de grandeza que estão a recolher.

3.1.3 Componentes, padrões e tecnologias

3.1.3.1 Dispositivos

As motes são os principais responsáveis por captar e recolher informações do mundo real, sendo a unidade central deste tipo de redes contudo, podem ter também a capacidade de processamento e envio de dados para outros dispositivos [7]. Normalmente estes dispositivos são de dimensões reduzidas e apresentam sérias limitações no que concerne ao processamento, memória e bateria. Apesar de apresentarem a vantagem de terem um custo e tamanho reduzido, facilitando assim sua implementação e distribuição pelas mais diversas regiões, é essencial que se estude os melhores métodos para lidar com as suas restrições seja pelo fraco processamento ou pela quantidade de bateria que possuem. O facto desta proposta se enquadrar nos projetos acima referidos, I-RAMP3 e Selsus, faz com que sejam considerados os dispositivos que deles fazem parte, fala-se de sensores de toque, humidade, pressão, temperatura, indutivos, acústicos e óticos.

Tipicamente a arquitetura da mote divide-se em cinco unidades: a unidade de microcontrolador, a unidade memória, a unidade de Sensor, a unidade de comunicação e a unidade de Bateria [7].

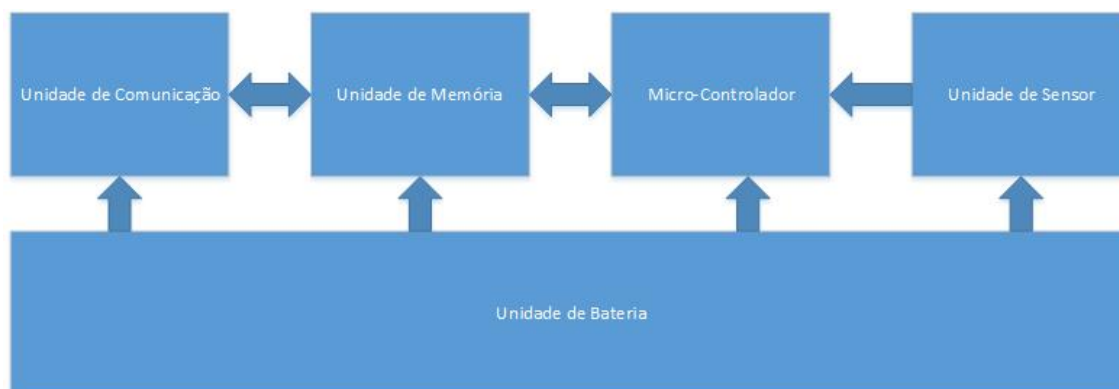


Figura 3.1: Componentes do dispositivo

O microcontrolador é o núcleo do dispositivo e está diretamente ligada à sua capacidade de decisão. É projetado para consumir a menor energia possível devido às limitações que estes dispositivos apresentam. A funcionalidade desta unidade baseia-se no processamento dos dados recolhidos pelo sensor, decidindo e controlando para onde e quando são enviados esses dados, gerindo inclusive toda a informação que poderá receber de outros nós. O microcontrolador tem a capacidade de gerir a sua energia quando a sua utilização

é nula ou escassa, controlando o seu período de atividade durante o tempo de vida do sistema. A unidade de memória pode ser dividida em duas partes distintas.

A RAM (*random access memory*), que dá suporte à unidade de microcontrolador e a ROM (*read-only memory*), usada para guardar os programas do dispositivo e os dados recolhidos pela unidade de sensor. Note-se que a velocidade de acesso e escrita na RAM é superior à da ROM, contudo, se existir alguma falha de energia no dispositivo, este pode perder todos os dados recolhidos e/ou analisados. Ler e escrever dados na memória tem como consequência um gasto de tempo e de memória que está dependente dos requisitos de cada aplicação.

A unidade de sensor contém todos os componentes que interagem e captam informação do mundo físico segundo vários parâmetros, sejam eles temperatura, humidade, luminosidade ou pressão. Possui uma unidade de conversão (ADC – *Analogue-to-Digital Converter*) com a função de converter os sinais analógicos, que são produzidos pelos sensores, para sinais digitais que são utilizados e processados pela unidade de microcontrolador. Os sensores podem ser desativados e ativados mediante a necessidade de uso ou simplesmente serem definidos para estar num estado de alerta diminuindo a sua atividade até ser estimulado por algum evento externo.

A unidade de Comunicação, como o próprio nome indica, contém todos os componentes necessários para que os dispositivos de uma rede comuniquem entre si. Numa rede cablada existe a necessidade de cada dispositivo estar conectado por um meio físico o que limita a flexibilidade e escalabilidade de uma rede de sensores e pode elevar o custo de implementação para valores insuportáveis.

Geralmente em algumas RSSF os dispositivos são equipados com um transceptor, que inclui um transmissor, recetor, antena, amplificadores de potência e outro tipo de componentes de modo a garantir a comunicação com os outros dispositivos integrados na rede. Normalmente o transceptor pode funcionar mediante quatro estados, é controlado pelo microcontrolador e pode adaptar a sua utilização de modo a controlar os gastos de energia e como consequência, maximizar o tempo de vida da rede. Dois estados óbvios desta unidade são a transmissão e receção. A transmissão acontece quando o dispositivo apenas está a transmitir dados pela rede, neste caso o transmissor encontra-se então ligado e a antena irradia energia, a receção ocorre quando o dispositivo está a receber dados e simultaneamente o transceptor apenas coloca ativo o recetor, note-se que a quantidade de energia despendida é maior no estado de transmissão do que receção de informação. De modo a lidar com a questão da gestão de energia, o transceptor pode

operar em dois modos, parado (*idle*) e dormir (*sleep*). No primeiro caso, o dispositivo não está a receber informação mas encontra-se à escuta e pronto a reagir se algum evento lhe é associado. O segundo caso é o que tem menor coeficiente de gasto energético pois todos os componentes do transceptor encontram-se desligados.

A unidade de bateria de um dispositivo sensor, como já foi mencionado ao longo deste trabalho, é um dos pontos críticos no que diz respeito a redes de sensores sem fios. A própria rede é projetada e definida em função da maximização do seu tempo de vida e por sinal da conservação de energia. Geralmente, esta unidade é constituída por baterias não recarregáveis e com energia finita, não obstante que poderá ser incorporado no dispositivo um sistema de recarga, como por exemplo painéis de captação de luz solar.

3.1.3.2 Padrões

Vários protocolos foram propostos para redes sem fios para automação industrial tais como: Bluetooth, Wi-Fi, ZigBee, WirelessHART e ISA SP100.11a.

O Bluetooth atua na banda industrial, científica e médica (ISM) entre 2,4 a 2,458 GHz, usando *spread spectrum* através de um sinal *full-duplex*. A modulação utilizada é OFDM e utiliza *adaptive frequency hopping* (AFH) de modo a lidar com certas interferências presentes no meio, tendo liberdade de escolher frequências diferentes das utilizadas por outros dispositivos. Dependendo da potência, os dispositivos são classificados em três classes, com alcance de um metro para a classe 3, dez e cem metros para as classes 2 e 3 respetivamente. A taxa de transmissão varia consoante a versão utilizada, situando-se entre 1 Mbp/s para a versão 1.2, 3 Mbp/s e 24 Mbps na versão 2.0 e 3.0 respetivamente.

A tecnologia Wi-Fi, baseada no padrão IEEE 802.11 atua na faixa de frequência de 2,4 GHz ou 5 GHz e consegue um alcance de trinta a trezentos metros. Para lidar com problemas de interferências e ruídos presentes no meio, faz recurso de técnicas de transmissão como DSSS e OFDM, bem como CSMA/CA para deteção de colisões.

O ZigBee tem como mais-valia o seu baixo consumo de energia, o que eleva autonomia da bateria de cada dispositivo e conseqüentemente o tempo de vida da rede. Segue as especificações da norma IEEE 802.15.4 e pode operar em diferentes faixas de transmissão, 2,4 GHz, 915 MHz e 868 MHz. Possui métodos criptográficos (AES-128) para segurança de dados e as taxas de transferência variam entre 20 kbps a 250 kbps. Para lidar com as interferências e o ruído que possa existir no meio, o padrão disponibiliza várias

técnicas de transmissão DSSS.

O primeiro padrão de comunicação sem fios a ser desenvolvido para a indústria de processo foi o WirelessHART, sendo compatível com o padrão IEEE 802.15.4. Tal como o ZigBee, este padrão é caracterizado pelo baixo consumo de energia, operando na faixa de transmissão 2,4 GHz com uma taxa de 250 kbps. Uma particularidade da sua arquitetura é o fato de cada dispositivo funcionar como um *router* ou repetidor de sinal, sendo a comunicação entre cada nó realizada sobre TDMA (*Time Division Multiple Access*).

A *International Society of Automation* (ISA) desenvolveu um padrão de comunicação para redes sem fios industriais, denominado ISA SP100.11. As suas funções e características são semelhantes ao protocolo WirelessHART, com a particularidade de fornecer a capacidade de integração com os protocolos existentes, como por exemplo, HART, Profibus, DeviceNet, Fieldbus, entre outros.

3.1.3.3 Sistemas Operativos

As RSSF são definidas geralmente como um sistema reativo e por vezes associadas a aplicações e sistemas críticos. Estas redes podem conter um número elevado de dispositivos sensores com sérias limitações de memória, processamento e energia, conectados entre si com o objetivo de detetar e transmitir fenómenos do meio físico. Os sistemas operativos devem ter em consideração todas as restrições físicas e lógicas dos dispositivos sensores que um sistema operacional comum seria incapaz de suportar. Surge então a necessidade de estruturar sistemas operativos capazes de gerir os recursos de *hardware* de forma eficaz, que suportem operações simultâneas e intensivas, consiga modularidade e robustez e uma boa gestão de energia [8].

Alguns sistemas operativos vêm sendo propostos para as RSSF como o MANTIS OS, o YATOS, o Contiki, entre outros. São sistemas que se apresentam leves, compactos, modulares e preparados para lidar com vários tipos de restrições presentes em RSSF quer sejam de carácter físico como lógico. Um dos sistemas mais famosos quer pela comunidade estudantil, investigadora e empresarial é sem dúvida o TinyOS [9].

O sistema operacional TinyOS é um sistema *open-source* especificamente desenhado para dispositivos wireless de baixa computação, computação ubíqua, PANs e edifícios inteligentes. É um sistema idealizado e construído de forma a lidar com as restrições dos dispositivos tipicamente presentes nas RSSF, que geralmente apenas possuem micro-controladores com poucos kB de RAM e pouco espaço para programas. Começou a ser

desenvolvido na universidade de Berkeley no ano de dois mil por Jason Hill na sua tese de mestrado, onde apresenta um sistema baseado em eventos, simples, pequeno, e com um conjunto de ferramentas disponíveis que suportam o desenvolvimento de código específico para cada aplicação. A nova versão do sistema suporta um vasto conjunto de dispositivos com diferente processador, radio e memória, nomeadamente a família telos, micaZ, IRIS, mica2, epic, mulle, tinynode, span e iMote2.

3.2 Cobertura e conectividade de uma RSSF

A cobertura de uma área e a conectividade dos nós podem ser classificadas como medidas de qualidade de serviços numa RSSF. Pretendem responder a questões simples e essenciais numa rede deste tipo, como o quão bem uma certa área de interesse está coberta pelos dispositivos e o quão precisos podem ser estes na captura de informação. Depois de distribuídos, os nós sensores formam uma rede que pode mudar dinamicamente ao longo do tempo conforme a mobilidade dos seus nós, a energia global da rede, os obstáculos presentes no meio ou o ruído a que está exposta, entre outros [10].

Basicamente o problema da cobertura especifica o número de nós mínimos que se tem de implantar numa determinada área para que esta fique otimamente coberta, ou seja, todos os pontos de interesse presentes no ambiente a monitorizar estão dentro da área de monitorização de pelo menos um sensor.

O melhor ambiente para a instalação de uma RSSF seria um, onde não existisse interferências nas comunicações, o terreno fosse plano e não existisse vegetação. Em tal cenário, os dispositivos sensores que constituem a rede estariam dispostos em forma de grelha, distanciados mediante o alcance do seu rádio, ou seja, a distância que separa cada elemento da rede é proporcional ao alcance do rádio de cada dispositivo [11].

Em [10] encontramos definidos três tipos de cobertura caracterizadas pela forma como estão dispostos os nós presentes na rede. Na primeira os nós são estáticos e dispostos de forma a maximizar o rácio de detenção dos pontos a analisar, foca-se essencialmente em garantir que toda a área seja coberta por pelo menos um sensor. A cobertura em forma de barreira caracteriza-se pela disposição de dispositivos em linha, onde a disposição estática de nós pretende minimizar a probabilidade de algo intruso trespassar a barreira criada pelos sensores. Como exemplo deste tipo de cobertura temos os alarmes de segurança ou uma rede de sensores a monitorar o local de pasto do gado criando barreiras virtuais para controlar os animais [12]. Algumas estratégias consideram ainda o recurso a dispositivos

móveis presentes na área de monitorização de maneira a se atingir um equilíbrio, entre a maximização do rácio das detenções de eventos de cada nó e a minimização de dados perdidos por unidade de área. Por exemplo, uma rede de sensores implementada numa enorme área florestal, por onde passam periodicamente determinados dispositivos móveis com a função de recolher as informações dos sensores.

Todos os modelos de cobertura e conectividade estão dependentes do objetivo da aplicação, da própria rede e dos algoritmos e protocolos utilizados. Os mesmos autores apresentam três tipos diferentes de estratégias para se determinar uma cobertura ótima. Estratégias que fazem recurso de ferramentas de geometria computacional tais como o diagrama de Voronoi e a triangulação de Delaunay, sendo estas estratégias direcionadas a aplicações que tentam detetar algo não autorizado e presente na rede. Estratégias que utilizam algoritmos que exploram a mobilidade dos nós com o objetivo de obter uma melhor cobertura, estes algoritmos assumem movimentações efetuadas pelos dispositivos depois de estarem nas suas posições iniciais para que se atinja a máxima cobertura de rede de forma a maximizar a cobertura da rede. Definem ainda, o recurso a algoritmos que lidam com os problemas de cobertura e de conectividade através de um *framework* integrado. Estas estratégias, normalmente focam-se mais a nível protocolar, tratam do controlo de topologia e escalonamento da atividade de cada nó, isto é, cada nó tem um modo que lhe permite ligar-se e desligar-se automaticamente segundo determinados critérios, com o intuito de minimizar o consumo de energia ou então tratam do desdobramento da estrutura da rede.

A cobertura de uma rede é então definida através da distância entre um determinado ponto no espaço de monitorização e a localização do dispositivo sensor, que fazem recurso de funções que podem ser definidas de diferentes formas dependendo da interpretação, do tipo de sensores e do tipo de aplicação. Tem como objetivo responder à questão do quão bem um sensor consegue captar um fenómeno físico numa determinada localização [7].

3.2.1 Modelos de Cobertura

Os modelos de cobertura consideram medidas de qualidade e capacidade, no que concerne a cada sensor individualmente. Através das coordenadas cartesianas do sensor e de um ponto a monitorizar, podemos calcular a distância que os separa através da função (3.1)[7]:

$$d(s, p) = \sqrt{(s_x, p_x)^2 + (s_y, p_y)^2} \quad (3.1)$$

Considerando o plano bidimensional, sendo p um ponto no espaço a monitorizar e S um conjunto de dispositivos sensores $\{s_1, s_2, \dots, s_n\}$ e $(s_x, s_y)(p_x, p_y)$ as coordenadas cartesianas do sensor e do ponto respetivamente, a função $d(s,p)$ define a distância euclidiana entre o sensor s e o ponto p , concluímos então que o vetor $d_n = [d(s_1, p), d(s_2, p), \dots, d(s_n, p)]$ representa o conjunto das distâncias entre n sensores e um determinado ponto p .

Em [7] encontramos alguns modelos definidos que têm em conta as características físicas dos dispositivos. O modelo de cobertura de setor é um modelo booleano utilizado quando o sensor tem uma determinada orientação e campo “visual”, como por exemplo uma câmara direcionada. A função deste modelo pode tomar os valores de um quando os pontos estejam situados dentro da área monitorizada pelo sensor ou zero caso contrário. Assumindo θ_s como o ângulo que representa a orientação do sensor, w seu ângulo visual e R_s o alcance seu raio de monitorização, a função de cobertura deste modelo é dada por:

$$f(d(s, p), \theta(s, p)) = \begin{cases} 1, & d(s, p) \leq R_s \text{ e } \theta_s \leq \theta(s, p) \leq \theta_s + w \\ 0, & \text{caso contrário} \end{cases} \quad (3.2)$$

Podemos então assumir, que a cobertura média do ponto monitorizado por vários sensores é a soma do valor da cobertura de cada sensor que o está a cobrir, ou seja:

$$f(d_n, \Theta_n) = \sum_n^{i=1} f_i(d(s_i, p), \Theta(s_i, p)) \quad (3.3)$$

Esta função tem como resultado o número de sensores que estão a monitorizar o ponto p .

Outro modelo booleano muito utilizado na literatura e talvez o mais famoso é o modelo de cobertura em disco. Este modelo caracteriza-se por definir que um sensor está apto a captar todos os pontos de interesse que estão contidos numa circunferência que tem como centro o próprio dispositivo e raio o alcance do seu sensor. Contudo, ele não leva em consideração que o processo de monitorizar determinada área pode ser influenciado, não só pelas características físicas do dispositivo como por exemplo pela má calibração do *hardware* ou por fatores como condições ambientais, obstáculos presentes no meio ou ruído existente.

Supõe-se então que toda a área abrangida por esta circunferência está a ser monitorizada pelo dispositivo e por isso, se a distância do dispositivo a um ponto qualquer na área for menor que R_s então a função (3.4) retorna um, caso contrário retorna zero.

$$f(d(s,p)) = \begin{cases} 1, & d(s,p) \leq R_s \\ 0, & \text{caso contrário} \end{cases} \quad (3.4)$$

Desta forma é possível determinar o número de sensores que estão a cobrir um determinado ponto no espaço pela soma das funções de cobertura de cada nó individual(3.5), ou seja, se $f(d_n) = k$, o ponto é coberto por k sensores.

$$f(d_n) = \sum_{i=1}^n f_i(d(s_i,p)) \quad (3.5)$$

Tipicamente, a qualidade dos dados recolhidos pelo dispositivo diminui com o aumento da distância entre este e o ponto a monitorizar no espaço. De forma a obtermos valores mais precisos, existem modelos que lidam com o problema da redução da qualidade do valor recolhido pelo sensor. A medida de cobertura de um dispositivo em relação a um ponto p presente na área pode ser definida através da função (3.6) onde 'a' representa o expoente de atenuação do caminho e C uma constante definida.

$$f(d(s,z)) = \frac{C}{d^a(s,z)} \quad (3.6)$$

Mediante esta situação podemos afirmar que dados dois pontos no espaço, o ponto situado mais próximo do sensor tem melhor média de cobertura e terá uma monitorização mais precisa. Desta forma, determinamos o conjunto de sensores que está a monitorizar um ponto de interesse na área através da soma dos valores de medida de cobertura de cada sensor individualmente, ou seja:

$$f(d_n) = \sum_{i=1}^n \frac{C}{d^a(s_i,p)} \quad (3.7)$$

Para lidar com o problema da perda de qualidade dos dados recolhidos quando aumenta a distância entre o dispositivo e o ponto a observar podem ser realizadas algumas aproximações truncando a média de cobertura para elevados valores de distância. Os autores de [13] idealizam uma função (3.8) que tem em consideração certas características físicas dos sensores (α), como a sua precisão ou calibração de hardware, de modo a estimar a variação do seu alcance.

$$f(d(s,p)) = \begin{cases} Ce^{-(\alpha d(s,p))}, & d(s,p) \leq R_s \\ 0, & \text{caso contrário} \end{cases} \quad (3.8)$$

Os mesmos autores em [14] definem uma nova função(3.9) que além de considerar o raio de alcance do sensor (R_s), têm em consideração um valor (R_u) que representa a distância onde um ponto de interesse tem uma baixa taxa de probabilidade de ser detetado mas que ainda não se encontra fora de alcance do dispositivo.

$$f(d(s, p)) = \left\{ \begin{array}{ll} 1, & d(s, p) \leq R_s - R_u \\ C e^{-\alpha(d(s, p) - (R_s - R_u)^\beta)}, & R_s - R_u < d(s, p) \leq R_s \\ 0, & \text{caso contrário} \end{array} \right\} \quad (3.9)$$

Neste projeto vamos representar cada dispositivo sensor como uma circunferência que tem o dispositivo no seu centro e o seu raio é determinado pelo raio de alcance do sensor. Este é talvez o método mais comum presente na literatura para definir um sensor e daí resulta a nossa escolha.

3.2.2 Modelos de Conectividade

Os modelos de conectividade centram-se na temática de comunicação entre nós numa rede de sensores sem fios. Assume-se que um dispositivo está apto a comunicar apenas até uma certa distância limite, representada como o raio de comunicação (R_c). Quando os nós possuem diferentes raios de comunicação de acordo com o seu poder de transmissão, dois nós S_i e S_j comunicam, se a distância euclidiana entre eles é menor ou igual ao mínimo dos seus raios (3.10), ou seja, o nó com menor R_c é abrangido pelo raio de comunicação do outro nó. Quando dois dispositivos se encontram nesta situação, estão aptos a comunicar entre si e são geralmente chamados de vizinhos.

$$d(s_i, s_j) \leq \min\{R_{ci}, R_{cj}\} \quad (3.10)$$

Uma rede de sensores, pode então ser caracterizada como um grafo onde cada nó representa um dispositivo sensor e a distância que separa os nós vizinhos representam os arcos que os ligam.

O poder do sinal recebido P_r , por parte de um nó é dado por (3.11), e tem em conta o poder do sinal transmitido P_s e um valor sobre a distância entre os pontos η que representa o expoente de perda de percurso situado entre 2 e 6 dependendo das condições ambientais[7].

$$P_r = \frac{P_s}{d(s_i, s_j)^\eta} \quad (3.11)$$

Ao longo do tempo de vida de uma rede de sensores sem fios pode acontecer que um ou vários nós recebam sinais que não eram supostos ou previstos, o que pode resultar em algumas interferências. A comunicação entre dispositivos é também afetada pelas condições do meio onde a rede se encontra seja por ruído, vegetação, edifícios, máquinas ou qualquer outro tipo de obstáculos que possa de alguma forma interferir com a comunicação entre dispositivos. Na hora de planear a instalação de uma rede de sensores sem fios, torna-se importante analisar o espaço físico onde será implementada e ter em conta este tipo de situações para garantir um funcionamento correto da rede assim como valores mais precisos.

3.2.3 Controlo de cobertura e conectividade

O controlo de cobertura e conectividade centraliza-se na questão da qualidade e capacidade que uma rede de sensores sem fio possui, com heterogeneidade de sensores e diferentes localizações geográficas, para garantir conectividade entre dispositivos e maximizar a cobertura de determinada área.

O primeiro desafio numa RSSF centra-se no facto de como cobrir perfeitamente uma região a monitorizar por isso, a conectividade e a cobertura são dois dos mais importantes problemas neste tipo de redes e têm grande impacto na sua performance. Uma boa estratégia de distribuição, um bom mecanismo de gestão de operação dos dispositivos e o raio de cobertura da rede, pode não só reduzir custos como aumentar o seu tempo de vida.

Segundo as definições teóricas a cobertura máxima define que todos os pontos localizados em determinada área a monitorizar estão cobertos por pelo menos um sensor e a conectividade requer que a rede seja conexa no que diz respeito ao raio de comunicação de cada dispositivo. Ou seja, a cobertura é afetada pela sensibilidade do sensor do dispositivo em detetar fenómenos do meio e a conectividade recai mais sobre a capacidade do alcance do raio de comunicação. Estas duas variáveis estão intrinsecamente relacionadas com o ambiente onde está implementada a rede, ou seja, a performance dos serviços é influenciada por ruído e interferências causadas por diversos fatores como por exemplo, obstáculos presentes na área a monitorizar.

O número de dispositivos que uma RSSF contém não garante, por si só, uma ótima cobertura e conectividade da rede. A quantidade de dispositivos presentes numa rede é variável conforme a aplicação e os requisitos esperados do sistema, não existindo um limite definido para a quantidade de nós que uma RSSF deve conter de forma garantir a

eficácia do serviço.

Em [15] a densidade de nós presentes numa determinada área é dada segundo $\frac{N}{A}$ onde N representa um número de nós aleatoriamente distribuídos por uma determinada área A . Quando existe um número relativamente elevado de nós sensores por área de interesse a rede é classificada como densa, sendo esta solução utilizada em situações em que a deteção de todos os eventos é de extrema importância ou simplesmente quando é relevante ter múltiplos nós sensores a cobrir determinada área, seja para melhorar a cobertura ou a conectividade ou para garantir determinado requisito protocolar. Pelo contrário, quando a rede contém poucos nós sensores por unidade de área, a rede classifica-se como esparsa. Este tipo de redes é útil quando o custo dos sensores torna inviável a construção de uma rede mais densa ou quando se deseja obter uma cobertura total da área com o mínimo de dispositivos necessários.

3.2.4 Mobilidade

Como já mencionado em capítulos anteriores, uma RSSF pode classificar-se como móvel devido à presença de dispositivos com capacidade motora que se deslocam ao longo do espaço de interesse de acordo com a necessidade. O facto é que a maioria dos estudos assume que os dispositivos são estáticos e que a sua posição não se altera desde do ponto de instalação inicial.

Ao utilizar nós móveis na conceção da RSSF podemos elevar a performance de cobertura de uma rede estacionária, não obstante que um nó se pode deslocar do seu ponto inicial por motivos relativos ao meio ambiente onde se encontram como fortes ventos, ondas ou animais, a aquisição de nós que suportem mobilidade é geralmente mais dispendiosa que os nós estáticos. Contudo, a adição de dispositivos com capacidades móveis pode melhorar e garantir a total cobertura de uma rede de forma continua cobrindo regiões que podem não ser alcançáveis pelos nós estáticos ou simplesmente ser utilizados como dispositivos de recolha de dados de outros dispositivos estáticos instalados numa área de interesse com dimensões elevadas.

Como exemplo deste tipo de abordagem temos algoritmo proposto em [16] que após a formação inicial da rede, cada dispositivo sensor tem a capacidade de determinar a melhor posição para onde se deve mover de modo a garantir máxima cobertura, contudo é necessário que este dispositivo esteja dentro do alcance do raio de cobertura de outros dispositivos para determinar a sua localização relativa e determinar qual o ponto ideal onde se deve colocar. Em [17] encontramos um método diferente em que cada nó comunica com os seus vizinhos de forma a determinar uma distância específica que os pode distanciar

de maneira a maximizar a cobertura e manter a conectividade, ou seja, os nós afastam-se uns dos outros após a distribuição inicial.

O serviço que se propõe nesta dissertação, pode trazer progressos para este tipo de problema, visto que além de sugerir a posição dos dispositivos, com vista a atingir máxima cobertura e conectividade de uma rede, possibilita a comunicação direta com o sistema e interagir diretamente com os dispositivos presentes na rede para que estes se posicionem de forma automática.

3.2.5 Distribuição e organização dos dispositivos

A área onde será instalada a rede e o tipo de função se espera que ela cumpra, vão ditar os seus requisitos de cobertura e conectividade. Um sistema de deteção de intrusos em que se forma uma barreira lógica com recurso a sensores distribuídos em linha para garantir uma zona intrespessável terá uma distribuição, e conseqüente cobertura de área diferente, de um sistema de recolha com fins estatísticos de temperatura ambiental numa floresta ou num deserto. Na organização e instalação de dispositivos no espaço de interesse pode ser feita de maneira manual ou com recurso a máquinas ou certos utensílios e podem ser meticulosamente posicionados em lugares cuidadosamente estudados, ou distribuídos aleatoriamente pela área que se deseja monitorizar. A distribuição aleatória pode por vezes ser obrigatória devido às características do ambiente, seja por difícil acesso ou por condições ambientais como por exemplo o interior de um vulcão. O posicionamento dos dispositivos tem bastante influência na operação e performance de uma RSSF seja na maximização de cobertura da mesma ou na conservação da sua energia gasta na troca de informação entre nós. Os dispositivos possuem normalmente a capacidade de observar eventos com interesse e transmitir a informação obtida para um serviço central através de protocolos de transmissão desenhados para este tipo de redes. A distribuição dos nós numa rede afeta também a gestão de recursos [18].

A distribuição dos nós de uma RSSF tendo como foco principal a cobertura máxima de uma área e a conectividade entre os seus dispositivos tem sido um dos temas bastante investigados dentro da área das RSSF, tendo surgido estudos e várias contribuições de soluções e algoritmos para a resolução deste tipo de problema[19]. Em [20] os autores apresentam um método eficiente para distribuir e implementar sensores numa determinada área com recurso a um robot, neste método é assumido que a área de interesse pode conter obstáculos que dificultam não só a recolha de informação como a troca da mesma pelos dispositivos. Descrevem um algoritmo com uma política para determinar o posicionamento de cada nó e uma forma de os distribuir que denominam de serpente, e

um método para lidar com os obstáculos que possam aparecer no caminho. Tem como objetivo maximizar a cobertura de uma área com o número mínimo de nós possível.

Têm vindo a ser propostos vários métodos de distribuição aleatória de sensores mas existem alguns problemas associados a este tipo de distribuição. Normalmente o número de nós que é necessário para se obter uma cobertura total é elevado quando comparado com o método determinístico, por outro lado, a densidade dos nós não é homogénea podendo existir áreas com maior densidade de nós do que outras e fazer com que se tenha um desperdício de hardware numas zonas e falta de cobertura em algumas zonas mais esparsas. A distribuição aleatória de sensores sobre a área que se deseja monitorizar é a forma mais fácil mas, contudo, pode levar a um aumento do custo de hardware assim como não consegue garantir que a área seja completamente coberta. Algumas propostas têm também em consideração alguns dispositivos com capacidade motora que depois de uma primeira implementação no terreno, têm a capacidade de se mover para zonas ótimas que são determinadas pela avaliação dos dispositivos vizinhos e da área a monitorizar, cobrindo assim zonas que não se encontram sob a monitorização de nenhum sensor. Na vertente deste projeto, sendo que será direccionado para a indústria, espera-se que os dispositivos sejam distribuídos de forma organizada e que seja possível a sua instalação manualmente, com o intuito de garantir uma cobertura óptima e conectividade da rede.

A maior parte da literatura concentra-se no posicionamento determinístico dos nós e tem como objetivo determinar a distribuição óptima da rede, ou seja determinar um padrão ótimo. Nesta abordagem são utilizados diferentes tipos de critérios para a disposição de todos os dispositivos de acordo com a aplicação e objetivos da RSSF. O principal objetivo da maioria destes trabalhos é assegurar a cobertura total, ou ótima, de uma área com o mínimo de nós possível.

Em [21] garantem que a implementação de uma rede em forma de grelha é uma boa abordagem ao problema da cobertura, devido à sua simplicidade e escalabilidade. A distribuição em forma de grelha garante um bom funcionamento da rede no que toca à cobertura e por isso é um bom esquema a ter em conta no planeamento da RSSF.

Tendo como base a construção de uma rede em forma de grelha, este artigo [22] apresenta e compara 4 tipos de formato de rede; grelha quadrada, retangular, triangular e hexagonal. Os autores comparam os quatro esquemas segundo a quantidade de área que cada um cobre e concluem que uma RSSF no formato hexagonal, garante uma melhor cobertura de área que os outros esquemas abordados. Outros autores em [23] com a mesma conclusão também afirmam que a distribuição em forma de hexágono fornece uma melhor

performance à rede, cobrindo a área com o mínimo de sensores possível.

Em [24, 25, 26] os nós encontram-se em forma de grelha mas, esta disposição pode não ser a mais eficiente, principalmente quando os nós possuem diferentes raios de comunicação e de cobertura o que dificulta não só o planeamento da rede como o aproveitamento das capacidades de cada dispositivo.

Um outro problema que se identifica neste método prende-se com a existência de obstáculos que podem ter influência na disposição da rede e de certo modo descaracterizar a grelha. Em [27] são considerados vários tipos de obstáculos que são definidos por varias figuras geométricas. São especificados círculos, retângulos e linhas que são combinados de modo a simular um objeto real presente na área. Em [28] é apresentado um algoritmo de distribuição de dispositivos ao longo de uma área assumindo que os nós podem ser implementados de forma manual, nas posições mais apropriadas aos requisitos da rede. Para considerarem obstáculos presentes na área a monitorizar, eles dividem a área a observar em partes mais pequenas de modo a colocar lá os dispositivos.

No artigo [29], os autores apresentam um método de distribuição de nós denominado r-strip, onde cada dispositivo sensor está separado pelo valor do raio de monitorização de cada sensor. O maior problema deste método é que se torna impraticável tentar implementar uma rede com tal formação, principalmente quando a área a monitorizar é grande ou quando existe um número muito elevado de sensores que pode elevar o custo da RSSF para valores incomportáveis. Em [30] podemos encontrar vários algoritmos para cobertura ótima de uma rede de sensores mas todos eles têm como pressuposto que se conhece a localização de cada dispositivo.

Capítulo 4

Desenho e Implementação

Neste capítulo vamos descrever todo o processo de implementação do projeto. Para isso, começamos com uma secção explicativa sobre as tecnologias usadas e posteriormente descrevemos o desenvolvimento das funcionalidades presentes, bem como as técnicas utilizadas para esse efeito.

É assumido que se conhece as dimensões da área a monitorizar, definidas por dois inteiros que representam a altura e comprimento. A área pode, ou não, conter obstáculos que são representados por quadrados ou retângulos e polígonos resultantes da junção destas duas figuras. Os obstáculos são definidos através de quatro inteiros, dois representam as duas dimensões e os restantes dois indicam a sua posição na área. Como já foi referido em capítulos anteriores, vamos considerar que cada sensor é representado por uma circunferência e que o seu raio é definido pelo valor do alcance de monitorização do próprio dispositivo. Será ainda considerada uma segunda circunferência que diz respeito ao alcance do rádio de cada dispositivo. Assume-se também que todos os dispositivos utilizados na constituição da rede são homogêneos ao nível de hardware.

Toda a informação acerca das características dos dispositivos são conhecidas e alcançáveis devido à utilização de tecnologias como OSGI e UPnP, fornecendo a possibilidade de comunicação e troca de informações entre todos os componentes da rede. Está disponível na rede a informação de cada agente e dos serviços que estes disponibilizam.

4.1 Tecnologias

As ferramentas e tecnologias utilizadas no desenvolvimento do projeto foram as seguintes:

4.1.1 JAVA

A linguagem escolhida para desenvolver este projeto foi o Java, desenvolvida na década de noventa pela Sun Microsystems. É uma linguagem de programação orientada a objetos e caracteriza-se pela sua portabilidade e por possuir um vasto número de bibliotecas com as mais variadas funcionalidades. A sua sintaxe é parecida ao C++ e é atualmente uma das linguagens mais usadas no mundo. Foi escolhida por ser compatível com outra tecnologia utilizada, o OSGi.

4.1.2 Universal Plug and Play - UPnP

O Universal Plug and Play surgiu em 1999, com a criação da organização denominada Fórum UPnP, que é a principal responsável pela conceção, gestão e melhorias da arquitetura UPnP. O principal foco desta tecnologia é garantir uma conexão direta, dinâmica e simples entre os vários componentes que fazem parte de uma rede. O UPnP é um conjunto de protocolos de rede de computadores que permitem a ligação automática entre um computador e um qualquer dispositivo presente na rede. Os dispositivos que suportam esta tecnologia são capazes de procurar, encontrar e usar os serviços disponibilizados por outros dispositivos UPnP, sem ter de efetuar qualquer tipo de configuração de rede. Este conceito de conectividade universal torna a interação entre quaisquer componentes, presente na rede, completamente independente do tipo de rede, do tipo de sistema operacional e da linguagem de programação utilizada. O UPnP utiliza protocolos abertos como o TCP/IP, HTTP e XML, sendo este último utilizado nas descrições de dispositivos e serviços, mensagens de controlo e eventos. As principais vantagens associadas a esta tecnologia incidem no facto de ser independente da linguagem utilizada, desde que siga as especificações do padrão, ser multiplataforma e não necessitar de configuração, automatizando todos os processos de conexão entre dispositivos. Permite também alcançar facilmente o conceito de *“plug-and-produce”* no *shop-floor* que possibilita a troca de equipamento como uma mote, máquina ou computador industrial de forma a estarem prontos a realizar a sua tarefa específica.

4.1.3 Open Service Gateway initiative - OSGi

Open Service Gateway initiative é uma especificação criada e mantida pela OSGi Alliance que foi fundada em 1999, por empresas como IBM, SUN, Motorola, entre outras. Hoje em dia são dezenas as empresas, como Adobe, Mitsubishi, Siemens ou Oracle apoiam o desenvolvimento e evolução desta plataforma que é utilizada em automóveis, servidores, automatização industrial, entre outros. O OSGi tem como objetivo reduzir a complexidade do *software*, ao fornecer uma arquitetura modular e orientada ao serviço para grandes sistemas distribuídos ou mesmo pequenas aplicações. Com esta tecnologia, as aplicações ou componentes, que são definidos como *bundles*, podem ser instalados remotamente, iniciados, atualizados e removidos do sistema, sem que seja necessário reiniciar ou parar o serviço. A especificação cria uma camada chamada de Ambiente OSGi, e possibilita de maneira transparente o acesso ao ambiente. Especificação que define como é que os *bundles* devem ser instalados, geridos e removidos. Além de reduzir a complexidade, ser dinâmico e de fácil implementação, é também uma tecnologia simples, compacta e rápida, o que fez despoletar um enorme interesse na comunidade desenvolvedora no seu uso. Neste projeto utilizamos a *framework* Felix, não só por existir um vasto leque de documentação e ser *open-source*, mas também por incluir especificações de implementação UPnP, que é outra tecnologia utilizada no projeto. O Apache Felix surge do esforço da comunidade para implementar OSGi e serviços relacionados sobre a licença Apache.

4.1.4 JAXB

Java Architecture for XML Binding ou simplesmente JAXB é uma especificação que fornece a possibilidade de mapear classes JAVA em ficheiros XML e vice-versa. É uma API que faz parte da plataforma JAVA SE e torna relativamente fácil manusear ficheiros com o formato XML no JAVA. Ao processo de transformação de um objeto, que é definido como *JAXElement*, num formato XML é denominado de *Marshal*. O método que fornece este o *Marshal* é o *JAXBContext* e o responsável por construir uma instância do *JAXElement* que seja apropriada para o tipo de objeto a ser serializado é o *ObjectFactory* criado pelo *Binding Compiler*. Quando pretendemos, pelo contrário, construir um objeto com dados presentes num ficheiro XML, além de utilizarmos o *JAXBContext* recorreremos ao *Unmarshaller*. O *Unmarshaller* é o responsável por receber um ficheiro XML e transforma-lo, devolvendo um *JAXBElement* que representa um objeto JAVA. Esta especificação foi escolhida para lidar com a comunicação DIL utilizada no projeto para comunicação entre atores da rede e que reside na troca de ficheiros XML, quer para anunciar serviços como para requerimento de tarefas e resposta às mesmas.

4.2 Implementação

O método de funcionamento do algoritmo de distribuição de dispositivo construído baseia-se em dois trabalhos [28, 20], já aqui mencionados e que achamos ser úteis e eficientes para este projeto. Ambos tratam do problema da cobertura de uma rede de sensores sem fios com o objetivo de alcançarem uma cobertura ótima da área de interesse, com o mínimo de dispositivos possível tendo em consideração a presença de obstáculos no meio que possam de alguma forma influenciar o bom funcionamento do sistema.

Neste projeto assume-se que todos os dispositivos são homogêneos, ou seja, todos eles possuem características iguais ao nível físico como o alcance máximo do sensor e o alcance máximo para comunicação com outras entidades. É também assumido que todas as entidades constituintes da rede estão aptas a comunicar via UPnP e que o serviço aqui implementado se encontra incluído na arquitetura SmartComponent, desenhada especificamente para o projeto I-RAMP3. A área a monitorizar é representada por um quadrado ou retângulo e consideramos o ponto (0,0) o seu canto superior esquerdo.

O algoritmo utilizado divide-se em duas partes, a fase de planeamento onde se define a melhor posição que os dispositivos devem ter para que se alcance o objetivo final e a fase de distribuição propriamente dita que trata de colocar os dispositivos ao longo da área tendo em conta as características tanto da área como dos dispositivos. Segue-se uma explicação mais pormenorizada do algoritmo.

4.2.1 Fase de Planeamento

A fase de planeamento da distribuição de sensores segue o modelo adotado em [28], que tem por base as características dos dispositivos como o raio de cobertura do sensor (R_s) e o raio de conectividade da mote (R_c) para formular um conjunto de formulas matemáticas que fornecem o valor das distâncias máximas que os dispositivos devem possuir entre eles. Os valores definidos não só garantem a cobertura máxima da área de interesse bem como garantem a conectividade da rede de forma eficiente, recorrendo ao menor número de dispositivos possível.

Deste método de planear a rede de sensores sem fios resultam variáveis importantes que servem de parâmetros para a fase seguinte do algoritmo que trata da distribuição de sensores, variáveis essas que iremos denominar como; incremento X (Inc_X), incremento Y (Inc_Y), distância máxima NorteSul (dm_{NS}) e distância máxima EsteOeste (dm_{EO}). O incremento X e o incremento Y, como o próprio nome indica, dizem respeito à distância

entre os nós segundo o eixo dos x 's e y 's respetivamente. Os valores dm_{NS} e dm_{EO} ditam o quanto podem os dispositivos estar distanciados das fronteiras da área ou dos obstáculos de maneira a garantir a total cobertura da rede nestas zonas, método este que iremos ver mais para à frente.

A fase de planeamento prende-se em determinar estas variáveis que resultam da comparação entre os raios do dispositivo de onde surgem 4 planos a seguir:

- $R_s > R_c$: Quando esta condição se verifica, poderá ser necessário adicionar dispositivos extra entre as linhas de sensores de modo a alcançar a conectividade entre toda a rede. Como a área se encontra completamente coberta e obtemos conectividade em cada linha de dispositivos, decidimos deixar essa opção ao administrador da rede que tratará de avaliar a necessidade e a melhor forma de se adicionar qualquer dispositivo extra. Como podemos verificar na figura 4.1, quando $R_s > R_c$ obtemos um incremento X de R_c e um incremento Y de $r_s + \sqrt{R_s^2 - \frac{R_c^2}{4}}$. As distâncias utilizadas para o tratamento de fronteiras, digam elas respeito à área ou aos obstáculos tomam os seguintes valores: $dm_{NS} = \sqrt{R_s^2 - \frac{R_c^2}{4}}$ e $dm_{EO} = \frac{R_c}{2}$;

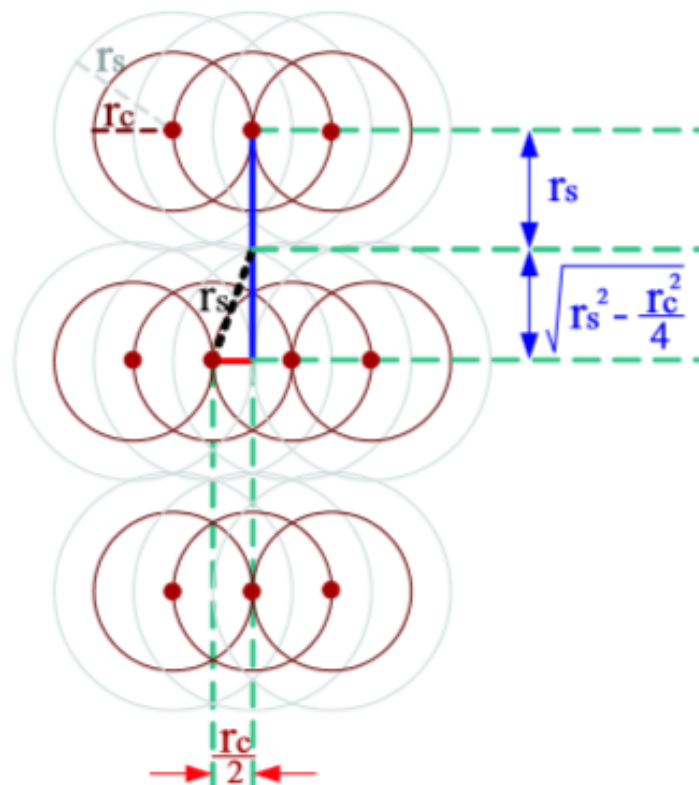


Figura 4.1: Planeamento da RSSF quando $R_s > R_c$

- $R_s = R_c$: Como representado na Figura 4.2, quando os raios do dispositivo têm o mesmo alcance, os dispositivos estão distanciados ao longo do eixo dos x 's por R_s e cada linha de dispositivos está separada ao longo do eixo dos y 's por $R_s + \frac{\sqrt{3}}{2}R_s$. As distâncias a considerar nas fronteiras NorteSul e EsteOeste são $\frac{\sqrt{3}}{2}R_s$ e $\frac{R_s}{2}$ respectivamente.

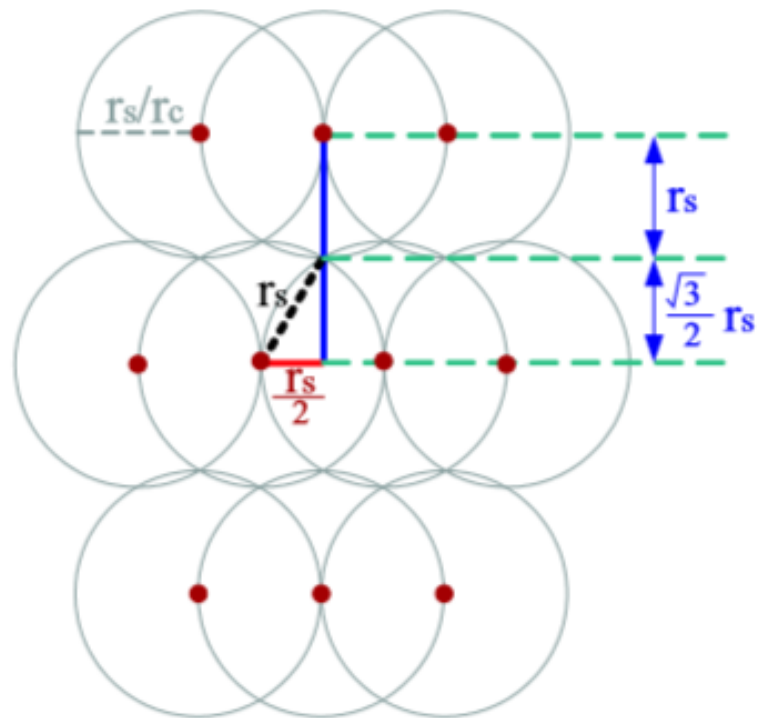


Figura 4.2: Planeamento da RSSF quando $R_s = R_c$

- $R_s < R_c \leq \sqrt{3}R_s$: Nesta condição o dispositivos estão distanciados pelo valor do R_c logo este será o incremento de X. As restantes variáveis tomam os seguintes valores, como ilustrado na Figura 4.3: incremento de Y é $r_s + \sqrt{R_s^2 - \frac{R_c^2}{4}}$, a dm_{NS} é $\sqrt{R_s^2 - \frac{R_c^2}{4}}$ e dm_{EO} tem o valor $\frac{R_c}{2}$.

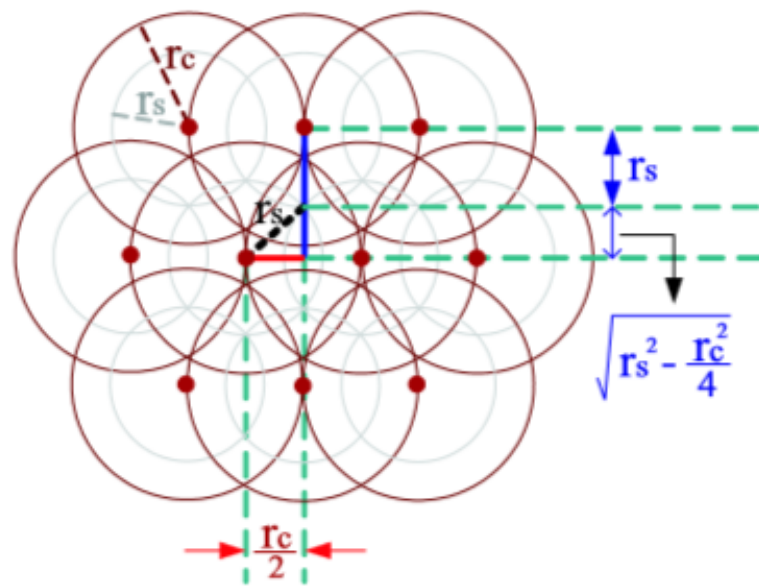


Figura 4.3: Planeamento da RSSF quando $R_s < R_c \leq \sqrt{3}R_s$

- $R_c > \sqrt{3}R_s$: Segundo os autores de [28], uma área encontra-se otimamente coberta se a rede for construída em forma de grelha e os seus dispositivos estão posicionados de maneira a formar triângulos equiláteros em que cada lado mede $\sqrt{3}R_s$. Como nesta condição o R_c é maior que este valor assumimos que a conectividade está assegurada e a rede é construída segundo este princípio. Como ilustra a Figura 4.4 as variáveis obtidas nesta condição são: incremento X é $\sqrt{3} * R_s$, o incremento Y $R_s + \frac{R_s}{2}$, a dm_{NS} tem o valor de $\frac{R_s}{2}$ e a dm_{EO} é igual a $\frac{R_s}{2}R_s$.

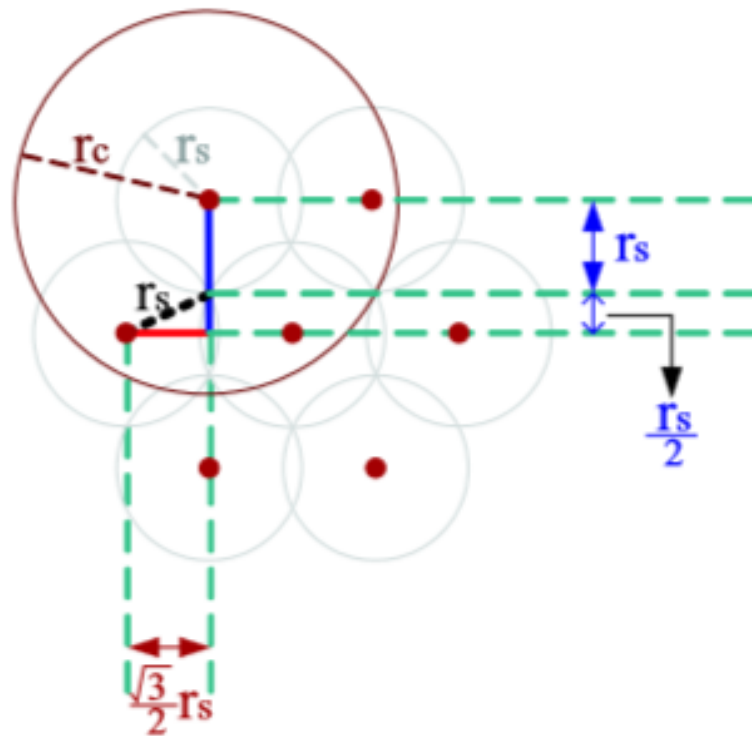


Figura 4.4: Planeamento da RSSF quando $R_c > \sqrt{3}R_s$

4.2.1.1 Vizinhos

Consideramos que cada dispositivo tem uma coordenada cartesiana (x,y) associada, que define a sua posição exata na área a monitorizar. Conseguimos desta forma determinar as posições dos seis dispositivos considerados os nós vizinhos de um determinado ponto. Segundo uma distribuição em forma de hexágono, os dispositivos estariam dispostos como demonstra a Figura 4.5.

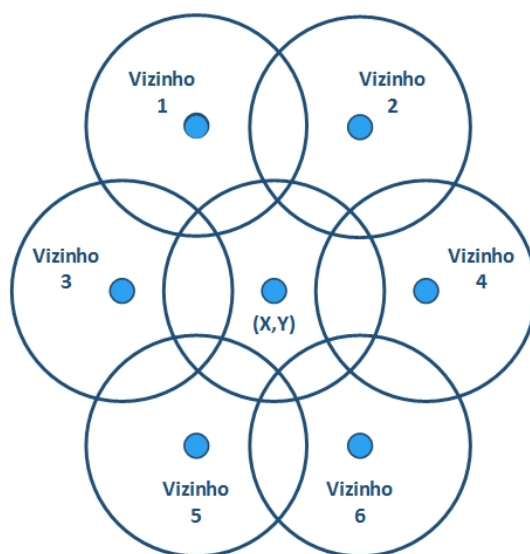


Figura 4.5: Vizinhos de um dispositivo, Distribuição em forma de hexágono

Se a distribuição das motes pela área ocorrer de forma normal e não existir nenhum obstáculo a interferir com a sua disposição, tendo em conta o planeamento efetuado e as variáveis obtidas, podemos afirmar que as posições dos seis nós vizinhos de um dispositivo estão representados na Tabela 4.1.

Vizinhos	Posição X	Posição Y
Vizinho1	$x - \frac{IncX}{2}$	$y - IncY$
Vizinho2	$x + \frac{IncX}{2}$	$y - IncY$
Vizinho3	$x - IncX$	y
Vizinho4	$x + IncX$	y
Vizinho5	$x - \frac{IncX}{2}$	$y + IncY$
Vizinho6	$x + \frac{IncX}{2}$	$y + IncY$

Tabela 4.1: Posições normais dos seis vizinhos de um dispositivo

4.2.2 Fase de distribuição

A distribuição dos dispositivos é baseado no trabalho [20] que apresenta um algoritmo desenhado para um robot com capacidade para instalar dispositivos automaticamente sobre uma determinada área, construindo uma RSSF. A junção do planeamento anteriormente descrito com este algoritmo faz com que se obtenha uma solução para vários tipos de problema se considerarmos as variâncias das características físicas de cada mote. Todos os dispositivos são representados através de *shapes* circulares do JAVA que contêm dois tipos de raios, R_s e R_c e um ponto cartesiano (x,y) que define o seu centro. A instalação e distribuição de nós segue as seguintes regras:

4.2.2.1 Tipos de Movimento

A implementação da rede de sensores sem fios segue uma metodologia em que os dispositivos são colocados na área um de cada vez e a partir da posição de cada mote instalada são calculadas as seis posições mais próximas e passíveis de instalação de novos dispositivos segundo a tabela 4.1. Segundo esta metodologia conseguimos identificar seis tipos de movimentos que o algoritmo está apto a realizar se realizada uma distribuição normal.

1	2	3	4	5	6
----->	-----<	↖ 	↗ 	↓ 	↓

Figura 4.6: Tipos de movimento usados

Na figura 4.6 podemos identificar os seis tipos de movimento aqui abordados, os quais podemos associar aos vizinhos identificados na tabela 4.1. Ao compararmos o tipo de movimento com os vizinhos definidos na tabela concluímos que o movimento 1 identifica o vizinho4, o movimento 2 representa o vizinho3, o movimento 3 associa-se ao vizinho1, o movimento 4 identifica o vizinho2, os movimentos 5 e 6 representam os vizinho5 e o vizinho6 respetivamente.

4.2.2.2 Prioridades

Algoritmo segue um conjunto de prioridades que definem o caminho a seguir, ou mais concretamente o próximo sensor a ser instalado na área. Existem dois tipos de movimentos básicos que definem a direção que o algoritmo está a tomar, sejam eles os movimentos 1 e 2 identificados na Figura 4.6, que representam a direção Este e Oeste respetivamente. Este conjunto de prioridades tem como objetivo garantir a distribuição em forma de grelha como identificado na Figura 4.7. Note-se que as setas vermelhas identifica a direção que o algoritmo toma sempre que dispõe um dispositivo sobre a área, sendo que o primeiro dispositivo implementado situa-se no ponto (dm_{EO}, dm_{NS}) , variáveis obtidas na fase de planeamento e garantem a total cobertura do canto superior direito.

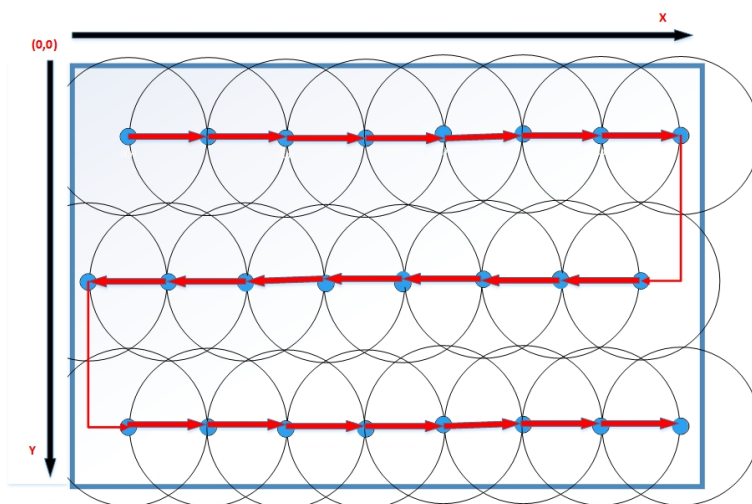


Figura 4.7: Distribuição de dispositivos sobre uma área

Como referimos anteriormente, o algoritmo segue apenas duas direções, Este e OESTE. É mediante a direção em que o algoritmo se encontra na implementação de um sensor que são definidas prioridades de movimentos que identificam a ordem de verificação da possibilidade de se instalar os dispositivos, ou seja, quando um dispositivo é colocado na área de interesse, o algoritmo testa se é ou não possível implementar o próximo sensor mediante a lista de prioridades que identificamos na Figura 4.8.

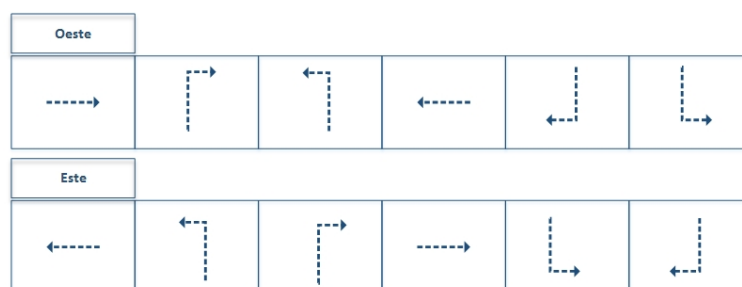


Figura 4.8: Prioridades

A implementação de um sensor na área é condicionada por três fatores que são verificados pelo algoritmo sempre que este tenta efetuar a próxima movimentação:

- se já existe um sensor instalado na posição definida como o próximo movimento a considerar da lista de prioridades;
- se na próxima posição a verificar existe um obstáculo;
- se a posição seguinte, segundo a lista de prioridades, se encontra fora da fronteira da área de interesse.

Sempre que uma tentativa de movimentação passa nestes três requisitos, o algoritmo segue para essa posição, instala um novo dispositivo na área e testa novamente todas as posições segundo a lista de prioridades definida para a direção em que se encontra a atuar. Quando o algoritmo percorre toda a lista de prioridades e nenhuma das posições satisfaz os requisitos mencionados acima, recomeça desde o último dispositivo instalado percorrendo do início a lista de prioridades.

4.2.2.3 Regras para resolução de problemas com fronteiras

Um dos problemas deste tipo de distribuição é garantir a cobertura da rede junto às fronteiras, seja da própria área ou de obstáculos que podem de alguma forma descaracterizar a grelha da rede.

Os obstáculos são representados como tipos *shape* do *JAVA* que podem ser quadrados ou retangulares. De maneira a obter-se obstáculos mais complexos é possível juntar várias *shapes* de forma a contruir o formato mais adequado como demonstrado na Figura 4.9.

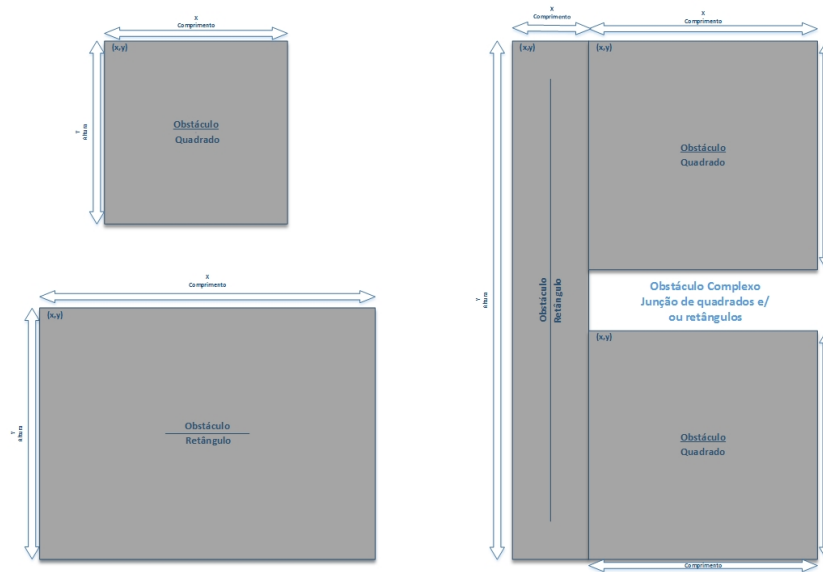


Figura 4.9: Representação de obstáculos

Sempre que o algoritmo coloca um dispositivo e tenta movimentar-se para um ponto situado fora da área ou pertencente a um obstáculo, essa movimentação falha e verifica-se se essa parte fronteira está a ser totalmente monitorizada pela rede. Para isso o algoritmo recorre às distâncias dm_{NS} e dm_{EO} obtidas na fase de planeamento decide se é necessário adicionar algum dispositivo extra à rede para garantir uma total cobertura. As regras de verificação diferem segundo o tipo de movimento efetuado, por exemplo, na Figura 4.10 o algoritmo segue a na direção ESTE e tenta movimentar-se para fora da área, um sensor é adicionado na fronteira se a distância entre o último dispositivo instalado e a fronteira da área for superior a dm_{EO} senão, é avaliada a próxima movimentação da lista de prioridades. Note-se que o mesmo procedimento acontece quando o algoritmo segue a na direção contrária, neste caso OESTE, ou quando se tenta movimentar para uma posição pertencente a um obstáculo.

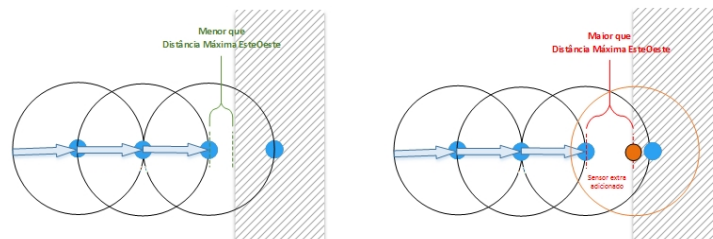


Figura 4.10: Resolução de problema na fronteira lateral

Foram definidas duas regras para lidar com este tipo de problemas para cada tipo de movimentação, a regra da fronteira 1 (Regra F1) que valida os movimentos 1 e 2 e a regra da fronteira 2 (Regra F2) que verifica os movimentos 3, 4, 5 e 6.

Regra F1

A Figura 4.11 serve de cenário para a explicação desta regra. Neste caso o algoritmo tenta movimentar-se do ponto 1 para o ponto 2 que está contido no obstáculo. A regra é definida por:

- Se a distância d for maior que dm_{EO} é colocado um dispositivo extra na fronteira da área ou obstáculo.
- Caso contrário procede com a distribuição desde o ponto 1.

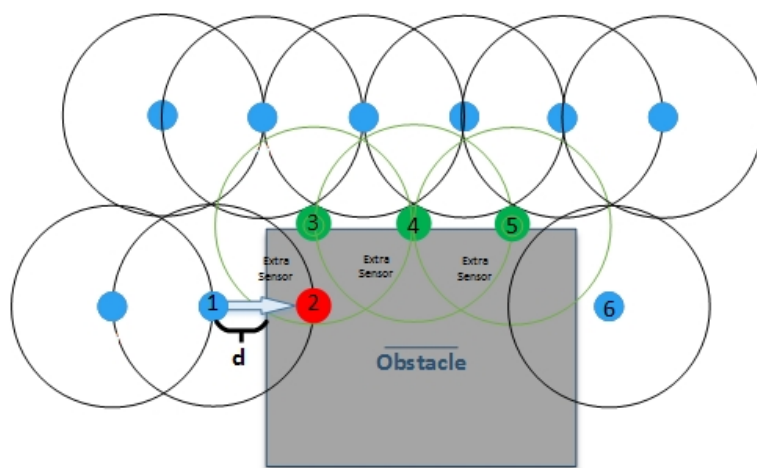


Figura 4.11: Resolução de problema na fronteira lateral de obstáculos

Na Figura 4.11 podemos observar também uma particularidade desta regra quando a movimentação falha devido presença de um obstáculo. Neste caso, é verificado se a fronteira superior do obstáculo se encontra a ser monitorizada pela rede, ou seja, depois de verificar fronteira lateral do obstáculo, se a fronteira superior não se encontra coberta por nenhum sensor são colocados sensores extra de modo a garantir essa mesma cobertura, como demonstrado pelos pontos 3, 4 e 5. Se esta situação ocorre o algoritmo continua com a distribuição desde o ponto 6 da figura, note-se que o procedimento é igual para qualquer uma das direções que o algoritmo pode tomar, ESTE ou OESTE.

Regra F2

Esta regra lida com os problemas da fronteira associados às movimentações do tipo 3, 4, 5 e 6. Tem o mesmo procedimento quer se trate da fronteira de um obstáculo ou da área e independentemente da direção em que se encontra a distribuição, ESTE ou OESTE.

Movimentos tipo 3 e 4

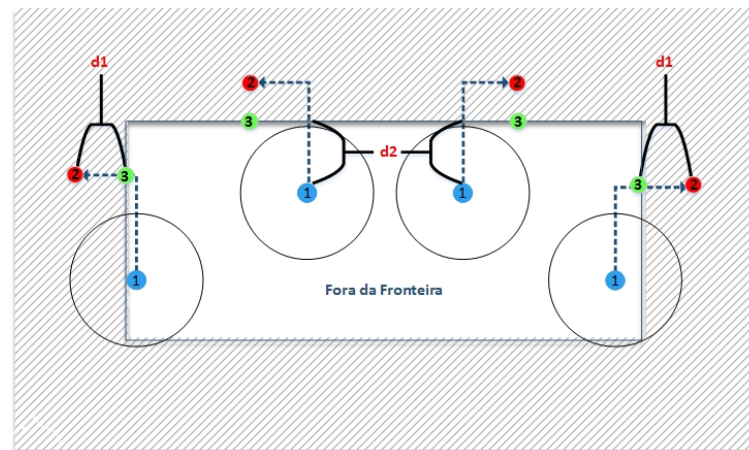


Figura 4.12: Regra fronteira da área para movimentos tipo 3 e 4

As Figuras 4.12 e 4.13 ilustram este problema quando ocorre uma movimentação deste tipo e ela falha devido à fronteira da área ou de um obstáculo respetivamente. Quando o algoritmo tenta movimentar-se da posição 1 para a posição 2 que se encontra sobre um obstáculo ou fora da área de interesse:

- é adicionado um dispositivo extra no ponto 3 se $d1 < dm_{EO}$ ou $d2 > dm_{NS}$ dependendo da fronteira que está a ser avaliada.
- Caso contrário procede com a distribuição desde o ponto 1.

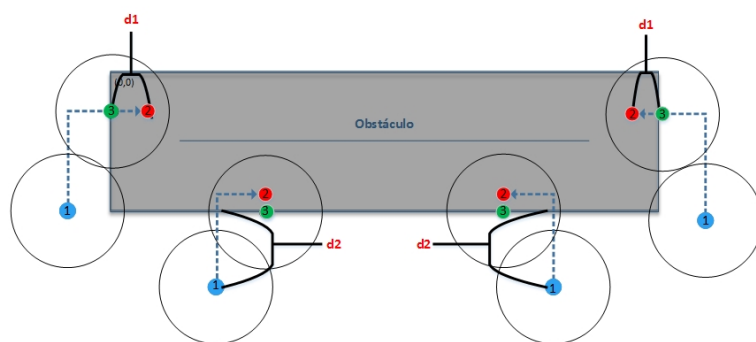


Figura 4.13: Regra fronteira do obstáculo para movimentos tipo 3 e 4

Movimentos tipo 5 e 6

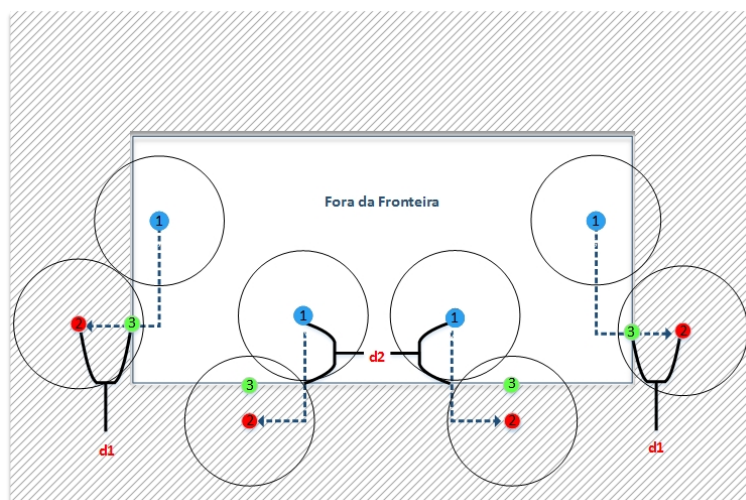


Figura 4.14: Regra fronteira da área para movimentos tipo 5 e 6

Da mesma forma as Figuras 4.14 e 4.15 ilustram este problema quando existe uma tentativa de movimentação do tipo 5 ou 6 que falha devido à fronteira da área ou de um obstáculo respetivamente. Quando o algoritmo tenta movimentar-se da posição 1 para a posição 2 que se encontra sobre um obstáculo ou fora da área de interesse:

- é adicionado um dispositivo extra no ponto vermelho 3 se $d1 < dm_{EO}$ ou $d2 > dm_{NS}$ dependendo da fronteira que está a ser avaliada.
- Caso contrário procede com a distribuição desde o ponto 1.

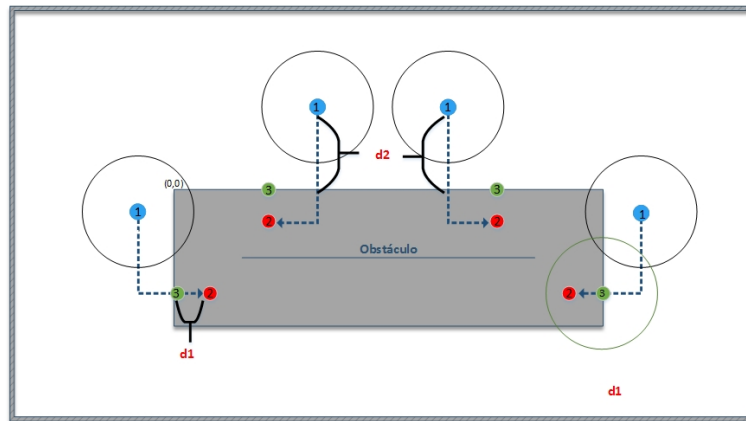


Figura 4.15: Regra fronteira do obstáculo para movimentos tipo 5 e 6

4.2.2.4 Código do Algoritmo

O algoritmo base consiste na utilização de todos os processos e métodos até aqui referidos, tendo como *input* as variáveis obtidas na fase de planeamento que alimentam a fase de distribuição e tratamento de obstáculos bem como as fronteiras da área.

A sua implementação assenta em quatro regras que avaliam e executam as movimentações permitidas e definidas na Figura 4.16. Sempre que um dispositivo é instalado na área, são verificadas cada uma das quatro regras de maneira a determinar o próximo passo e direção a tomar. Se por alguma razão a movimentação para uma direção com alta prioridade falhar, o algoritmo executa a regra da fronteira correspondente e segue para o próximo movimento na lista de prioridades, caso contrário avança para a posição definida pelo tipo de movimento e recomeça com a verificação das regras.

Direção	Movimento 1	Movimento 2		Movimento 3	Movimento 4	
Oeste	→	↖	↗	←	↙	↘
Este	←	↖	↗	→	↘	↙
	1ª Prioridade	2ª Prioridade	3ª Prioridade	4ª Prioridade	5ª Prioridade	6ª Prioridade

Figura 4.16: Prioridades de movimentos efetuados na distribuição de dispositivos

As quatro regras determinam a forma como a distribuição dos dispositivos é realizada e asseguram a distribuição em forma de serpente já mencionada e o tratamento dos obstáculos presentes no meio. A sua definição tem em consideração os movimentos descritos na Figura 4.16 e são definidas da seguinte forma:

Regra 1

- Verifica a possibilidade de se efetuar o Movimento 1;
- No caso de falhar a movimentação, avança para a Regra 2;
- Caso contrário, efetua o movimento e segue nessa direção;

Regra 2

- Verifica a possibilidade de executar o Movimento 2;
- Se falhar a movimentação segue para a Regra 3;
- Caso contrário, avança para a posição determinada pelo Movimento 2;

Regra 3

- Verifica se é possível realizar o Movimento 3;
- Se falhar, o algoritmo segue para a Regra 4;
- Caso contrário, efetua o Movimento 3;

Regra 4

- Verifica a possibilidade de avançar com o Movimento 4;
- Em caso de falha na movimentação corre todas as regras no último dispositivo instalado;
- Caso contrário, efetua o Movimento 4;

A construção do algoritmo resulta da execução destas quatro regras mencionadas bem como das regras definidas anteriormente para lidar com o problema das fronteiras. O Algoritmo 1 representa uma demonstração do pseudo-código do serviço aqui proposto.

Algoritmo 1 Construção de uma RSSF com cobertura máxima e garantia de conectividade

Entrada: Variáveis obtidas na fase de planeamento, dimensões da área e obstáculos**Saída:** Configuração de uma RSSF, Posições dos dispositivos**início**

```
    enquanto Existir área por monitorizar faça
        Executa Regra 1;
        se Regra 1.sucesso = false então
            Executa Regra F1;
            Executa Regra 2;
            se Regra 2.sucesso = false então
                Executa Regra F2;
                Executa Regra 3;
                se Regra 3.sucesso = false então
                    Executa Regra F1;
                    Executa Regra 4;
                    se Regra 4.sucesso = false então
                        Executa Regra F2;
                    senão
                        Troca de direção;
                    fim
                fim
            fim
        fim
    fim
```

4.3 Enquadramento do serviço na arquitetura - Comunicação

Num sistema Multi-Agente, a interação e comunicação entre os vários agentes que pertencem ao sistema, são parâmetros de grande importância. É necessária a implementação de um sistema de comunicação bem estruturado, que garanta uma correta interpretação por parte de todos os agentes, de modo a garantir uma compreensão mútua entre os vários dispositivos e os seus serviços.

Neste projeto assume-se que qualquer ator presente numa rede industrial é considerado um dispositivo UPnP o que facilita a comunicação entre todos os componentes da rede. O fluxo de informação segue o conceito utilizado no projeto I-RAMP3 ao usar DIL (*Device Integration Language*), uma linguagem específica que garante a comunicação entre os atores. O serviço encontra-se inserido numa arquitetura desenhada especificamente para suportar este tipo de sistema.

4.3.1 Arquitetura SmartComponent

SmartComponent [32] é uma arquitetura orientada ao serviço para redes de sensores industriais. Foi desenhada com o propósito de manter um registo de todos os dispositivos sejam, máquinas, motes ou *gateways*, sendo estes encapsulados como serviços da arquitetura. Essa encapsulação permite que os serviços consigam comunicar entre si e sejam orquestrados pela arquitetura para atingirem um fim próprio da aplicação. A tecnologia usada para desenvolver a arquitetura foi o OSGi, os componentes de orquestração de serviços são módulos de *software*, assim como os serviços complexos que a arquitetura disponibiliza. Esses serviços complexos, são componentes de *software* que encapsulados pelas interfaces da arquitetura SmartComponent, permitem interagir com os serviços virtualizados dos dispositivos. Tipicamente na indústria, são incluídos serviços de validação e agregação de dados assim como controlo e configuração. Este trabalho insere-se nos serviços de configuração e controlo, encapsulado com um *bundle* (componente de *software* que permite integração numa framework OSGi), integra a arquitetura permitindo interagir com os serviços de dispositivos existentes na rede.

4.3.2 Universal Plug and Play

A comunicação com o serviço aqui proposto pode ser realizada através de UPnP que fornece uma maneira fácil, rápida e intuitiva de aceder aos métodos do projeto. Todos os métodos do serviço estão disponíveis através de UPnP e podem ser acedidos por todos os atores com capacidade de comunicação sob esta tecnologia. Também é através de UPnP que se dá a comunicação de dispositivos do projeto I-RAMP3 compatíveis com a linguagem DIL, baseada em troca de informações em formato XML. Na imagem 4.17 podemos verificar a lista de serviços e métodos disponíveis e os argumentos que devem ser fornecidos para se obter a resposta esperada.

The image shows a screenshot of the Device Spy application. On the left, a tree view displays the service structure under 'um.schemas-upnp-org.service:AreaCoverage:1'. It lists several methods with their parameters, such as 'MakeWSN', 'MakeWsnAccuracy', 'MakeWsnDeviceList', and 'MakeWsnDeviceListAccuracy'. Below this, other services like 'AreaCoverageWithObstacles:1', 'NetDevService:1', and 'VerifyRequirements:1' are also visible.

On the right, a detailed view of the 'Area Coverage and WSN Requirements' service is shown. It includes the following metadata:

Base URL	http://192.168.1.68:4004/
Device icon	None
Device URN	um.schemas-upnp-org:device:service-manager:1
Embedded devices	0
Expiration timeout	1800
Friendly name	Area Coverage and WSN Requirements
Has presentation	True
Interface to host	192.168.1.68
Manufacturer	ISR
Manufacturer URL	http://paginas.fe.up.pt/isp/
Model description	A Device that acts as a controller and reconfigurator of services
Model name	Area Coverage and WSN Requirements
Model number	1.0
Model URL	http://paginas.fe.up.pt/isp/
Presentation URL	http://192.168.1.68:4004/
Product code	1213456789
Proprietary type	
Remote endpoint	192.168.1.68:4004
Serial number	0123
Services	4
Standard type	service-manager
Unique device name	AreaCoverageAndWsnRequirements:1
Version	1.0

Figura 4.17: Comunicação via UPnP - Device Spy

4.3.3 Device Integration Language

A elaboração de documentos XML (eXtensible Markup Language) que preenchem os requisitos necessários a uma boa interação entre os vários agentes, justifica-se por vários motivos. É simples e facilmente legível, quer por humanos quer por computadores, é portátil, ou seja, não depende de nenhuma plataforma de *hardware* ou *software* e além disso, existem diversas ferramentas capazes de transformar informação modelo em documentos baseados em XML e vice-versa. O fluxo de informação da rede contempla um conjunto de documentos definidos que podem ser recebidos e enviados por qualquer ator presente na rede. Os quatro documentos são abordados à frente sendo que em Apêndice B é apresentado o documento completo que descreve o serviço aqui proposto e que é parte integrante do sistema.

4.3.3.1 NetDev Service Description

O NetDev Service Description é um ficheiro XML que descreve qualquer ator presente na rede. Todas as entidades que constituem a rede devem possuir um NSD que anuncia as capacidades e os serviços de cada elemento, sendo que cada um dos membros deve possuir um. As informações que este ficheiro contém, dita o que cada serviço pode ou não fazer descrevendo com pormenor todo o *input* que será preciso fornecer para que seja calculada e apresentada uma solução.

Podemos verificar em 4.18 um exemplo deste tipo de ficheiros onde encontramos várias informações detalhadas da entidade que representa como o nome, o tipo de entidade, o modelo e o fabricante. No NSD estão definidos todos os serviços que o dispositivo é capaz de fornecer, os quais podem ser acedidos pelas restantes entidades da rede.

```
<?xml version="1.0"?>
<nsd xmlns="http://iramp-3.eu/nsd" xmlns:upnp="http://iramp-3.eu/nsd/dataConnectors/upnp"
xmlns:bus="http://iramp-3.eu/nsd/dataConnectors/fieldbus" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://iramp-3.eu/nsd ./nsd-base-1.0.xsd http://iramp-3.eu/nsd/dataConnectors/upnp upnp-connectors.xsd">
  <description>
    <name>Area Coverage And Wsn Requirments</name>
    <netdevType>/netdev/virtual/area/coverage</netdevType>
    <power unit="kW">999</power>
    <model>V1.0</model>
    <manufacturer>FCUP + FEUP</manufacturer>
    <serial>FCUP20132014FEUP</serial>
    <built>2014-01-30T09:00:00</built>
    <physicalProperties />
  </description>
  <configuration>
    <standardConfiguration sncd="/config/netdev/virtual/area/coverage"/>
    <vendorSpecificConfiguration sncd="/config/netdev/virtual/area/coverage"/>
  </configuration>
  <!-- The task range contains a list of task definitions in NSD, which can be executed by this netdev -->
  <taskRange>
    <taskDefinition name="getWSN" sntd="/task/netdev/virtual/getWSN">
    <taskDefinition name="makeWSN" sntd="/task/netdev/virtual/makeWSN">
    <taskDefinition name="makeWSN Device" sntd="/task/netdev/virtual/makeWSN Device">
    <taskDefinition name="makeWSN DeviceList" sntd="/task/netdev/virtual/makeWSN DeviceList">
    <taskDefinition name="makeWSN Accuracy" sntd="/task/netdev/virtual/makeWSN Accuracy">
    <taskDefinition name="makeWSN DeviceList Accuracy" sntd="/task/netdev/virtual/makeWSN DeviceList Accuracy">
```

Figura 4.18: Parte do Ficheiro NetDev Service Description

Esta lista de serviços ou tarefas que se encontra definida no NSD indica quais os métodos disponíveis e que são aceites pelo dispositivo, como fornece os parâmetros e condições que são aceites por este. Em 4.19 temos um exemplo desta mesma lista onde encontramos o nome de todos os serviços disponibilizados que são definidos como *TaskDefinitions*. Cada serviço tem definido o tipo de *input* e *output* que recebe e o tipo de valores que são aceites e que devem ser cumpridos para que seja apresentada uma solução.

```

<!-- The task range contains a list of task definitions in NSD, which can be executed by this netdev -->
<taskRange>
  <taskDefinition name="getWSN" sntd="/task/netdev/virtual/getWSN">
    <taskDefinition name="makeWSN" sntd="/task/netdev/virtual/makeWSN">
      <dataDefinitions>
        <!-- Input -->
        <dataDefinition id="area width">
          <metaData name="datatype">float</metaData>
          <metaData name="minValue">0</metaData>
          <metaData name="maxValue">1000000</metaData>
          <metaData name="unit">cm</metaData>
        </dataDefinition>
        <dataDefinition id="area length">
          <metaData name="datatype">float</metaData>
          <metaData name="minValue">0</metaData>
          <metaData name="maxValue">1000000</metaData>
          <metaData name="unit">cm</metaData>
        </dataDefinition>
        <dataDefinition id="sense radius">
          <metaData name="datatype">float</metaData>
          <metaData name="minValue">0</metaData>
          <metaData name="maxValue">1000000</metaData>
          <metaData name="unit">cm</metaData>
        </dataDefinition>
        <dataDefinition id="communication radius">
          <metaData name="datatype">float</metaData>
          <metaData name="minValue">0</metaData>
          <metaData name="maxValue">1000000</metaData>
          <metaData name="unit">cm</metaData>
        </dataDefinition>
        <!-- Output -->
        <dataDefinition id="position">
          <metaData name="datatype">position</metaData>
        </dataDefinition>
        <dataDefinition id="device">
          <metaData name="datatype">device</metaData>
        </dataDefinition>
        <dataDefinition id="devices">
          <metaData name="datatype">device</metaData>
          <metaData name="unit">array</metaData>
        </dataDefinition>
        <dataDefinition id="positions">
          <metaData name="datatype">position</metaData>
          <metaData name="unit">array</metaData>
        </dataDefinition>
      </dataDefinitions>
      </tddSchema>
      <qrdSchema>
        <result dataDefinitionRef="devices" />
        <result dataDefinitionRef="positions" />
      </qrdSchema>
    </taskDefinition>
  </taskDefinition>
  <taskDefinition name="makeWSN Device" sntd="/task/netdev/virtual/makeWSN Device">
  </taskDefinition>
  <taskDefinition name="makeWSN DeviceList" sntd="/task/netdev/virtual/makeWSN DeviceList">
  </taskDefinition>
  <taskDefinition name="makeWSN Accuracy" sntd="/task/netdev/virtual/makeWSN Accuracy">
  </taskDefinition>
  <taskDefinition name="makeWSN DeviceList Accuracy" sntd="/task/netdev/virtual/makeWSN DeviceList Accuracy">
  </taskDefinition>
</taskRange>

```

Figura 4.19: Parte do Ficheiro NetDev Service Description

4.3.3.2 Task Description Document

O Task Description Document é utilizado para descrever os objetivos da tarefa que se pretende efetuar. Além da obrigatoriedade de seguir o modelo definido para este género de ficheiros, ele tem de corresponder à estrutura de dados referente ao *input* definido no NSD do serviço para o método correspondente. Em 4.20 temos um exemplo de um TDD que pretende executar o método *MakeWSN_withObstacles* que é representado através de uma *task*. Cada *task* possui um identificador único e um nome, tal como o número de ciclos que se pretende até ser obtida uma resposta. Os argumentos do método são definidos como *goals* e que cada *goal* representa um parâmetro necessário à execução da tarefa e são constituídos por um id que representa o nome da variável e o seu valor. A lista de *goals* deve corresponder à estrutura do *input* definido no NSD para a tarefa em questão.

```

<?xml version="1.0"?>
<tdd xmlns="http://iramp-3.eu/nsd/tdd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://iramp-3.eu/nsd/tdd ./nsd-tdd-1.0.xsd">
  <tasks>
    <task id="1001-1234-5678-11222112" name="MakeWSN_withObstacles" cycles="1">
      <goals>
        <goal id="area_width">10000</goal>
        <goal id="area_height">6000</goal>
        <goal id="obstacles">100;100;250;300|300;430;100;100</goal>
        <goal id="sense_radius">1000</goal>
        <goal id="communication_radius">1000</goal>
      </goals>
    </task>
  </tasks>
</tdd>

```

Figura 4.20: Ficheiro Task Description Document

4.3.3.3 Quality Result Document

O Quality Result Document descreve o resultado obtido pela execução da tarefa definida no TDD. Pode afirmar-se que é o resultado positivo ao recebimento de um ficheiro TDD por uma entidade qualquer desde que este cumpra com os requisitos esperados. O ficheiro deve seguir a estrutura de dados definida para o *output* da tarefa que está a tratar e que está representada no NSD da entidade e só é construído se todos os requisitos forem cumpridos.

```

<?xml version="1.0"?>
<qrd xmlns="http://iramp-3.eu/nsd/grd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://iramp-3.eu/nsd/grd ./nsd-grd-1.0.xsd">
  <task id="1001-1234-5678-11222112" name="makeWSN" cycle="0">
    <result id="devices">[Libelium_1,Libelium_2,Libelium_3,Libelium_4,Libelium_5]</result>
    <result id="positions">[(100,100), (200,100), (300,100), (150,250), (250,250)]</result>
  </task>
</qrd>

```

Figura 4.21: Ficheiro Quality Result Document

Na imagem 4.21 podemos ver um exemplo de um Quality Result Document. O ficheiro está dividido em *tasks* que definem cada tarefa e são representadas por um identificador único e o nome do método que foi chamado. Cada *task* contém uma variável denominada *cycle* que indica o número de ciclos que ainda faltam até à conclusão da tarefa, quando a tarefa é concluída o ficheiro é construído com o resultado obtido da execução da tarefa que foi definida no TDD. O resultado é representado com um identificador próprio e o valor obtido pelos algoritmos implementados para o serviço em questão. Em 4.21 verificamos

uma resposta ao método *makeWsn* que tem como retorno dois *arrays*; um com os nomes de cada dispositivo e o outro com as posições de cada um sobre uma área fornecida.

4.3.3.4 Task Fullfilmente Document

O Task Fullfilmente Document representa um resultado negativo da execução de um TDD quando o serviço não obedece a determinados parâmetros específicos. Pode também ser usado para reportar a mal formação de um ficheiro TDD recebido e indicar o porque de não cumprir os requisitos esperados. Este tipo de ficheiro 4.22 é usado para informar o estado atual das configurações, os parâmetros e os valores que estão definidos no TDD. Se uma entidade qualquer verificar o TFD de uma outra entidade pode garantir a execução da tarefa representada se for capaz disso ou então é necessário esperar o envio de um novo TDD com uma nova configuração.

```
<?xml version="1.0"?>
<tfd xmlns="http://iramp-3.eu/nsd/tfd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://iramp-3.eu/nsd/tfd ./nsd-tfd-1.0.xsd">
  <tasks>
    <task id="1001-1234-5678-11222112" name="MakeWSN_withObstacles">
      <acknowledge type="remainingCycles" value="1" />
      <goals>
        <goal id="area_width">10000</goal>
        <goal id="area_height">6000</goal>
        <goal id="obstacles">100;100;250;300|300;430;100;100</goal>
        <goal id="sense_radius">1000</goal>
        <goal id="communication_radius">1000</goal>
      </goals>
    </task>
  </tasks>
</tfd>
```

Figura 4.22: Ficheiro Task Fullfilmente Document

4.4 Métodos, funções, output

Este projeto tem como objetivo disponibilizar um dispositivo virtual na rede que fornece quatro serviços distintos. Cada serviço possui uma série de métodos que podem ser acedidos por outras entidades através de UPnP e formular respostas adequadas ao tipo de pedido efetuado por estas. Dois serviços têm o propósito de apresentar soluções para uma distribuição ótima de dispositivos sobre uma determinada área, com ou sem obstáculos, construindo uma RSSF com garantias de uma boa qualidade de serviço, nomeadamente uma boa cobertura da área. É garantido ainda um serviço para comunicação DIL onde se efetuam trocas de ficheiros XML sob UPnP, permitindo assim a integração no projeto I-RAMP3 e um serviço que se foca na verificação de requisitos da rede com a possibilidade de recolha de informações úteis sobre a RSSF já implementada. Apresentamos aqui estes quatro serviços com mais pormenor.

4.4.1 Cobertura de área

Este serviço está dividido em cinco métodos e todos eles têm como retorno uma solução para a distribuição de dispositivos ao longo de uma área que não contempla obstáculos. Em todos eles são fornecidas as dimensões da área, comprimento e altura, e o retorno dá-se em forma de dois *arrays*, um com o nome dos dispositivos e outro com um conjunto de tuplos (x,y) que representam as posições cartesianas de cada dispositivo ao longo da área. Os cinco métodos existentes são:

- **MakeWsn:** Este é o método mais simples do serviço. Recebe apenas o comprimento e a altura da área de interesse e o valor dos raios do dispositivo, o raio de cobertura e o raio de conectividade, com que queremos contruir a RSSF. Todos estes valores devem ser fornecidos em centímetros e são declarados como do tipo float.
- **MakeWsnAccuracy:** Além de receber as dimensões da área onde se pretende instalar a rede, este método recebe como argumento o tipo de sensor pretendido e o valor da precisão que este deve fornecer. Ocorre então uma busca na rede por dispositivos com capacidade de fornecer o serviço com a precisão pretendida e calcula a solução mediante o valor do seu raio de cobertura e de conectividade.
- **MakeWsnDevice:** Recebe como argumento um identificador único de um dispositivo e o tipo de sensor que se pretende utilizar como por exemplo temperatura. Efetua a procura do dispositivo para recolher o valor do raio de conectividade e do raio de cobertura associado ao sensor e apresenta uma sugestão de configuração de uma RSSF.
- **MakeWsnDeviceList:** Este método recebe uma lista de identificadores únicos de

dispositivos, ou seja, uma lista de dispositivos e o tipo de sensor que se pretende usufruir. Como no método anterior ele apresenta uma solução para uma distribuição ótima de dispositivos sobre a área indicada mas tem a particularidade de informar se os dispositivos presentes na lista fornecida são suficientes para cobrir toda a área. Se o número de dispositivos contidos na lista não for suficiente para garantir uma cobertura ótima o método indica o número de dispositivos em falta ou então o número de dispositivos que estão a mais, se for o caso.

- **MakeWsnDeviceListAccuracy:** Este método apenas difere do anterior por incluir mais uma variável na sua avaliação. Além dos restantes argumentos é indicada a precisão requerida ao serviço e avaliados os sensores presentes na lista. O retorno é o mesmo que o método anterior, com a particularidade de fornecer o identificador dos dispositivos que não cumpram com os requisitos pretendidos.

4.4.2 Cobertura de área com Obstáculos

As entidades recorrem a este serviço apenas quando a área onde se pretende instalar a rede de sensores sem fios contém obstáculos. Em nada mais difere do serviço anterior a não ser a indicação das coordenadas e dimensões dos obstáculos em cada um dos métodos. A definição de obstáculos neste serviço segue no formato de uma *string* que contém valores do tipo *float* separados por ponto e vírgula. Cada obstáculo é definido por “x;y;comprimento;altura”, x e y representam as coordenadas cartesianas do obstáculo na área de interesse e o comprimento e altura as suas dimensões. É possível definir mais do que um obstáculo separando cada *string* com uma barra como por exemplo

x1; y1; comprimento1; altura1|x2; y2; comprimento2; altura2|x3; y3; comprimento3; altura3

Os métodos presentes neste serviço são:

- **MakeWsn_withObstacles;**
- **MakeWsnAccuracy_withObstacles;**
- **MakeWsnDevice_withObstacles;**
- **MakeWsnDeviceList_withObstacles;**
- **MakeWsnDeviceListAccuracy_withObstacles;**

4.4.3 Verificação de Requisitos

Ao contrário dos serviços anteriores que têm como retorno uma sugestão de uma RSSF e que retornam um conjunto de dispositivos com posições exatas, este serviço retorna

informações sobre os dispositivos de um sistema já implementado. A verificação dos requisitos exigidos a um sistema é importante para garantir que a rede se comporta como o esperado tal como, pode ser importante para questões de manutenção, saber o estado e informações detalhadas de todos os dispositivos. O serviço inclui quatro métodos:

- **GetAllSensors:** Devolve informações sobre todos os sensores presentes na rede. Retorna um *array* que contém um *array* por dispositivo que inclui; o identificador do dispositivo, o raio de conectividade, os serviços que ele dispõe; o raio de cobertura e a sua precisão.
- **GetSensorsByType:** Este método recebe como argumento o tipo de sensor que se deseja procurar na rede e devolve um *array* com o identificador único de cada dispositivo que preencha esse requisito.
- **VerifyServiceAccuracy:** Recebe como argumento o tipo de sensor e um certo valor de precisão. O método devolve um *array* com o identificador único de todos os dispositivos que forneçam o sensor e a precisão dados como argumento.
- **VerifyServiceAccuracyDeviceList:** Este método retorna um valor booleano. Recebe como argumento uma lista de identificadores únicos de dispositivos, um tipo de sensor e um valor que representa a precisão do sensor. Devolve *true* se todos os dispositivos contêm o tipo de sensor e a precisão dada ou *false* caso contrário.

4.4.4 Comunicação DIL

O Serviço que possibilita a comunicação via Device Integration Language(DIL) entre todas as entidades. Os atores conseguem comunicar através de ficheiros XML sob a tecnologia UPnP e para isso foram implementados 4 métodos:

- **GetNSD:** Devolve uma string que representa o ficheiro XML NetDev Service Description ou NSD.
- **GetQRD:** Devolve uma string que representa o último ficheiro XML NetDev Service Description ou QRD obtido.
- **GetTFD:** Devolve uma string que representa o último ficheiro XML Task Fullfilmente Document ou TFD obtido.
- **SubmitTDD:** Método que fornece a possibilidade de submeter um ficheiro XML Task Description Document ou TDD contendo uma tarefa representada que vai ser analisada e interpretada. Tem como retorno um ficheiro QRD se a tarefa for concluída com sucesso ou um TFD caso contrário.

Capítulo 5

Testes e Resultados

Os testes efetuados ao serviço visam explorar a distribuição de dispositivos sobre uma área, a comunicação UPnP e garantir uma comunicação via DIL garantindo uma boa integração na arquitetura da rede. Para a realização de todos os testes foi utilizada a ferramenta *Device Spy*, que foi especialmente desenvolvida pela empresa *Intel Corporation* para ajudar os desenvolvedores de *software* e de *hardware* a acelerar todo o processo de implementação e testes. Esta aplicação tem a capacidade de procurar e interagir com todos os dispositivos UPnP presentes na rede e é através do *Device Spy* que acedemos ao nosso serviço.

Utilizamos para simular netdevs, encapsulando motes, uma ferramenta de simulação que permite criar o número requerido de motes, contendo um número especificado de sensores, com características configuráveis de maneira a serem exportados para a rede. Foram criados quarenta dispositivos com diferentes tipos de serviço, diferentes tamanhos no que diz respeito ao raio de comunicação e monitorização. Recorde-se que já aqui foi mencionado que o serviço se encontra integrado na arquitetura *SmartComponent* que permite a procura e recolha de informação de todos os atores presentes na rede.

Para a bateria de testes efetuados foram utilizados dois computadores, um com um processador Intel Core i5 - 2.50GHz, com 6Gb de RAM e Windows 8.1 Pro 64-bit onde foi simulada uma rede baseada na arquitetura *SmartComponent*, com vários dispositivos registados e com o serviço aqui proposto instalado como um *bundle* e uma segunda máquina com processador Intel Core i5 - 2.3GHz, com 4Gb de RAM e Windows 7 Pro 64-bit que foi utilizada para aceder ao serviço e realizar pedidos de ações de maneira a testar todas as suas funcionalidades.

- **Descrição do Serviço:** O primeiro teste realizado ao serviço foi efetuar o pedido da descrição do mesmo através do método *GetNSD*. O resultado deste teste pode ser visualizado com mais pormenor no Apêndice B. Este método devolveu-nos uma *string* que representa um ficheiro XML contendo a descrição do serviço, o que é, o que faz e como faz, quais os métodos que tem disponíveis e quais os resultados dos mesmos.

Os métodos, ou ações UPnP que estão descritas neste documento são as mesmas que vamos utilizar para realizar os seguintes testes e assim verificar a comunicação entre o serviço e a própria rede, bem como testar a devolução de uma solução e a garantia de se poder comunicar através da linguagem já referida e explicada DIL.

- **Verificação de requisitos:** Para o segundo teste foi utilizada a ferramenta que nos ajuda a simular dispositivos e a exportá-los para a rede de forma a ficarem acessíveis ao nosso serviço via UPnP. O teste baseia-se na procura de todos os dispositivos que estão registados no sistema e para isso recorreu-se à ação *GetAllSensors*, definida no NSD e que nos devolve um *array* com as informações de todos os dispositivos. A informação recebida sobre cada dispositivo vem no formato pré-definido com o seguinte formato [Identificador da Mote:Raio de conectividade, Tipo de Sensor: raio de cobertura: precisão], ou seja;

```
[[uuid: c80294ee-5c8b-4aea-b043-95fc4b31aa484: radius:100, TemperatureService:
radius:150: accuracy:10, LuminosityService: radius:50: accuracy:2], ..., [uuid: e9041285-
7bff-445e-b63d-250e8694e2f0: radius:100, TemperatureService: radius:150: accu-
racy:10, LuminosityService: radius:50: accuracy:10]]
```

Para verificar os métodos que devolvem uma solução de rede de sensores sem fios foram escolhidos dispositivos que garantissem um teste sobre cada uma das abordagens resultantes da fase de planeamento que aqui definimos, para uma área sem obstáculos construída com mil centímetros (dez metros) de comprimento e setecentos centímetros (sete metros) de altura.

Como tal, escolhemos quatro tipos de dispositivos com diferentes raios de monitorização e de conectividade, sejam eles; (100,100); (150;100); (100;150); (200;100); Note-se que os valores são definidos em centímetros e os tuplos estão no formato (Raio de comunicação, Raio de monitorização). Os testes foram efetuados através da comunicação DIL com o envio de um ficheiro TDD para o serviço de modo a ser calculada uma solução para distribuição dos dispositivos propostos sobre a área definida. Seguem-se o resultado segundo as variáveis obtidas pelas condições da fase de planeamento:

- $R_s = R_c$: Esta solução apresentou as posições para **quarenta e um** dispositivos com raios de comunicação e monitorização iguais, **cem** centímetros. Como podemos verificar na Figura 5.1 a distribuição começa no dispositivo S0 seguindo para ESTE, sendo que na segunda linha inverte a direção. Podemos verificar também que alguns dispositivos foram adicionados na fronteira da área segundo a Regra F1, nomeadamente os dispositivos S11 e S32.

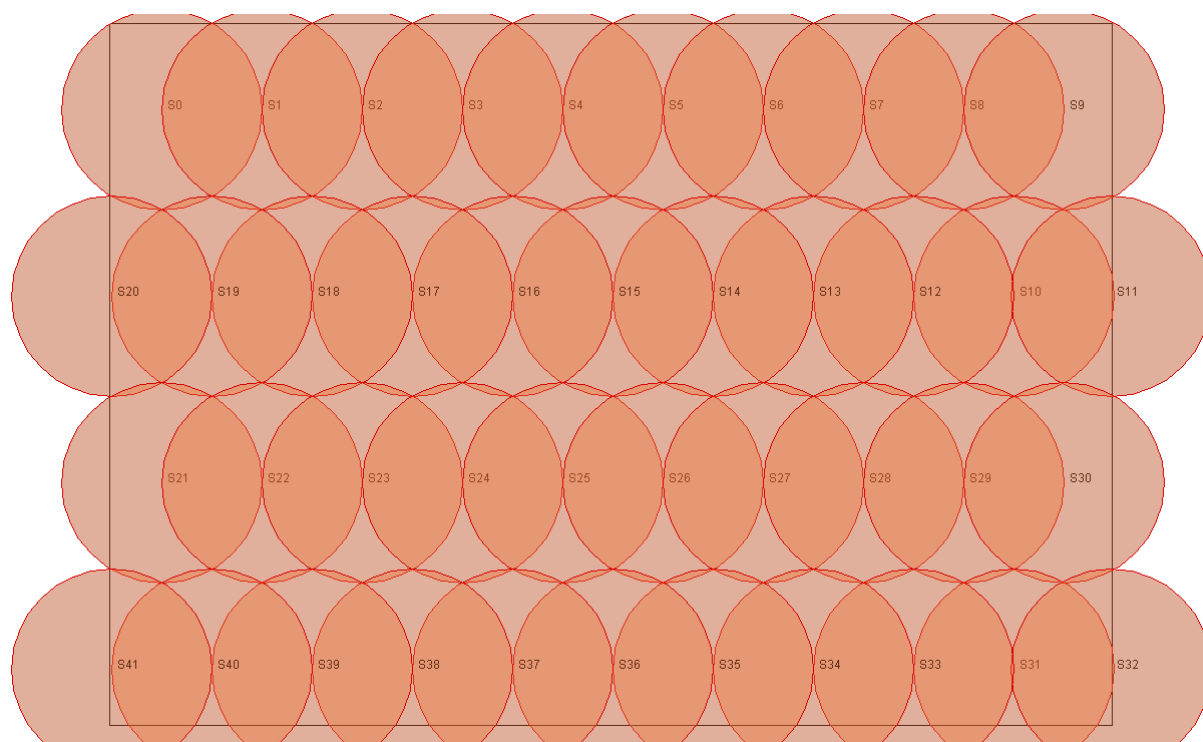


Figura 5.1: Distribuição quando $R_s = R_c$

- $R_s < R_c \leq \sqrt{3}R_s$: Na Figura 5.2 podemos observar o resultado obtido na execução deste teste onde cada dispositivo é representado por dois círculos, o vermelho refere o raio de monitorização do sensor e o azul o raio de conectividade da mote. A cobertura total foi alcançada com **trinta e seis** dispositivos e podemos verificar a execução das regras F1 e F2. O dispositivo S8 e S23 foram adicionados à fronteira lateral de modo a garantir total cobertura, bem como toda a ultima linha de dispositivos foi instalada sobre a fronteira inferior da área através da regra F2.

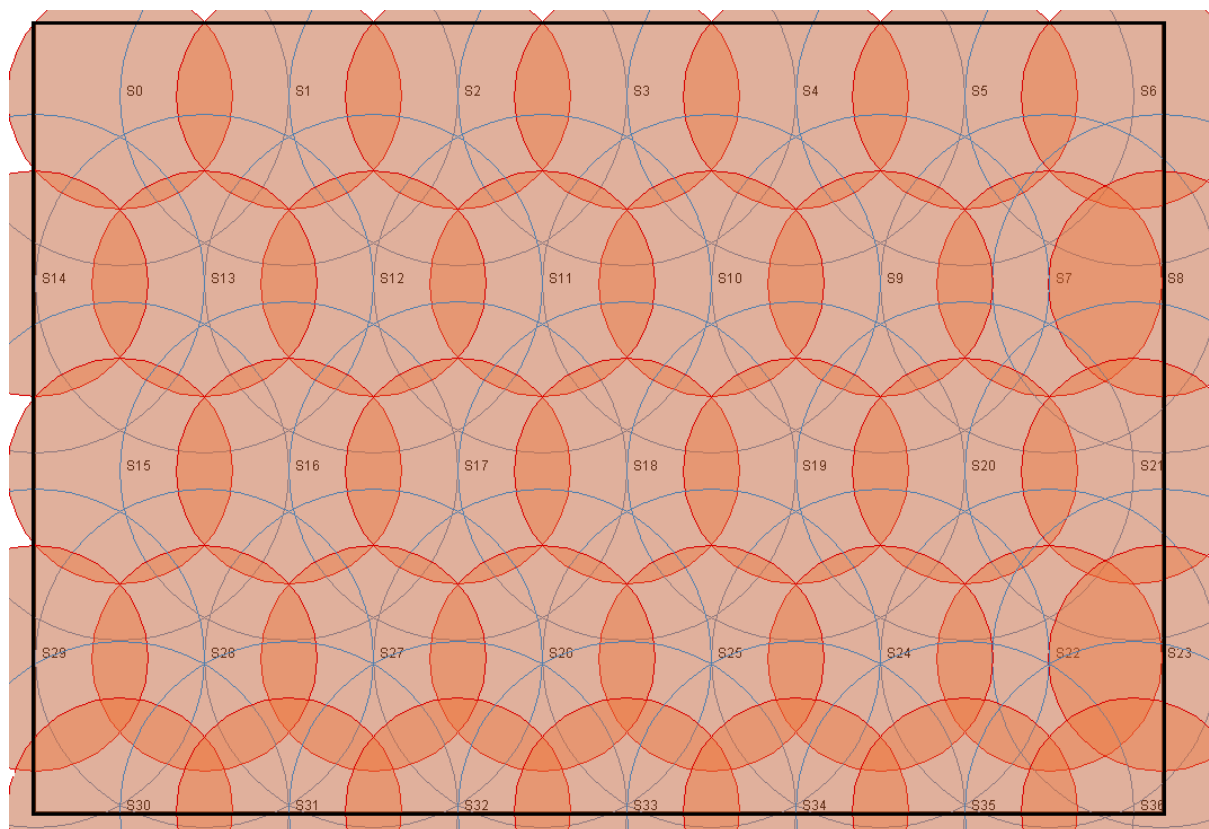


Figura 5.2: Distribuição quando $R_s < R_c \leq \sqrt{3}R_s$

- $R_c > \sqrt{3}R_s$: Esta distribuição alcança a cobertura máxima com menor número de dispositivos do que as distribuições até aqui testadas. O resultado foram as posições de **trinta e um** dispositivos como podemos verificar na Figura 5.3 Isso deve-se ao fato da mote possuir um raio de conectividade maior, garantindo assim uma boa conectividade da rede com a possibilidade de alcance de comunicação entre todos os vizinhos.

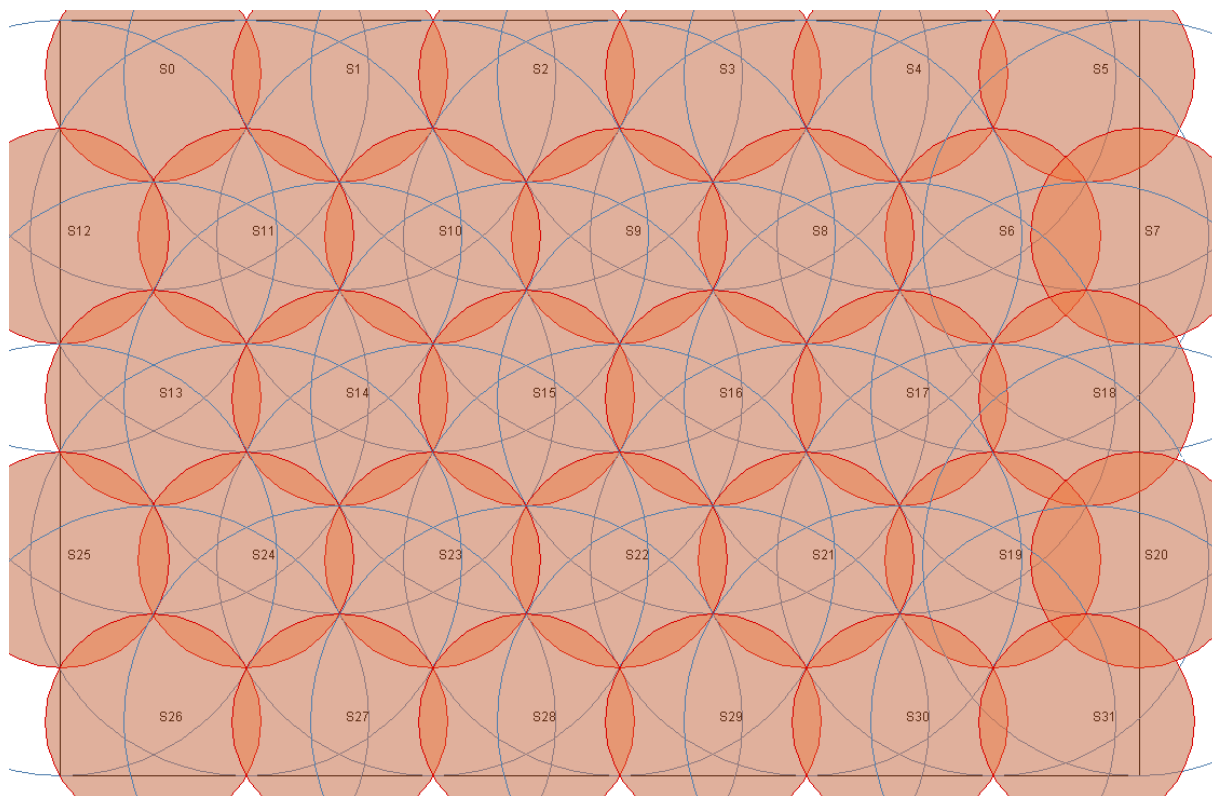


Figura 5.3: Distribuição quando $R_c > \sqrt{3}R_s$

- $R_s > R_c$: Esta foi a distribuição que conseguiu um melhor resultado em termos de número de dispositivos, contudo a conectividade só está assegurada em cada linha da grelha, Figura 5.4. Foram distribuídos **trinta** dispositivos, conseguindo uma cobertura máxima por parte da rede mas, poderá ser necessário adicionar dispositivos extra para que se obtenha uma conectividade máxima entre linhas.

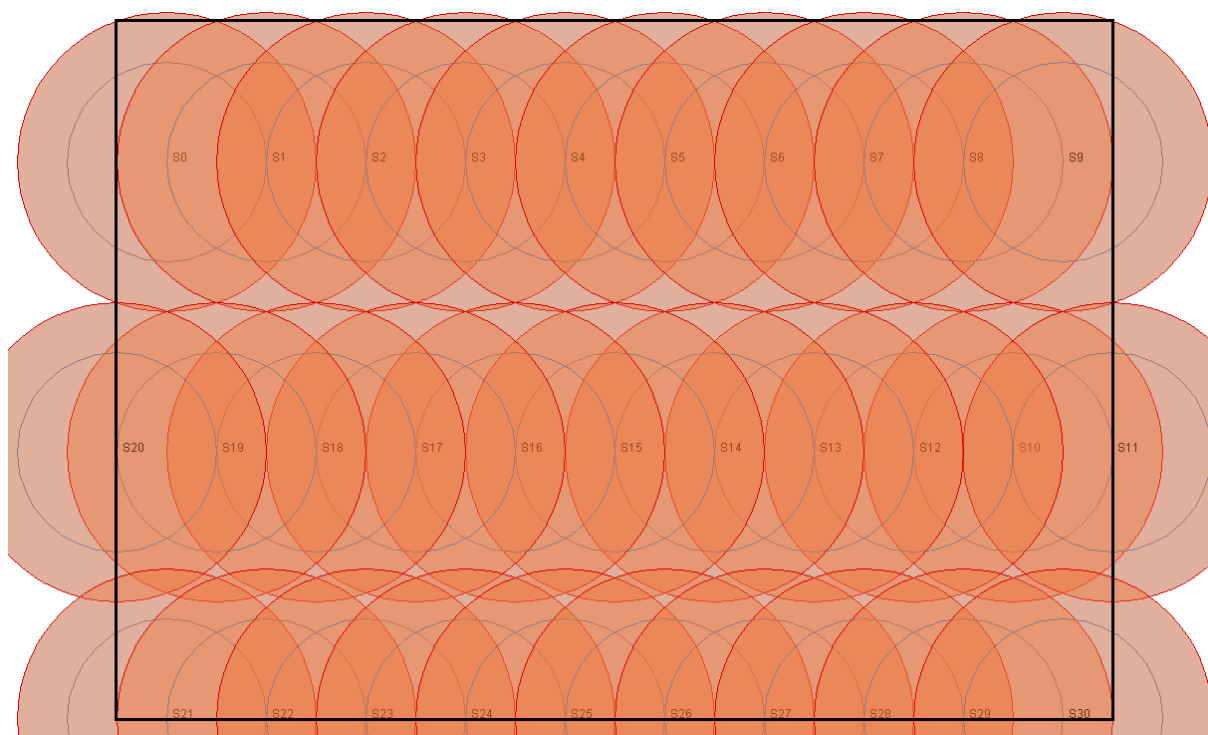


Figura 5.4: Distribuição quando $R_s > R_c$

Para a distribuição sobre obstáculos realizamos dois testes onde definimos obstáculos simples e compostos sobre uma área com as mesmas dimensões anteriores. Verificamos assim as regras definidas para lidar com os problemas de fronteira.

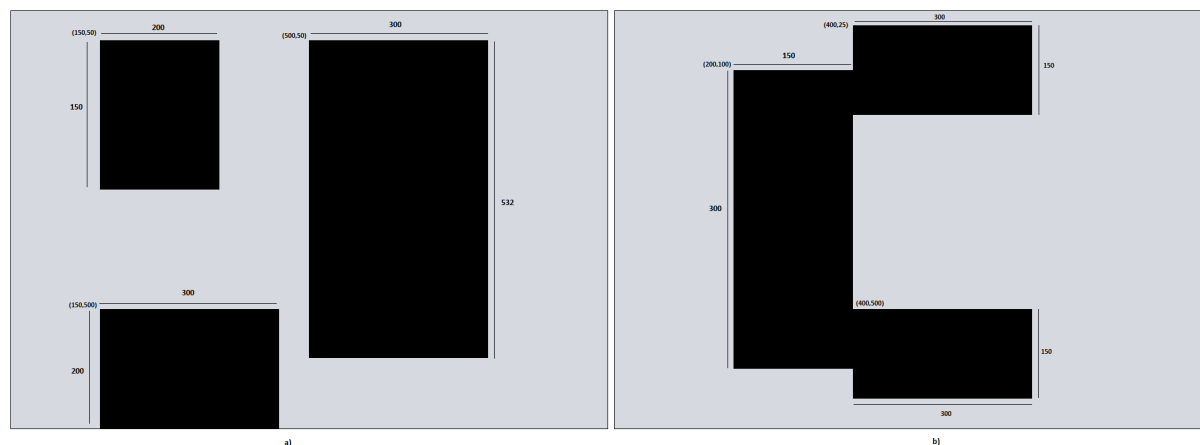


Figura 5.5: Distribuição sobre obstáculos

Os obstáculos foram definidos com as dimensões e nas posições definidas na Figura 5.5 sobre a mesma área definida anteriormente. Foi definido um dispositivo com raio de monitorização e comunicação iguais a *cem* centímetros e verificou-se as seguintes distribuições:

- **Distribuição sobre obstáculos:** Podemos facilmente observar através das Figuras, 5.6 e 5.7 o algoritmo a contornar todos os obstáculos que encontra pelo caminho segundo as Regras F1 e F2. Nas imagens, o algarismo presente no nome de cada dispositivo segue a ordem de posicionamento na área, para facilmente se perceber a direção do algoritmo em cada ponto. Observa-se o uso das prioridades definidas para cada movimento, de forma a atuar da melhor maneira quando encontra um obstáculo ou a fronteira da área.

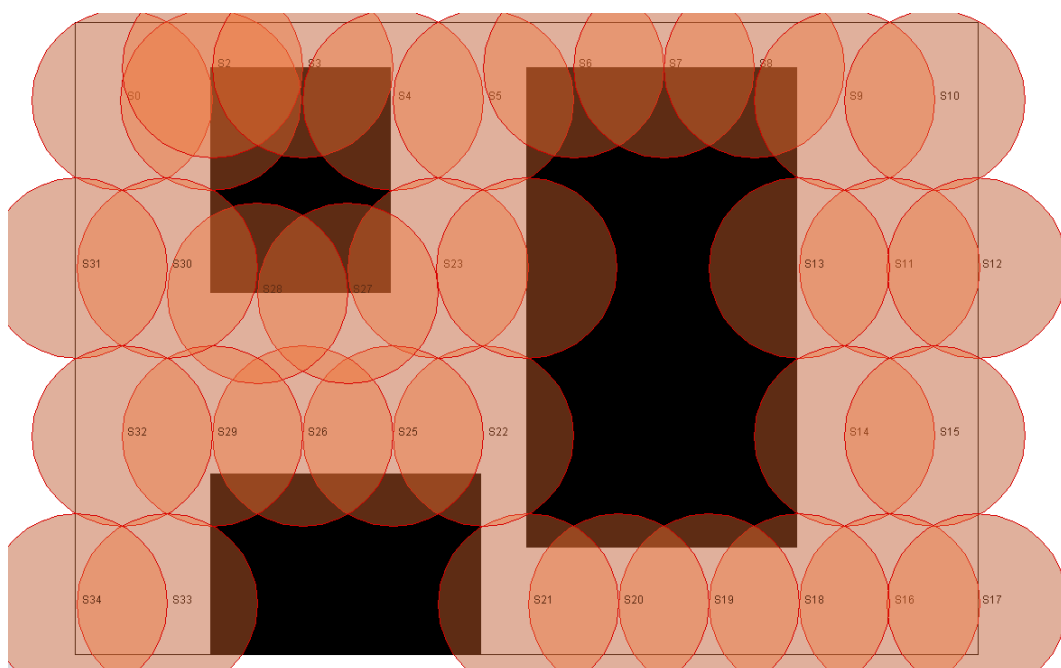


Figura 5.6: Distribuição sobre obstáculos simples

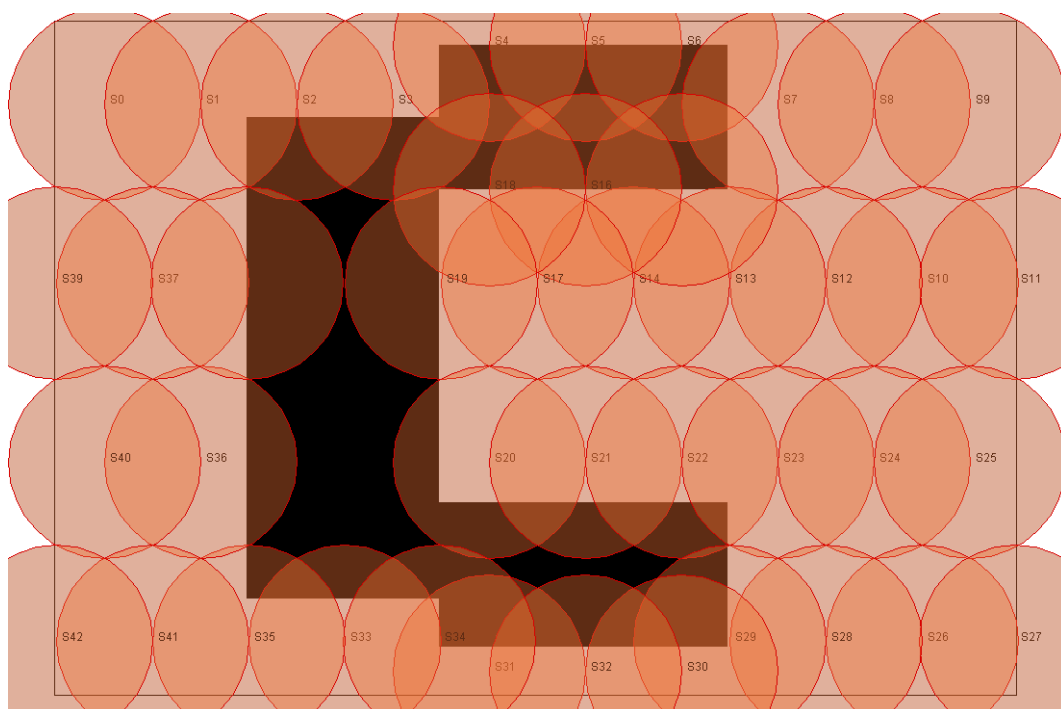


Figura 5.7: Distribuição sobre obstáculo composto

Capítulo 6

Conclusões e trabalho futuro

No setor industrial, as redes de sensores sem fios são utilizadas para diversos tipos de função, seja na monitorização de células individuais presentes no *shop-floor*, para garantir que certos parâmetros alcancem valores específicos, monitorização do produto resultante das máquinas de maneira a se certificar que cumpre todos os requisitos definidos para o produto final ou mesmo na monitorização de salas e edifícios com determinado fim específico, como por exemplo recolher a medida de luminosidade ou temperatura de modo que seja calibrada uma câmara ou um espaço que requeira um certo valor de luminosidade específico.

Pelo fato de existirem bastantes abordagens para este tipo de rede, não existe um tipo específico de sensor que seja usado no setor industrial, ou outro qualquer, pois o tipo de sensor e mote a utilizar está dependente do processo a utilizar e das funções esperadas para a aplicação. Contudo, como estamos focados na Indústria, em fábricas e armazéns, a possibilidade de instalar uma RSSF de forma manual e normalmente em posições pré-definidas serviu de parâmetro à escolha deste tipo de algoritmo. Normalmente a cobertura e conectividade de uma rede é estudada para áreas onde os dispositivos são distribuídos aleatoriamente e/ou onde se sabe a posição de cada mote antes de se verificar e planejar a rede. Neste projeto assume-se que se conhece a área a monitorar mas não a localização de cada dispositivos e a sugestão de posições específicas para cada mote numa determinada área, com o intuito de se garantir uma cobertura eficiente e assegurar conectividade da rede foi conseguida com recurso a algoritmos que fornecem uma sugestão de distribuição de uma RSSF.

Tendo em conta os testes realizados e o estudo aqui apresentado, de maneira a considerar a perda de precisão que se dá com o aumento de distância de um ponto a monitorizar em relação ao dispositivo, podemos diminuir os raios de conectividade e monitorização na definição de cada *NetDev*, garantindo uma rede de sensores sem fios mais forte.

O serviço aqui proposto mostrou-se capaz de uma boa resposta, tanto ao problema da

integração num sistema, com a ajuda da arquitetura SmartComponent, como ao problema da cobertura máxima de uma rede de sensores sem fios numa área específica com um número reduzido de dispositivos. Fica assim facilitada a tarefa de planeamento e administração de toda a RSSF, um dos pontos importantes no que toca ao RMS. Através dos testes efetuados verificou-se a funcionalidade da troca de ficheiros XML de modo a obter informação sobre a rede e todos os atores presentes. A recolha de dados importantes para o planeamento de uma RSSF como para um gestor de redes que tem como objetivos administrar uma grande quantidade de dispositivos, como a precisão de cada serviço, o raio de alcance quer de cobertura quer de conectividade são disso o exemplo. Pelos testes realizados, quando comparados os quatro métodos de distribuição resultantes da fase de planeamento do algoritmo, verificamos que a distribuição requer um maior número de dispositivos quando os dispositivos possuem os dois raios iguais. Em contra partida, obteve-se um melhor resultado quando $R_s > R_c$ muito se deve ao fato de possuir dispositivos com um raio de monitorização maior do que os restantes testes, contudo, neste caso verifica-se a necessidade de se adicionar dispositivos extra de modo a assegurar uma conectividade máxima e por isso, parece-nos mais eficiente uma distribuição de dispositivos com $R_c > \sqrt{3}R_s$ que alcança a cobertura máxima apenas com um dispositivo a mais que o teste referido anteriormente e assegura uma forte conectividade da rede. Será uma informação importante a ter em conta na escolha de dispositivos para o sistema que se deseja implementar.

Mostrou-se também a possibilidade da definição de obstáculos, simples e compostos e verificou-se como se comporta o algoritmo no caso da existência de alguns obstáculos no meio, que de algum modo podem interferir com o bom funcionamento da rede.

No futuro, este serviço pode ser útil à formação de grupos de nós automáticos de acordo com o seu posicionamento e a sua função, assim como as suas características físicas, a comunicação direta com nós móveis de maneira a se posicionarem nos locais corretos de modo a garantir que são cumpridos todos os requisitos do sistema ou mesmo a comunicação direta com um robot que procederá à implantação dos dispositivos ao longo da área a monitorizar. Devido à complexidade do projeto ainda há alguns ajustes que se continuam a fazer com o objetivo de tornar o produto mais eficaz. A criação de uma interface gráfica para uma melhor visualização de toda a rede, com possibilidade de alterar a posição dos dispositivos e formular novas soluções de forma rápida e intuitiva ajustando a rede da maneira que melhor se adapte ao tipo de aplicação.

Uma vez que este é um produto abrangente a uma grande diversidade de entidades e aplicações, é possível que sejam efetuadas alterações para obter uma melhor adaptação às circunstâncias de cada cliente e objetivo. A criação de diferentes algoritmos de distribuição com a possibilidade de uma fácil integração no serviço graças à tecnologia OSGI, de forma a se adaptar melhor o serviço a cada aplicação.

Apêndice A

Acrónimos

RSSF	Rede de Sensores Sem Fios
DIL	Device Integration Language
NSD	NetDev Service Description
TDD	Task Description Document
TFD	Task Fullfilmente Document
QRD	Quality Result Document
OSGI	Open Service Gateway Initiative
UPnP	Universal Plug and Play
I-RAMP3	Intelligent Reconfigurable Machines for Smart Plug&Produce Production
SeISus	Health Monitoring and Life-Long Capability Management for SELf- SUSTaining Manufacturi
RMS	Reconfigurable Manufacturing System
XML	eXtensible Markup Language
JAXB	Java Architecture for XML Binding
LSTS	Laboratório de Sistemas e Tecnologias Subaquáticas
RAM	Random Access Memory
ROM	Read-Only Memory
ADC	Analogue-to-Digital Converter
DSSS	Direct Sequence Spread Spectrum
OFDM	Orthogonal Frequency-Division Multiplexing
AES	Advanced Encryption Standard
ISA	International Society of Automation
AFH	Adaptative Frequency Hopping
ISM	Industrial, Scientific and Medical
TDMA	Time Division Multiple Access
CSMA/CD	Carrier Sense Multiple Access with Collision Detection

Apêndice B

Ficheiro NSD

```
<?xml version="1.0"?>
<nsd xmlns="http://fe.up.pt/nsd" xmlns:upnp="http://fe.up.pt/nsd/dataConnectors/upnp"
xmlns:bus="http://fe.up.pt/nsd/dataConnectors/fieldbus" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://fe.up.pt/nsd.../nsd-base-1.0.xsd http://fe.up.pt/nsd/dataConnectors/upnp...
upnp-connectors.xsd">

<description>
<name>Area Coverage And Wsn Requeriments</name>
<netdevType>/netdev/virtual/area/coverage</netdevType>
<power unit="kW">999</power>
<model>V1.0</model>
<manufacturer>FCUP + FEUP</manufacturer>
<serial>FCUP20132014FEUP</serial>
<built>2014-01-30T09:00:00</built>
<physicalProperties />
</description>

<!-- The task range contains a list of task definitions in NSD, which can be executed by netdev -->
<taskRange>

<taskDefinition name="makeWSN" sntd="/task/netdev/virtual/makeWSN">
<dataDefinitions>

<!-- Input -->
<dataDefinition id="area_width">
<metaData name="datatype">float</metaData>
<metaData name="minValue">0</metaData>
<metaData name="maxValue">1000000</metaData>
<metaData name="unit">cm</metaData>
</dataDefinition>
<dataDefinition id="area_height">
<metaData name="datatype">float</metaData>
<metaData name="minValue">0</metaData>
<metaData name="maxValue">1000000</metaData>
<metaData name="unit">cm</metaData>
</dataDefinition>
<dataDefinition id="sense_radius">
<metaData name="datatype">float</metaData>
```

```

    <metaData name="minValue">0</metaData>
    <metaData name="maxValue">1000000</metaData>
    <metaData name="unit">cm</metaData>
  </dataDefinition>
  <dataDefinition id="communication_radius">
    <metaData name="datatype">float</metaData>
    <metaData name="minValue">0</metaData>
    <metaData name="maxValue">1000000</metaData>
    <metaData name="unit">cm</metaData>
  </dataDefinition>

  <!-- Output -->
  <dataDefinition id="position">

  </dataDefinition>
  <dataDefinition id="device">

  </dataDefinition>
  <dataDefinition id="devices">
    <metaData name="datatype">device</metaData>
    <metaData name="unit">array</metaData>
  </dataDefinition>
  <dataDefinition id="positions">
    <metaData name="datatype">position</metaData>
    <metaData name="unit">array</metaData>
  </dataDefinition>
</dataDefinitions>

<tddSchema>
  <goals>
    <goal dataDefinitionRef="area_width" />
    <goal dataDefinitionRef="area_height" />
    <goal dataDefinitionRef="sense_radius" />
    <goal dataDefinitionRef="communication_radius" />
  </goals>
  <boundaryConditions> </boundaryConditions>
</tddSchema>

<qrdSchema>
  <result dataDefinitionRef="devices" />
  <result dataDefinitionRef="positions" />
</qrdSchema>

</taskDefinition>

<taskDefinition name="makeWSN_Device" sntd="/task/netdev/virtual/makeWSN_Device">
</taskDefinition>

<taskDefinition name="makeWSN_DeviceList" sntd="/task/netdev/virtual/makeWSN_DeviceList">
</taskDefinition>

<taskDefinition name="makeWSN_Accuracy" sntd="/task/netdev/virtual/makeWSN_Accuracy">
</taskDefinition>

<taskDefinition name="makeWSN_DeviceList_Accuracy"
  sntd="/task/netdev/virtual/makeWSN_DeviceList_Accuracy">

```

```

</taskDefinition>

<taskDefinition name="makeWSN_withObstacles" sntd="/task/netdev/virtual/makeWSN">
</taskDefinition>

<taskDefinition name="makeWSN_Device_withObstacles" sntd="/task/netdev/virtual/makeWSN_Device">
</taskDefinition>

<taskDefinition name="makeWSN_DeviceList_withObstacles" sntd="/task/netdev/virtual/makeWSN_DeviceList">
</taskDefinition>

<taskDefinition name="makeWSN_Accuracy_withObstacles" sntd="/task/netdev/virtual/makeWSN_Accuracy">
</taskDefinition>

<taskDefinition name="makeWSN_DeviceList_Accuracy_withObstacles"
sntd="/task/netdev/virtual/makeWSN_DeviceList_Accuracy">
</taskDefinition>

<taskDefinition name="get_All_Sensors" sntd="/task/netdev/virtual/getWSN">
</taskDefinition>

<taskDefinition name="get_Sensors_by_Type" sntd="/task/netdev/virtual/getWSN">
</taskDefinition>

<taskDefinition name="verifyServiceAccuracy" sntd="/task/netdev/virtual/verifyServiceAccuracy">
</taskDefinition>

<taskDefinition name="verifyServiceAccuracy_DeviceList"
sntd="/task/netdev/virtual/verifyServiceAccuracy_DeviceList">
</taskDefinition>

<taskDefinition name="verifyServiceAccuracyDeviceList" sntd="/task/netdev/virtual/verifyServiceAccuracy">
</taskDefinition>

<taskDefinition name="verifyServiceAccuracy_DeviceList"
sntd="/task/netdev/virtual/verifyServiceAccuracy_DeviceList">
</taskDefinition>

</taskRange>
</nsd>

```

Referências

- [1] L. B. Ruiz, J. M. Nogueira, and A. A. Loureiro, “Manna: A management architecture for wireless sensor networks,” pp. 116–125, 2003.
- [2] C. Zhu, C. Zheng, L. Shu, and G. Han, “A survey on coverage and connectivity issues in wireless sensor networks,” *Journal of Network and Computer Applications*, vol. 35, no. 2, pp. 619 – 632, 2012, simulation and Testbeds. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1084804511002323>
- [3] S. F. P. F. of the European Commission, “I- ramp3 - intelligent reconfigurable machines for smart plug&produce production,” <http://www.i-ramp3.eu/>, Consultado em Janeiro de 2014.
- [4] —, “Selsus - health monitoring and life-long capability management for self-sustaining manufacturing systems,” <http://www.selsus.eu/>, Consultado em Janeiro de 2014.
- [5] M. van de Panne and E. Fiume, “Sensor-actuator networks,” New York, NY, USA, pp. 335–342, 1993. [Online]. Available: <http://doi.acm.org/10.1145/166117.166159>
- [6] A. A. Loureiro, J. M. S. Nogueira, L. B. Ruiz, R. A. de Freitas Mini, E. F. Nakamura, and C. M. S. Figueiredo, “Redes de sensores sem fio,” pp. 179–226, 2003.
- [7] B. Wang, “Coverage control in sensor networks,” 2010.
- [8] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister, “System architecture directions for networked sensors,” ACM, pp. 93–104, 2000.
- [9] T. Alliance, “Tinyos,” <http://www.tinyos.net/>, Consultado em Janeiro de 2014.
- [10] A. Ghosh and S. K. Das, “Coverage and connectivity issues in wireless sensor networks: A survey,” pp. 303–334, 2008.
- [11] T. Camilo, A. Rodrigues, S. Silva, and F. Boavida, “Redes de sensores sem fios, considerações sobre a sua instalação em ambiente real,” in *Wireless Sensor Networks—some Considerations on Deployment in Real Environments*, CSMU2006-

- Conferência sobre Sistemas Móveis e Ubíquos, Guimarães, (Portugal) (June 2006) (in Portuguese)*, 2006.
- [12] T. Wark, D. Swain, C. Crossman, P. Valencia, G. Bishop-Hurley, and R. Handcock, "Sensor and actuator networks: protecting environmentally sensitive areas," *Pervasive Computing, IEEE*, vol. 8, no. 1, pp. 30–36, 2009.
- [13] Y. Zou and K. Chakrabarty, "A distributed coverage- and connectivity-centric technique for selecting active nodes in wireless sensor networks," *Computers, IEEE Transactions on*, vol. 54, no. 8, pp. 978–991, Aug 2005.
- [14] —, "Sensor deployment and target localization in distributed sensor networks," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 3, no. 1, pp. 61–91, 2004.
- [15] A. Hossain, P. Biswas, and S. Chakrabarti, "Sensing models and its impact on network coverage in wireless sensor network," *IEEE*, pp. 1–5, 2008.
- [16] A. Howard, M. J. Matarić, and G. S. Sukhatme, "An incremental self-deployment algorithm for mobile sensor networks," *Autonomous Robots*, vol. 13, no. 2, pp. 113–126, 2002.
- [17] G. Wang, G. Cao, T. La Porta, and W. Zhang, "Sensor relocation in mobile sensor networks," in *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, vol. 4. IEEE, 2005, pp. 2302–2312.
- [18] J. Li, L. L. Andrew, C. H. Foh, M. Zukerman, and H.-H. Chen, "Connectivity, coverage and placement in wireless sensor networks," *Sensors*, vol. 9, no. 10, pp. 7664–7693, 2009.
- [19] R. Mulligan and H. M. Ammari, "Coverage in wireless sensor networks: A survey," 2010.
- [20] C.-Y. Chang, C.-Y. Chang, Y.-C. Chen, and H.-R. Chang, "Obstacle-resistant deployment algorithms for wireless sensor networks," *Vehicular Technology, IEEE Transactions on*, vol. 58, no. 6, pp. 2925–2941, July 2009.
- [21] M. Monica and A. K. Sharma, "Comparative study of energy consumption for wireless sensor networks based on random and grid deployment strategies," pp. 28–35, 2010.
- [22] P. S. Hepsibha and G. S. Rao, "Comparative analysis of area coverage in wsns using various grid-based node deployment schemes," 2013'.

- [23] M. M. Iqbal, I. Gondal, and L. Dooley, “Dynamic symmetrical topology models for pervasive sensor networks,” in *Multitopic Conference, 2004. Proceedings of INMIC 2004. 8th International*. IEEE, 2004, pp. 466–472.
- [24] S. Shakkottai, R. Srikant, and N. B. Shroff, “Unreliable sensor grids: Coverage, connectivity and diameter,” *Ad Hoc Networks*, vol. 3, no. 6, pp. 702–716, 2005.
- [25] T. Clouqueur, V. Phipatanasuphorn, P. Ramanathan, and K. K. Saluja, “Sensor deployment strategy for target detection,” in *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*. ACM, 2002, pp. 42–48.
- [26] S. S. Dhillon, K. Chakrabarty, and S. Iyengar, “Sensor placement for grid coverage under imprecise detections,” in *Information Fusion, 2002. Proceedings of the Fifth International Conference on*, vol. 2. IEEE, 2002, pp. 1581–1587.
- [27] I. Chatzigiannakis, G. Mylonas, and S. Nikolettseas, “Modeling and evaluation of the effect of obstacles on the performance of wireless sensor networks,” in *Proceedings of the 39th annual Symposium on Simulation*. IEEE Computer Society, 2006, pp. 50–60.
- [28] Y.-C. Wang, C.-C. Hu, and Y.-C. Tseng, “Efficient deployment algorithms for ensuring coverage and connectivity of wireless sensor networks,” in *Wireless Internet, 2005. Proceedings. First International Conference on*, July 2005, pp. 114–121.
- [29] K. Kar, S. Banerjee *et al.*, “Node placement for connected coverage in sensor networks,” in *WiOpt’03: Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks*, 2003.
- [30] M. Li, P.-J. Wan, and O. Frieder, “Coverage in wireless ad hoc sensor networks,” *Computers, IEEE Transactions on*, vol. 52, no. 6, pp. 753–763, 2003.
- [31] Z. Bi, S. Lang, W. Shen, and L. Wang, “Reconfigurable manufacturing systems: the state of the art,” *International Journal of Production Research*, vol. 46, no. 4, pp. 967–992, 2008.
- [32] L. Neto, “Reconfigurable smartcomponent system,” 2014.