UNIVERSIDADE DA BEIRA INTERIOR
Engenharia

Convergência de Redes Sem Fios para Comunicações M2M e Internet das Coisas em Ambientes Inteligentes

Paulo Renato Neves Torres Gouveia

Dissertação para obtenção do Grau de Mestre em
Engenharia Electrotécnica e de Computadores
(2º ciclo de estudos)

Orientador: Prof. Doutor Fernando José da Silva Velez

Covilhã, outubro de 2013

"No futuro, os computadores poderão pesar menos do que 1.5 toneladas"
Revista *Popular Mechanics*, 1949

"Ele não sabia que era impossível. Ele foi lá e fez"
Jean Cocteau

Agradecimentos

Apesar de toda a dedicação, esforço e imenso trabalho de investigação realizado para tornar possível o desenvolvimento desta dissertação de mestrado, esta contou com importantes apoios e incentivos de várias pessoas que sem as quais não se teria tornado uma realidade. Por isso, quero deixar aqui o meu profundo agradecimento a todos os que acreditaram em mim e me apoiaram:

Ao Professor Fernando José da Silva Velez, meu orientador, pela cooperação, motivação, sugestões de ideias e por me ter dado a oportunidade de integrar no grupo de investigação do Instituto de Telecomunicações da Covilhã.

Ao grupo de investigação do Instituto de Telecomunicações da Covilhã, especialmente ao Norberto Barroca, Luís Borges e Henrique Saraiva, pela amizade, partilha de conhecimentos, sugestões e tempo despendido, sem os quais esta dissertação não teria sido possível.

À bolsa de iniciação científica, "Consumo de Energia em Redes de Comunicação sem Fios em Ambientes Inteligentes: Monitorização e Optimização", atribuída pelo Instituto de Telecomunicações no âmbito do PEst-OE/EEI/LAooo8/2011, de 1 de Junho de 2012 a 31 de Dezembro de 2012.

Ao Instituto de Telecomunicações e à Universidade da Beira Interior por terem disponibilizado todos os recursos que tornaram esta dissertação possível. Em particular, quero agradecer à Drª Sara Correia.

Aos projectos, onde estive envolvido: PEst-OE/EEI/LAooo8/2013, PLANOPTI, PROENERGY-WSN, COST IC1004, COST IC0905 e CREaTION.

Á Susana Palinhas e à sua família pelo apoio, incentivo e carinho neste momento tão importante da minha vida, e aos quais estarei para sempre grato.

E acima de tudo, a toda a minha família, especialmente aos meus pais e avós, por todo o amor, suporte, compreensão, segurança e oportunidade que, incondicionalmente, sempre me deram para eu poder percorrer este caminho, ultrapassar os momentos mais difíceis da minha vida e continuar a conquistar os meus sonhos.

Resumo

Actualmente, os Ambientes Inteligentes (AmI) estão a emergir através da convergência entre as redes de comunicação sem fios, a microelectrónica e a Internet. Este tipo de aplicações cria novas perspectivas no âmbito da partilha de informação dentro das sociedades humanas, ao mesmo tempo que surgem ideias inovadoras envolvendo espaços, objectos e entidades físicas com as quais lidamos diariamente. Com os AmI e as redes de comunicação, a partilha de informação passa a englobar coisas físicas que geram e disponibilizam dados sobre si mesmas. Nesta dissertação foram enquadradas as tecnologias de informação e a convergência entre redes de comunicação sem fios heterogéneas que suportam as aplicações dos AmI. Daqui surgem dois conceitos extremamente relevantes: a *Internet of Things* (IoT) e as comunicações *Machine-to-Machine* (M2M). A IoT, além de fornecedora de serviços virtuais, integra objectos físicos com representação virtual, interligados em rede para partilharem informação sobre vários factores. As comunicações M2M resultam da convergência de redes heterogéneas que permitem a comunicação directa entre dispositivos e objectos sem qualquer intervenção humana. A IoT é no fundo a base principal para a interligação das várias redes M2M entre objectos. As comunicações M2M são formadas por dois tipos de sub-redes: redes capilares e redes celulares. As redes capilares são compostas pelos dispositivos e os objectos embutidos nos ambientes inteligentes, que geram e difundem dados. As redes celulares são a espinha dorsal para a partilha destes dados através da Internet e centros de dados. As redes de sensores sem fios (RSSF), em conjunto com a norma IEEE 802.15.4, foram abordadas nesta dissertação para integrarem as redes capilares. As RSSF são redes com ritmos de transmissão e potência reduzida, que possibilitam o desenvolvimento de uma vasta variedade de aplicações AmI. Foi elaborado um estudo sobre a eficiência energética dos dispositivos RSSF disponíveis no mercado. A interligação das RSSF com a Internet é possível através da atribuição de endereços IPv6 aos dispositivos das RSSF. A camada de adaptação 6LoWPAN possibilita a atribuição de endereços IPv6 com *overheads* bastante reduzidos. Para permitir a máxima eficiência das RSSF, integradas com a IoT, foi adoptado o protocolo de encaminhamento RPL, desenvolvido no âmbito das redes 6LoWPAN. A contribuição principal desta dissertação centra-se na identificação e resolução de problemas ao nível do encaminhamento nas RSSFs, e no estudo de algumas métricas utilizadas para calcular o custo dos encaminhamentos entre nós. O protocolo RPL foi implementado no simulador OMNeT++ com o objectivo de analisar os resultados e os comportamentos das métricas de custo ETX, HOP-COUNT e RSSI. O ETX atingiu os melhores resultados para o débito binário útil, entre 75 % e 95 %, enquanto o HOP-COUNT tem resultados abaixo do 50 %. Contudo, o HOP-COUNT tem tempos de convergência superiores e latências inferiores. Com o RSSI obtêm-se resultados modestos e provou-se ser esta uma métrica de custo bastante precisa e equilibrada em todos os resultados.

Palavras-chave

Internet of Things, M2M, redes de sensores sem fios, 6LoWPAN, RPL, métricas de custo

Abstract

Nowadays, Ambient Intelligent (AmI) applications are emerging from the convergence between wireless networks, microelectronics, and Internet. The development of this kind of applications creates new perspectives on information sharing inside human societies, while new innovative ideas arise, involving spaces, objects and physical entities, which we deal with every day. With AmI and communication networks, information sharing is no longer only between and for the people, to encompass physical things that generate and provide data and information of themselves. One of the objectives of this thesis is to address the information technologies as well as the convergence between heterogeneous wireless networks that support the AmI applications. In this context two extremely important concepts arise: the Internet of Things (IoT) and the Machine-to-Machine (M2M) communications. The IoT is, beyond a virtual service provider, the integration of physical objects with virtual representation, networked together to share information about various factores of the surrounding environment. In turn, M2M communications result from the convergence of heterogeneous networks that allow direct communication between devices and objects, with no human intervention. The IoT is basically the main base for the interconnection between objects of M2M networks. M2M communications are composed by two types of sub-networks: capillary and cellular networks. Capillary networks are composed of devices and objects that are embedded in the AmI, while generate and disseminate relevant data about themselves. Cellular networks participate, as the necessary backbone, to share this data over the Internet and data centers. Wireless Sensor Networks (WSN) ate low rate and low power networks that enable the development of a wide variety of AmI applications. In this thesis, the adoption of WSN and IEEE 802.15.4 standard for the capillary networks were assumed. A study was conducted on the energy efficiency and lifetime of the WSN devices available on the market. The interconnection of WSNs with the Internet is possible by assigning IPv6 addresses to low-power devices. The 6LoWPAN adaption layer enables the IPv6 addresses assignment with low levels of overhead. To facilitate maximum efficiency of WSN, integrated with IoT, the routing protocol RPL was adopted, developed to be compatible with 6LoWPAN networks. The main contribution of this thesis is on identifying and solving problems at the level of routing in WSN, and a study of specific metrics used to calculate the forwarding cost between nodes. The RPL protocol was implemented on OMNeT++ simulator with the objective of analyze the results and behaviors of ETX, HOP-COUNT and RSSI routing metrics. ETX achieved the highest goodput results, between 75 % and 95 %, while HOP-COUNT has results above 50 %. However, HOP-COUNT has the fastest convergence times and the shortest latencies. In turn, RSSI has modest results but proved to be very accurate and a balanced metric for every set of results.

Keywords

Internet of Things, M2M, wireless sensor networks, 6LoWPAN, RPL, cost metrics

Índice

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

1.1 Introdução aos Ambientes e Redes Inteligentes

A revolução microelectrónica iniciou-se em 1948 com a invenção do transístor. Considerado o componente electrónico mais importante do século XX, revolucionou o funcionamento e a construção dos dispositivos electrónicos, substituindo as válvulas a vácuo de três polos e despoletando os processos de miniaturização dos equipamentos nos últimos 60 anos. A produção da electrónica traduzir-se-ia em três características fundamentais: dispositivos com tamanhos cada vez mais reduzidos, mais velozes e com menores custos. A partir destes princípios, Gordon Moore observou e previu que o número de transístores dentro de um determinado espaço limitado dobraria de 18 em 18 meses (de 24 em 24, segundo alguns investigadores), duplicando consequentemente a capacidade de processamento de um microprocessador. Actualmente, um microprocessador alberga centenas de milhões de transístores. Embora a verificação da Lei de Moore tenha um futuro questionável segundo as leis da física, tem sido incontestável até aos dias de hoje.

A evolução da electrónica não se resume somente à invenção dos semicondutores e do transístor, ou à observação de Moore. Na verdade, tem sido a progressiva e nos últimos anos massiva adesão das sociedades humanas às novas tecnologias, que permitiu o despoletar de novas ideias inovadoras e soluções para os mais variados problemas e dificuldades quotidianas. Os computadores e as comunicações, com funcionamento cada vez mais eficiente e eficaz, têm sido alvo de interesse crescente das pessoas à medida que estas vão perdendo o "*receio*" e a utilização das tecnologias se vai tornando mais transparente da óptica do utilizador.

Na mesma medida, estas inovações só têm sido possíveis devido à evolução da computação e da informática. Foi fundamental o aparecimento de técnicas de *engenharia do software* novas e impulsionadoras que resolvessem problemas cada vez mais complexos. Em contraste com a evolução do *hardware*, a evolução do *software* era verdadeiramente lenta. Actualmente, e quase quatro décadas após o aparecimento do primeiro computador pessoal, é inquestionável a evolução da própria computação, abrindo-se um vasto leque de linguagens de programação, ferramentas e plataformas que facilitaram a aderência das novas tecnologias. Todos os dias, novas aplicações facilitam a utilização dos equipamentos electrónicos, oferecendo cada vez maior dinamismo, oportunidades de inovação e entretenimento.

O sucesso da revolução microelectrónica possibilitou o tremendo sucesso das comunicações digitais. De novo, o sucesso das comunicações digitais deveu-se à adesão das sociedades

humanas às suas aplicações, à medida que a confiabilidade crescia e a troca de informação e conhecimento aumentava. O mundo abria portas à Era da Informação, anulando barreiras como a distância e o isolamento entre países.

A partir dos segmentos da microelectrónica, da computação e da comunicação nasceu um novo conceito: o dos Ambientes Inteligentes (AmI). Este conceito apresenta um novo paradigma que alberga sistemas digitais embutidos, capazes de processar vários factores sobre o ambiente e de comunicar informação útil com vários propósitos. O objectivo é incluir num espaço, e sobre um determinado *contexto*, um número considerável de dispositivos distribuídos que operem em conjunto de forma transparente e em consonância com as pessoas e os objectos diariamente. Estes ambientes digitais prevêm a curto e médio prazo as necessidades das pessoas. São personalizáveis e adaptativos consoante os requisitos das pessoas e respondem à presença destas e de objectos [LFZS09].

Mark Weiser idealizou uma sociedade digital no seu trabalho "*The Computer for the 21st Century*" em Setembro de 1991, ao qual denominou de *Computação Ubíqua*. Foi pioneiro na sua visão de como os computadores, de uma forma algo omnipresente, poderiam um dia ajudar as populações nas tarefas do dia-a-dia e na distribuição de informação útil. Aqui os computadores seriam dispositivos móveis e/ou embutidos no ambiente que recolhiam dados do interesse dos utilizadores, processavam os valores e punham a informação disponível e acessível de uma forma ubíqua na rede. Na última década tem-se assistido ao fomentar de vários projectos e trabalhos que partilham directa ou indirectamente a visão de Weiser. Naturalmente, a miniaturização da electrónica e o fornecimento de elevadas capacidades de processamento, comunicação e armazenamento de informação de forma eficiente, ajuda no desenvolvimento dos AmI como um segmento cada vez mais importante e inovador das novas tecnologias. Mais uma vez, será a aceitação deste tipo de aplicações em ambientes e redes inteligentes pelas sociedades humanas que ditará o sucesso das mesmas no futuro.

A gradual informatização da sociedade tem transformado o modo como vivemos. A computação ubíqua promete continuar esse percurso. Antes, as redes de comunicação e computadores eram maioritariamente cabladas, isto é, realizadas através de cabos de cobre e de normas como o IEEE 802.3 (Ethernet). A visão de Weiser e dos que o sucederam consistia em oferecer inteligência aos ambientes através de tecnologias RFID. Actualmente, com a evolução das redes sem fios e a sua larga adopção da parte dos utilizadores, esta visão da ubiquidade e da livre mobilidade está cada vez mais nítida e atingível. Nomeiam-se inúmeras normas e tecnologias em constante actualização que oferecem consistência e elevadas garantias de qualidade de serviço aos utilizadores, como as redes Wi-Fi e 3G.

Estão disponíveis estruturas de comunicação que podem ser reutilizadas para a criação destes ambientes inteligentes, interligados maioritariamente por tecnologias *sem fios*. Igualmente, podem ser reutilizadas arquitecturas computacionais maduras e consistentes. A própria

Internet e a Web são excelentes exemplos de tecnologias futuramente reutilizáveis. A Internet agrega o maior conjunto de redes de comunicação à escala mundial, composta por servidores, computadores, e outros dispositivos com capacidade para se ligarem à rede. Juntamente com o esmagador sucesso da Internet verifica-se a afirmação da Web. O *World Wide Web* (WWW, ou simplesmente W3) foi criado pelo cientista do CERN, Tim Berners-Lee, em 1989. O objectivo inicial visava construir um sistema de hipertexto como troca de informação relacionada com o CERN, direccionado para os físicos e engenheiros. Este sistema global de informação tornou-se um sucesso em pouco anos anos. Em 1993 a equipa de CERN disponibilizou o seu código fonte, tornando-o *open-source software*, contando actualmente com mais de 500 servidores, a ocupar 1 % do tráfego Internet [CERN1]. Até à actualidade, foram construídas e actualizadas arquitecturas, protocolos e *web browsers* que tornaram fácil a utilização do W3 de forma livre, fazendo dele o sistema mais utilizado da Internet. Em 2013, ultrapassou-se a barreira dos 100 petabytes de dados relacionados somente com o CERN [CERN2].

1.2 Motivação, Desafios e Abordagem

Basicamente, um AmI é um sistema electrónico e computacional inserido nos mais variados espaços e objectos que criam processos e oferecem serviços. O objectivo do AmI é automatizar e monitorizar funções diárias, simples e complexas, tendo em conta o estado dos acontecimentos no mundo real onde se insere. Do lado do utilizador, esta visão de uma *computação ubíqua* permite antecipar acções, obter informação sobre si e sobre o ambiente, interagir com aplicações ou dar mais liberdade de forma praticamente transparente.

Um ambiente inteligente funciona a partir de dispositivos fixos embutidos e equipamentos portáveis que actuam no ambiente consoante o que dele *sentem*, trocando a maioria das vezes informação e dados. Através dos sensores e actuadores que compõem o sistema, este actua e processa consoante as alterações e o estado actual do ambiente sendo o utilizador e/ou o objecto parte do núcleo do sistema. O contexto é esta informação gerada do ambiente. Contexto é a informação que descreve a situação de uma entidade, sendo esta entidade o objectivo alvo mais relevante da aplicação inserida no ambiente [LFZS09].

Um AmI é composto por três processos principais: aprendizagem, comunicação e actuação, como se apresenta na Figura 1.1. A comunicação é essencial após a aprendizagem. O processo de aprendizagem é o processo de recolha de dados essenciais do estado da aplicação. A comunicação permite a troca de informação sobre os dados recolhidos, muitas vezes de uma entidade para outra.

Figura 1.1 - Os três processos principais dos AmI.

O conceito "Ambiente Inteligente" consegue ir mais além da utilização de dispositivos electrónicos móveis ou embutidos no ambiente envolvente. Segundo os princípios básicos apresentados por Mark Weiser, as futuras aplicações AmI passarão pela introdução de objectos diários e comuns com capacidade de processamento, armazenamento e comunicação. Ou seja, qualquer objecto físico que interaja com as pessoas poderá ser capaz de aumentar a actividade humana e ajudar a realizar algum objectivo específico.

Os princípios gerais dos ambientes inteligentes foram classificados em [CLSB12]. Vários autores deram a sua contribuição na enumeração de vários princípios para os ambientes inteligentes. Apesar de visões diferentes os autores contém conceitos semelhantes tal como detecção e correcção de erros, consistência, feedback, controlo do utilizador sempre que este desejar, flexibilidade e fácil utilização (aplicações ao nível do utilizador e camadas adjacentes 100 % transparentes).

Duma perspectiva do ser humano inserido no ambiente, interagindo com este e partilhando as suas actividades com outros, existem dois tipos de canais de comunicação *humano-ambiente*: comunicação Ambiente-para-Humano e comunicação Humano-para-Ambiente. A Tabela 1.1 classifica os princípios associados e ambas as comunicações em detalhe.

Tabela 1.1 - Interacção entre humanos e ambiente.

Tipo de Comunicação	Descrição	Princípios
Ambiente-para-Humano	Tecnologias que sejam entendidas pelo ser humano de *outputs* vindos do ambiente.	1.Perceptibilidade 2.Compreensibilidade
Humano-para-Ambiente	*Input* de dados de seres humanos para o ambiente através de sensores.	1.Capacidade de actuação 2.Facilidade e liberdade 3.Agradável 4.Segurança 5. Interoperabilidade

Entre todos estes princípios, a tecnologia deve ser escalável, as falhas não deverão comprometer o correcto funcionamento do sistema, os algoritmos e as redes de comunicação devem ser o mais eficientes possível.

O ambiente é descrito como personalizável, adaptativo, antecipatório, ubíquo, interactivo, interoperável, distribuído e escalável. Os AmI são por isso construídos com base em vários aspectos. O primeiro é a capacidade de oferecer e possuir autonomia. O principal motivo da existência de AmI é a liberdade do ser humano face às funções diárias, das mais complexas às mais simples. As funções fornecidas ao sistema encontram-se sempre na possibilidade de tornar essas acções automáticas, transparentes aos utilizadores. Igualmente, o sistema deverá também possuir o máximo de autonomia possível. Um sistema autónomo e auto-organizável é um sistema inteligente.

Os dados devem ser consistentes. Um sistema inteligente inclui-se no mesmo conceito dos sistemas distribuídos. Uma das restrições dos sistemas distribuídos é a consistência dos dados, isto é, os dados e as suas réplicas (dados redundantes) devem estar actualizados e coerentes. Caso a consistência dos dados falhe, os utilizadores acederão a dados desactualizados, quebrando o correcto fornecimento de informação.

O funcionamento transparente é, desde sempre, o principal motivo para que qualquer pessoa seja um utilizador activo das novas tecnologias. Se os sistemas forem demasiado complexos e difíceis de utilizar, a sua adopção estará gravemente limitada. O objectivo é fazer com que as aplicações AmI sejam facilmente utilizáveis por todos.

A segurança é outro tópico fundamental em AmI. Desde o momento que algumas funções diárias da nossa vida passam a ser exercidas por dispositivos e máquinas que comunicam dados e informações pessoais, também ficamos mais expostos. Se a rede de comunicação que o ambiente inteligente necessariamente possui estiver desprotegida, qualquer pessoa pode visualizar, alterar ou desviar os dados gerados dos AmI que nos rodeiam.

Actualmente existem leis que protegem o direito à privacidade. E hoje em dia, a tecnologia disponível para as pessoas possui graves possibilidades de violar esses direitos. Consequentemente, essa tecnologia tem obrigatoriamente de possuir segurança e confiabilidade das entidades que as governam. A troca de dados de um paciente médico entre farmácias e hospitais, conversas em redes sociais e móveis, serviços de mensagens, web sites e serviços web acedidos, são casos concretos em que a grande maioria dos utilizadores não aceitam que a informação seja do conhecimento de pessoas desconhecidas. As aplicações no futuro não fugirão à regra. A tecnologia poderá oferecer conforto mas aumenta a necessidade de manter os níveis de segurança elevados. Várias aplicações seguem a estratégia de escolher

qual a informação pessoal que pode ser partilhada com terceiros. Uma abrangência concedida da privacidade com conhecimento e aceitação prévia. Outras partilham dados de grupos específicos de pessoas, como uma família numa casa. Noutros casos, os dados serão necessariamente partilhados com especialistas, como o médico ou a entidade controladora de um serviço de monitorização.

O controlo sobre os dados, as informações e funções de um sistema podem-se basear em dois princípios: controlo centralizado e controlo descentralizado.

O controlo centralizado é utilizado em aplicações inteligentes de espaços públicos, como cidades e jardins. A monitorização do tráfego citadino ou das luzes noturnas de uma região é realizada por entidades únicas maioritariamente sem fins lucrativos e de utilidade pública. A monitorização do estado destes ambientes deve ser devidamente controlada restringindo permissões de dados mais sensíveis de serem visualizados, e assegurando que o controlo sobre o estado dos dispositivos e suas funções é unicamente realizado por essa entidade.

O controlo descentralizado é utilizado em aplicações de carácter privado, como a automação de uma casa, de um edifício ou de um sistema agrícola. Nestas aplicações, o utilizador ou um grupo coordenador é o principal coordenador do ambiente. Os dados podem ser visualizados e monitorizados por pessoas autorizadas. Este é o tipo de arquitectura *Personal Environment Service* (PES) [OH10].

Figura 1.2 - Características fundamentais das aplicações AmI.

As aplicações AmI são maioritariamente locais. A intensão é controlar e monitorizar determinadas entidades, objectos ou aspectos do ambiente, inseridos ou pertencentes num local. A raiz de qualquer aplicação passará por construir uma rede de comunicação de área

local sem fios – *Wireless Local Area Network* (WLAN). Algumas características são essenciais como a cobertura da rede, interferências com redes vizinhas e eficiência energética. Esperam-se que existam milhares ou milhões de pequenos dispositivos embutidos no ambiente, a esmagadora maioria sem fios. Nasce aqui uma severa limitação quanto a consumos de energia, primeiro porque os dispositivos são em grande número e segundo por operarem "*livres de fios*". Estas redes locais sem fios de baixa potência – *Low Rate Wireless Personal Area Networks* (LR-WPAN) - são alvo de importantes investigações que melhorem os seus desempenhos.

A disseminação da informação proveniente das AmI além de ubíqua e transparente, deverá ser segura e estar consistentemente disponível. O objectivo é oferecer livre mobilidade e facilidade de acesso aos serviços AmI. As redes móveis são poderosas candidatas na construção de uma espinha dorsal (*backhaul*) para gerir e promover as características necessárias no acesso aos AmI. Devido ao fácil acesso à Internet da parte das tecnologias de redes móveis e à fácil utilização do sistema mundial W3, estes são sistemas de comunicação com elevada capacidade para oferecerem suporte às aplicações AmI de e para os utilizadores. No entanto, existem vários desafios que necessitam de ser ultrapassados. Actualmente, as redes de comunicação consomem 2 % da energia produzida mundialmente [HBB11] e mais de cinco biliões de dispositivos móveis estão ligados a estas redes. As estimativas calculam um aumento para cinquenta biliões no final da década. Com a integração de redes com centenas ou milhares de dispositivos embutidos que recolhem e comunicam dados prevêem-se números futuramente bem maiores. Portanto, o consumo energético, a utilização dos recursos de rádio e o crescimento exponencial de dados trocados através das redes interligadas são apenas alguns exemplos de desafios que necessitarão de ser continuamente melhorados/aperfeiçoados.

1.3 Objectivos Principais

A dissertação tem o objectivo de apresentar soluções em redes de comunicação para as aplicações AmI, focando-se essencialmente em redes sem fios embutidas no ambiente que recolhem e partilham informação de forma autónoma e inteligente. Os objectivos principais da dissertação são:

- Construir o estado da arte das redes de comunicação e dos sistemas de informação idealizados para aplicações em Ambientes Inteligentes;
- Apresentar arquitecturas genéricas e normalizadas;
- Adoptar e descrever as características de uma rede capilar no âmbito das comunicações *Machine-to-Machine* e *Internet of Things*;
- Realizar um estudo das plataformas *hardware* disponíveis no mercado para redes de sensores sem fios;

- Descrever uma camada de adaptação para a utilização do protocolo IP em redes sem fios de potência e ritmos de transmissão reduzidos;
- Propor estratégias de encaminhamento de dados, organização topológica e prevenção/correcção de falhas em redes de sensores sem fios;
- Realizar um estudo das métricas mais utilizadas em protocolos de encaminhamento para redes de sensores sem fios.

1.4 Contribuições

Até ao momento, o trabalho desta dissertação de mestrado foi valorizado através de duas publicações aceites e apresentadas em conferências. O primeiro artigo científico, foi publicado no 9^{th} *Conference on Telecommunications - ConfTele 2013*, em Castelo Branco, Portugal, que discute e analisa os tempos de latência de transição de estado dos dispositivos em redes de sensores sem fios [BGV13a].

O segundo artigo científico foi publicado no 24^{th} *International Symposium on Personal Indoor and Mobile Radio Communications - PIMRC 2013*, em Londres, e engloba as características energéticas dos dispositivos em redes de sensores sem fios, para a integração de um sistema de recolha de energia electromagnética [BSGT13a]. Este último artigo foi também apresentado em reuniões do Comité de Gestão do COST IC1004, [BSGT13b], e incluem diferentes aspectos do trabalho da equipa de investigação dos projectos PROENERGY-WSN [PWSN13] e CREaTION [CREa13].

Adicionalmente, o primeiro artigo científico citado foi submetido recentemente para o *Best Student Paper Award* da 7^{th} *URSI Seminar of the Portuguese Committee*, em Lisboa, Portugal [BGV13b].

Foi igualmente realizado um relatório interno no âmbito do projecto PROENERGY-WSN que engloba o estudo e análise realizadas nesta dissertação sobre a eficiência energética e tempos de vida das plataformas *hardware* para RSSF disponíveis no mercado [BTGV13].

Futuramente, pretende-se incluir a análise e discussão dos resultados obtidos por simulação do protocolo RPL em conjunto com o protocolo 6LoWPAN, realizada nesta dissertação, numa única publicação.

1.5 Estrutura da Dissertação

Esta dissertação está organizada em sete capítulos, incluindo este, e mais três anexos. O Capítulo 2 apresenta o estado da arte de dois conceitos intensivamente utilizados e descritos nos últimos anos: A *Internet of Things* e as comunicações *Machine-to-Machine*. Ambos os conceitos convergem na concepção, realização e implementação das redes de comunicação em ambientes inteligentes. Suportados tanto por tecnologias antigas como mais recentes, os

estudos mais concretos e precisos direcionam-se para a implementação de redes heterogéneas fortemente interligadas e com objectivos bastante específicos. Para que seja possível, vários projectos e propostas estão actualmente a ser discutidas para a criação de arquitecturas e cenários que sustentem as aplicações idealizadas e ultrapassagem os desafios previstos e descritos na literatura. Como conclusão do capítulo, são descritas as aplicações mais comuns em AmI, bem como os ambientes onde estes se inserem e alguns projectos actualmente implementáveis. No Capítulo 3 é proposto a utilização das redes de sensores sem fios como principal suporte das aplicações AmI. As redes de sensores sem fios têm um imenso potencial para serem adoptadas como as principais redes de comunicação embutidas nos ambientes e objectos inteligentes, gerarem os dados necessários e disseminarem informação relevante das aplicações. Existe um vasto conjunto de protocolos MAC e de encaminhamento que integradas com a norma IEEE 802.15.4 e juntamente com os requisitos e equilíbrios (*tradeoffs*) impostos, oferecem robustez, eficiência e fiabilidade a estas redes. Assim, o quarto capítulo descreve o funcionamento das duas primeiras camadas protocolares, física e ligação de dados. No Capítulo 4 é realizada uma avaliação das plataformas *hardware* mais conhecidas disponíveis no mercado, nomeadamente os consumos energéticos e tempos de vida útil das mesmas. O Capítulo 5 apresenta a camada de rede. Esta camada é suportada por uma sub-camada de adaptação com o protocolo 6LoWPAN cuja adopção permite a atribuição de endereços IPv6 nas redes de sensores sem fios. A camada de rede proposta utiliza o protocolo de encaminhamento *Routing Protocol for Low-power and Lossy-networks* (RPL) concebido pelo IETF para a norma IEEE 802.15.4. São propostas técnicas para a construção topológica, prevenção/recuperação de falhas e são apresentas as métricas normalmente utilizadas nos protocolos de encaminhamento destas redes. No Capítulo 6 são descritas as simulações realizadas para as métricas consideradas em RPL e apresentados os resultados dessas simulações. Por fim, o Capítulo 7 apresenta as conclusões e termina com algumas sugestões para trabalho futuro.

A dissertação também contém três anexos. O primeiro (Anexo A) descreve o simulador OMNeT++, utilizado para obter os resultados pretendidos com o protocolo RPL e com as métricas de custo consideradas. O segundo (Anexo B) apresenta a *framework* MiXiM, utilizada em conjunto com o simulador OMNeT++, que implementa as características das redes de sensores sem fios e norma IEEE 802.15.4. O terceiro (Anexo C) descreve e apresenta o código desenvolvido no simulador.

Capítulo 2

Redes de Comunicação em Ambientes Inteligentes

2.1 Internet of Things

Os ambientes inteligentes são compostos por aplicações e serviços que em conjunto constroem um mundo digital integrado no mundo real, envolvendo pessoas e objectos.

Desta visão nasceu um conceito no qual integra o mundo real com o mundo virtual: A *Internet of Things* (IoT), ou *Internet das Coisas*. A IoT visiona a interligação de dispositivos electrónicos em rede onde, de forma ubíqua, tem capacidade para aproximar cenários diversificados do mundo real e das pessoas. A definição de IoT é simples e simultaneamente complexa, com diferentes visões e dependendo fortemente da aplicação e dos serviços pretendidos. Actualmente, facilmente se constata que a permanente presença da Internet alterou radicalmente a maneira de pensar e viver das sociedades. Nas últimas duas décadas a partilha de informação, bem como a facilidade de acesso e disponibilização da mesma, cresceu exponencialmente devido à existência da Internet. De forma praticamente ubíqua, a Internet dá acesso a notícias, eventos, reuniões, seminários, entretenimento e todo o tipo de informação. Alterou a forma como as pessoas acediam a produtos e serviços. Na mesma medida, aproximou pessoas, culturas e países.

A Internet tradicional é uma rede global que interliga e aproxima as pessoas, possibilitando a troca de informação, dados e conhecimento. A sua evolução ao longo dos anos abriu portas e derrubou progressivamente fronteiras. Comumente, a definição de Internet pode ser resumida da seguinte forma:

> "A Internet é uma rede global de dispositivos que tem como objectivo fornecer uma variedade de serviços de informação e comunicação, constituída através de redes interligadas cujo funcionamento é garantido por protocolos de comunicação normalizados."

A IoT transportará esta Internet tradicional para outro patamar. O seu potencial está precisamente, mais uma vez, na forma como todo o tipo de informação vai estar disponível para as pessoas. E que tipo de informação pretendida pelas pessoas. A IoT interliga os objectos do dia-a-dia, em casa, nas cidades, nos edifícios e na indústria. O objectivo é novamente a troca de informação relevante, mas na IoT, esta informação é proveniente de todo o tipo de objectos e dispositivos com capacidade de processamento e comunicação. O contraste com as aplicações de AmI é perfeitamente visível. A IoT é a rede global que suportará a ligação entre as aplicações AmI e as pessoas.

Na prática, oferecer "inteligência" a um objecto é adicionar-lhe microprocessamento, sensores/actuadores e capacidade de comunicação com outros elementos de uma rede. Os exemplos mais básicos e actuais são as aplicações de monitorização e controlo. A informação proveniente dos objectos é partilhada e visualizada (em tempo real ou não) por utilizadores, *softwares* ou servidores dedicados, participantes activos das aplicações, com objectivos bastante específicos. Os "objectos inteligentes" possuem as seguintes características:

- Têm existência física e real no mundo;
- Têm uma identidade;
- Comunicam e interagem;
- Realizam funções, tomam decisões e têm um comportamento dentro de um contexto.

O conceito IoT surgiu do artigo "*The Electronic Product Code (EPC) – A naming Scheme for Physical Objects*" publicado em 2001 [B01] cujo conceito consistia principalmente na substituição do sistema UPC (*Universal Product Code*), mais conhecido por *código de barras*, por um sistema que usufruiria das capacidades da Internet e da globalização, tirando proveito da infraestrutura das redes de comunicação digital [B01]. Este sistema ficou conhecido como o *Electronic Product Code* (EPC). Ao contrário do UPC, que identificava um único tipo de produto, este novo sistema identifica especificamente o produto em si dentro de um conjunto. Cada objecto tem uma identidade. O EPC possibilita o acompanhamento do objecto físico ao longo do seu tempo de vida através de *tags* RFID. A informação sobre o seu estado, localização e outras informações úteis não se encontram directamente armazenadas nos produtos. O sistema utiliza as redes de comunicação digital (Internet), onde a informações e os dados são armazenados em servidores.

Inicialmente proposto, o ID dos objectos é gerido por um sistema denominado *Object Name Service* (ONS), que traduz o código ID para endereços IP e que por sua vez são identificados pelo sistema *Uniform Resource Identifier* (URI). A reutilização de protocolos já existentes e extremamente utilizados é fundamental para o sucesso do novo sistema de identificação.

Os princípios da IoT apoiam-se na evolução de infraestruturas normalizadas, escaláveis, abertas e seguras no âmbito das tecnologias de informação e comunicação. Como por exemplo, a miniaturização e decréscimo dos custos de dispositivos electrónicos, evolução das redes sem fios, desenvolvimento de novas tecnologias e protocolos de comunicação, aplicações e *frameworks*.

As abordagens globais deste tipo de infraestrutura são reconhecidas como possíveis ferramentas a utilizar ao nível do negócio e comercial, no futuro. As *intranets of things* possibilitaram alargar a gestão dos negócios das empresas e facilitar o controlo e supervisão dos seus produtos. A escalabilidade de algumas tecnologias de informação expandiu as abordagens iniciais para uma *extranet of things*. À medida que novas ideias e grupos de

trabalho surgiram neste âmbito, a ideia central de *Internet of Things* permitiu a sua adesão. Há disponível um imenso número de serviços e informação, através de ferramentas facilmente acessíveis e utilizáveis pelas pessoas, desde equipamentos móveis, até infraestruturas e plataformas escaláveis, abertas e integradoras de outros serviços.

Vários aspectos da Internet tradicional são essenciais na construção da IoT. O primeiro é o *Internet Protocol* (IP), sendo este o suporte principal para o sucesso da Internet. O segundo, amplamente aceite e discutido, é a arquitectura *Representational State Transfer* (REST). Estes são os segmentos que contribuíram para a elaboração, criação e/ou actualização de inúmeras ferramentas e tecnologias que possibilitam o aparecimento lento mas progressivo da IoT.

O termo *Web 2.0* ofereceu uma nova visão sobre a Internet como uma ferramenta mais vasta, dinâmica e diariamente utilizável por milhões de pessoas em todo mundo. A web deixou de ser vista como um conjunto estático de serviços e páginas de hipertexto, transitando para uma plataforma aberta, normalizada e global na troca de serviços, como *wikis*, redes sociais, fóruns, bases de dados, *applets*, *live vídeo/streaming* e outros *web-services*. A interação humana com a web é cada vez mais fácil através de plataformas de utilização intuitiva, sem que seja necessário alterar os padrões e as normas que compõe a Internet.

Vários autores defendem que a IoT tem vindo a ser fortemente influenciada pelo aparecimento da Web 2.0, impulsionando desenvolvimentos paralelos de ambos os conceitos [UHM11]. No contexto da universalidade e constante presença na web, os utilizadores poderão aceder às informações dos objectos do mundo real que os rodeia, onde estiverem e quando quiserem, como se estes fossem serviços puramente virtuais.

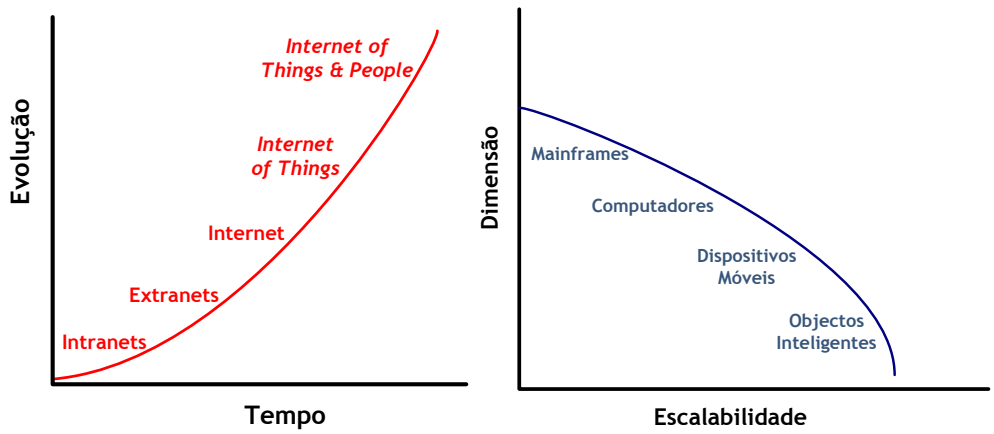

Figura 2.1 - Perspectivas da evolução das redes de comunicação e dos dispositivos.

A IoT só é desejável se tiver capacidade de se auto-organizar e auto-conFigurar, sem a intervenção directa das pessoas e com absoluta fiabilidade quanto ao seu interfuncionamento e à sua interoperabilidade. Isto só será intrinsecamente possível se, para além da estrutura actual da Internet permanecer praticamente inalterada, for assegurada a normalização de protocolos que assegurem essa dinâmica, for possível assegurar extensibilidades, tiver actualizações contínuas e permitir incorporação de futuras tecnologias facilmente adaptáveis às existentes. Na prática, a IoT deve ter um nível elevado de escalabilidade. As aplicações que trazem até às pessoas a informação e serviços da IoT têm como base, o mesmo formato dos serviços virtuais da actualidade. Mas, ao contrário dos serviços virtuais suportados por *data centers* e servidores, estes serviços serão físicos e reais fornecidos directamente e/ou indirectamente pelos objectos e dispositivos embutidos nos ambientes.

Vários projectos encontram-se no âmbito da IoT. Alguns desses projectos assentam na vontade e motivação em integrar sistemas inteligentes diferentes. O EPoSS é uma iniciativa que contribui para o crescimento da estratégia europeia face a esta visão da IoT. Em [LS08], sugere-se que todos os objectos e dispositivos, desde computadores, sensores, actuadores e dispositivos móveis, tenham obrigatoriamente que possuir um único endereço que os identifique, munidos da capacidade de se identificarem a si mesmos e de verificarem essa mesma identidade enquanto recolhem, processam e partilham informação. É aqui que se verifica uma semântica idêntica utilizada pela pilha protocolar TCP/IP que deu vida e forma à Internet tradicional duma maneira totalmente transparente e funcional. Além deste projecto, a União Europeia iniciou outros projectos: IoT-A, ebbits, NISB, SPRINT, NEFFICS.

A IoT tem vindo a ser, para além do RFID, contextualizada na área das redes de sensores e actuadores sem fios (RSSF). Os projectos SENSEI [ICTSE] e COBIS [COBIS] focam-se essencialmente na projecção das redes de sensores e actuadores sem fios como fontes de dados capazes de realizar tomadas de decisão autonomamente e comunicar com redes externas. O SENSEI propôs construir uma *Framework* que possibilita realizar esta integração em conjunto com serviços que gerem e entregam a informação associada a determinados contextos no ambiente. O COBIS propôs a realização de uma arquitectura de sensores embutidos em ambientes *enterprise* orientados por serviços, focando vários aspectos das redes de sensores sem fios quando integradas no sector do negócio e da indústria. Um dos objectivos era, por exemplo, gerir e monitorizar o estado de objectos e ferramentas, considerando-os inteligentes o suficiente para colaborarem e comunicarem.

O grupo IERC (*IoT European Research Cluster*) tem como missão aproximar os projectos europeus com o objectivo de definir e contextualizar ideias e desenvolvimentos em relação à IoT [ITREU]. O grupo realiza vários eventos, debates e documentos de forma a dar conhecimento e promover a futura adopção e utilização à escala mundial da IoT. Após vários trabalhos o grupo apresentou um livro em 2012 que reúne todas as ideias e inovações realizadas para a IoT [SVFF12].

Foram propostas várias plataformas para dar suporte aos dispositivos presentes na IoT. As redes de comunicação, como as RSSF, são a principal base da IoT pois agrupam um número indispensável de capacidades computacionais e de comunicação que a tecnologias RFID não conseguem atingir.

2.1.1 Protocolo Internet

Um dos protocolos que mais contribuiu para a fácil manutenção e consequente utilização massiva da Internet foi o *Internet Protocol* (IP), um protocolo de rede implementado pela pilha protocolar *Transmission Control Protocol/Internet Protocol* (TCP/IP).

Os dois modelos de referência mais importantes que estruturam a arquitectura lógica e as funções internas das redes de comunicação são o modelo *Open Systems Interconnection* (OSI) e o modelo TCP/IP (Figura 2.2). A pilha protocolar TCP/IP organiza hierarquicamente a interligação das redes ao mesmo tempo que abstrai as camadas, tornando-as independentes umas das outras. Isto é conseguido através da normalização dos protocolos das várias camadas, com características que as tornam independentes e através do encapsulamento de tramas desde a camada mais "superior" (Camada de Aplicação) até à camada mais "inferior", constituída pelo *hardware* (Camada Física). Desta maneira, independentemente do sistema *hardware* utilizado, a pilha TCP/IP é implementada, permitindo uma interface comum entre a rede de comunicação e as aplicações ao nível do utilizador. O sistema deve ser transparente para o utilizador, ou seja, basicamente "esconde" todos os procedimentos protocolares e físicos que possibilitam a entrega de dados às aplicações. Outro aspecto fundamental da abstração entre camadas é a capacidade de desenvolver, actualizar e/ou estender implementações alternativas e secundárias às já existentes.

O modelo OSI é mais pormenorizado na descrição das camadas e separação das mesmas. No entanto, o modelo internet TCP/IP, sendo um modelo mais genérico, é igualmente mais flexível como ponto de partida para outros modelos de arquitectura semelhantes.

O protocolo IP insere-se no contexto desta natural abstração entre camadas, não necessitando, por exemplo de saber se a comunicação é síncrona (com ligação) ou assíncrona (sem ligação). O IP possibilita a construção de encaminhamentos hierárquicos na Internet atribuindo endereços lógicos e únicos ao nível da rede, que por sua vez divide os dispositivos em sub-redes ligadas à rede pública através de pontos de acesso.

O IP é um protocolo com uma estrutura extremamente simplificada. Não realiza controlo de erros, retransmissões de pacotes e controlo do fluxo dos dados. Tem como objectivo principal o reencaminhamento de pacotes através de redes interligadas a partir de endereços lógicos atribuídos aos dispositivos. Sendo independente, tanto do *hardware,* como das aplicações, é a principal base que oferece suporte eficiente e extremamente eficaz às intercomunicações.

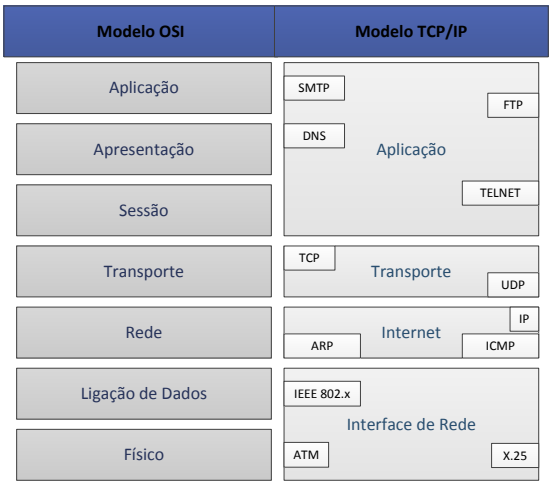

Figura 2.2 - Modelo OSI vs Modelo TCP/IP.

Existem vários protocolos que se baseiam no esquema do IP, entre eles o IPX e o *Apple Talk*. O mais difundido e utilizado é o IPv4 devido à sua elevada escalabilidade. O endereço lógico IPv4 tem 32 bits, apresentados em formato decimal, que identificam os dispositivos (Figura 2.3). O primeiro conjunto de bits identifica a sub-rede onde se encontram os dispositivos, enquanto que o segundo conjunto identifica o dispositivo em si. Existem quatro tipos de classes (A, B, C, D e E) com diferentes comprimentos de conjuntos de bits para identificar a rede e consequentemente dar capacidade numérica para atribuir endereços aos *hosts* nas diferentes redes. O endereçamento realizado por classes (*classful*) está obsoleto devido à sua inflexibilidade em atribuir IPs. O endereçamento *classful* foi, por isso, substituído pelo endereçamento sem classe (*classless*) que permite uma atribuição de endereços muito mais flexível e dinâmica ao organizar os dois conjuntos de bits (rede e *host*) livremente, sem restrições ou limites impostos pelo antigo sistema de classes. O endereçamento IPv4 destingue endereços privados (endereços utilizados dentro de redes locais) de endereços públicos (utilizados em redes públicas, como por exemplo a própria Internet).

Apesar do seu bom desempenho, os endereços lógicos IPv4 estão actualmente a ser substituídos pela versão 6, o IPv6, com um comprimento total de 128 bits e com capacidade para acomodar um número extremamente mais elevado de endereços. Ao contrário do IPv4, o IPv6 apresenta os endereços no formato hexadecimal. O IPv6 especifica dois tipos de endereçamento: global e local. O endereço global é utilizado através da rede pública da Internet para identificar externamente os *hosts*. O endereço local é, como o nome indica, utilizado dentro das redes locais onde estão ligados os *hosts*.

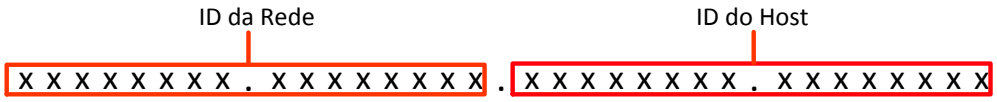

Figura 2.3 - Formato geral do endereço IPv4.

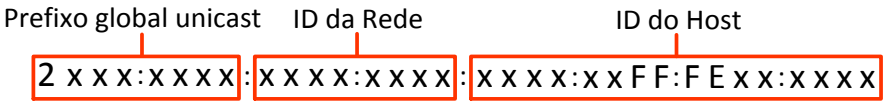

Figura 2.4 – Formato do endereço IPv6 global.

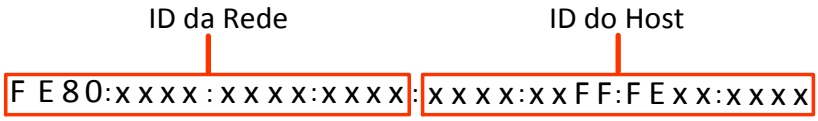

Figura 2.5 – Formato do endereço IPv6 local.

O formato do IPv6 global (Figura 2.4) é formado por um prefixo global nos primeiros *n* bits mais significativos do endereço (poderão ser de 32, 64 ou até de 128 bits) que identifica a topologia pública atribuída pelo *Internet Service Provider* (ISP). Os três bits mais significativos do prefixo têm sempre o valor fixo 001 (2, em decimal). O grupo seguinte contém o endereço da sub-rede do *host* com $64 - n$ bits. O último conjunto de 64 bits é o endereço do próprio *host*, cujo ID é directamente conFigurado através do endereço MAC (formato EUI-64).

O endereço IPv6 local (Figura 2.5) é o endereço utilizado dentro das redes locais, análogo aos endereços privados IPv4. O endereço IPv6 local é dividido em dois grupos: ID da rede e ID do *host*. Todos os endereços locais são identificados com um prefixo de 16 bits iniciais com o formato hexadecimal FE80::/64.

O ID do *host*, tanto nos endereços globais como nos endereços locais, é baseado no endereço MAC, de 48 bits, da máquina. Este ID de 64 bits tem o formato EUI-64, o qual adiciona um conjunto de 16 bits, entre os primeiros 24 bits e os últimos 24 bits do endereço MAC, com o valor hexadecimal FFFE, como é apresentado na Figura 2.6.

Existe uma estratégia básica e simples para diminuir o comprimento elevado dos endereços IPv6. Conjuntos de bits com o valor 0 (zero) podem ser "escondidos", diminuindo o *overhead* dos endereços nos cabeçalhos. Por exemplo, um endereço local com o formato FE80:0000:0000:0000:1111:2222:3333:4444/64 pode ser representado como FE80::1111:2222:3333:4444/64, diminuindo o número de bits enviados. Esta técnica só pode ser aplicada a um conjunto, ou vários conjuntos seguidos de bits a zero por cada endereço, isto é, se existirem dois grupos de 16 bits a zero que não estejam seguidos, a técnica só é aplicável a um dos conjuntos.

Ambas as versões do IP (versões 4 e 6) fazem o encapsulamento de um cabeçalho de rede ao segmento de dados proveniente da camada de transporte, contendo informação sobre o endereço lógico dos dispositivos de origem e destino. Outros campos estão presentes no cabeçalho, incluindo a variável *Time-to-Live* (TTL), cujo valor define o número de saltos

máximo que o pacote pode dar antes de ser descartado da rede, garantindo a não ocorrência de problemas de caminhos fechados (*loops*), nem a congestão da rede durante o encaminhamento de pacotes.

As causas motivantes para a progressiva adopção do IPv6 têm origem no aparecimento de cada vez mais dispositivos ligados à Internet. Estes dispositivos, extremamente diferentes em *hardware* e funções, desde servidores, computadores e *smart phones*, podem hoje, graças essencialmente ao IP, oferecer acesso à rede global. Uma das tecnologias que mais evoluiu como factor de impacto nesta matéria é o acesso *Wireless Broadband* dos dispositivos móveis que incluem a tecnologia 3G e as mais recentes implementações do aclamado 4G onde se insere a tecnologia *Long Term Evolution* (LTE) e a norma IEEE 802.16 (WiMAX).

Por outro lado, o IPv6 é auto-configurável, isto é, os endereços IPv6 são conFigurados e atribuídos pelos próprios dispositivos da rede. Ao contrário do IPv6, o IPv4 necessita de ser sempre atribuído manualmente por um administrador da rede ou por servidores que executem *Dynamic Host ConFiguration Protocol* (DHCP) para atribuição dinâmica de endereços IP. No entanto, o IPv6 pode também ser conFigurado através de servidores DHCP versão 6 (DHCPv6) se se pretender manter tabelas de endereçamento em servidores dedicados.

O IPv6 não especifica nenhum endereço de difusão (*broadcast*), portanto, este tipo de comunicação não é possível. Esta é uma vantagem face ao IPv4 pois diminui problemas relacionados com o congestionamento da rede. O IPv6 define grupos *multicast* direccionados para dipositivos ou objectivos específicos. Estes grupos de endereçamento *multicast* distinguem-se pelo prefixo utilizado nos 32 bits mais significativos dos endereços [Multiv6]. A Tabela 2.1 apresenta alguns endereços *multicast* definidos pelo *Internet Enginnering Task Force* (IETF) e atribuídos pela corporação responsável pela distribuição dos endereços IP, *Internet Assigned Numbers Authority* (IANA).

Outra vantagem para existir uma progressiva adopção do IPv6 é a utilização de mecanismos que criam *túneis* na rede, com o objectivo que converter os endereços IPv6 das redes locais, em endereços IPv4. Assim, é possível a travessia de pacotes IPv6 em redes IPv4, e vice-versa.

Tabela 2.1 – Exemplos de endereços *multicast* IPv6.

Endereços	Descrição
FF02::1	Endereço para todos os *hosts*
FF02::2	Endereço para todos os *routers*
FF02::6	Endereço para *routers* OSPFIGP
FF02::A	Endereço para *routers* EIGRP
FF02::1A	Endereço para redes RPL
FF02::1:2	Endereço para todos os agentes DHCP
FF02::1:3	Endereço de *Link-local Name Resolution*

Estes mecanismos de *tunneling* encapsulam os pacotes IPv6 em pacotes IPv4. Um mecanismo que torna estas comunicações IPv6-IPv4 possíveis é o *Network Address Translation – Port Translation* (NAT-PT). A versão original do NAT tinha como objectivo converter endereços IPv4 privados em endereços IPv4 públicos.

A abordagem das redes convergidas por IP pretende conceber uma arquitectura eficaz e facilmente adaptada para serviços extremamente distintos e de uma forma altamente compatível entre dispositivos e sistemas heterogéneos.

As ligações IP estão disponíveis para diferentes tipos de acesso assim como para todo o tipo de dados que circulam pela rede. Este é o conceito das *Redes All-IP*. O IP/MPLS – *Multiprotocol Label Switching* - é um tipo de arquitectura que se encaixa nas redes *all-IP*. O IP/MPLS é composto pelo *tunneling* MPLS/VPLS, tecnologia Ethernet e o encaminhamento por IP [NSZH07]. O MPLS permite o transporte de diferentes tipos de tráfego e oferece um controlo autónomo eficiente. A ligação MPLS tem de estar triplamente preparada para dar acesso Internet a serviços que requerem ritmos de transmissão elevados, *streaming*, televisão e voz, numa única ligação com uma largura de banda com vários níveis de agregação.

Como na Internet tradicional, as redes *all*-IP poderão suportar atribuições flexíveis para determinados níveis de qualidade de serviço (QoS), assim como gestão da largura de banda e tratamento do tráfego. A qualidade de serviço não pode ser afectada independentemente do tipo de acesso.

O *IP Multimedia Subsystem* (IMS), reconhecido internacionalmente, classifica-se como uma arquitectura genérica que pode ser englobada na nova estrutura visionada para as redes *all*-IP. Esta nova estrutura é composta essencialmente por uma Camada de Serviço, uma Camada de Controlo e uma Camada de Ligação para o acesso das várias redes. Iniciada pelo *3rd Generation Partnership Project* (3GPP), actualmente, o IMS oferece serviços *Voice over IP* (VoIP) e serviços multimédia, independentemente do tipo de acesso, como o *Global System for Mobile* (GSM), o *Wide Code Division Multiple Access* (WCDMA) e as redes sem fios locais – *Wireless Local Area Network* (WLAN).

O multi-acesso é um aspecto necessário no IMS. Além dos diferentes tipos de acesso e das redes serem heterogéneas, também os serviços têm diferentes requisitos, uns com necessidades elevadas de largura de banda, outros com a necessidade de reduzir a latência das comunicações. Portanto, a rede tem de estar habilitada a lidar com as diferentes características impostas pelos métodos de acesso e pelos recursos pretendidos.

Todos estes aspectos do IP (e das novas tecnologias que o incorporam) dão um enfâse maior à ideia de atribuir endereços únicos aos dispositivos da IoT. A fácil implementação do IP poderá permitir construir interligações directas ponto-a-ponto (P2P) entre a Internet e estes

dispositivos, e assim criar a IoT a partir das presentes infraestruras de redes e da utilização dos protocolos já existentes. O *IPSO Alliance* promove a utilização do IP como a solução mais fiável para a comunicação e acesso de objectos inteligentes como entidades únicas, pois prevê-se uma dimensão de milhões destes dispositivos ligados e sempre presentes na rede.

2.1.2 Arquitectura RESTful

A reutilização das tecnologias e dos protocolos presentes hoje na Internet é a base para a formação da IoT. O mesmo argumento pode ser considerado quanto à reutilização das tecnologias *web* existentes quando se pretende organizar os dados produzidos pelos ambientes inteligentes como se organizam os serviços *web*. Ou seja, as tecnologias utilizadas na construção de aplicações que acedem aos conteúdos da IoT são as mesmas hoje em dia utilizadas na *web*. Desta perspectiva nasceu o conceito *Web of Things* (WoT) [GTW10] [UHM11].

O *Remote Procedure Call* (RPC) é um processo de comunicação que utiliza o modelo *cliente-servidor* para partilhar serviços *web* entre utilizadores. Um grande número de serviços derivou deste processo, nomeadamente o *Simple Object Access Protocol* (SOAP) e o *Web Services Description Language* (WSDL). Devido à sua complexidade elevada e escalabilidade reduzida, os modelos derivados do RPC, referidos como arquitecturas baseadas em serviços, começaram a ser substituídos por arquitecturas baseadas em recursos, nomeadamente a arquitectura *Representational State Transfer* (REST).

O REST é uma arquitectura *web* que fundamentalmente reúne um conjunto de princípios que indica a correcta utilização das normas e protocolos *web* [UHM11], [GTPL09]. O REST organiza os recursos e serviços da web através dos identificadores *Uniform Resource Identifier* (URI). Estes identificadores URI são a base principal de interacção entre humanos e serviços *web*. Estes identificadores podem ser nomes, localizações ou ambos. Os nomes são uma prática muito comum para os utilizadores acederem aos recursos e serviços *web*. É através destes nomes que objectos podem ser identificados, localizados e partilhados de forma legível para o utilizador humano. Estes nomes são simbólicos, na medida em que são uma referência específica para identificadores de serviços e objectos dificilmente manipuláveis e memorizáveis pelos utilizadores (como, por exemplo, os endereços IP). O espaço de nomes é essencial para manter as regras de identificação e o contexto onde se inserem os serviços, estruturados hierarquicamente com vários serviços de nomes.

Os ficheiros *web* são identificados pelo esquema *Uniform Resource Locator* (URL), cujos nomes identificam o servidor (ou qualquer outro tipo de máquina) onde os ficheiros estão armazenados e define como os recursos podem ser obtidos (especificando o protocolo de transferência de dados como o *ftp* e o *http*). Os *Uniform Resource Name* (URN), por outro lado, representam nomes únicos. Ambos são identificadores especiais que juntos compõe o identificador URI. Portanto, o URI é o identificador geral dos recursos disponíveis na *web*.

Assim, os recursos são localizados, identificados e acedidos implementando protocolos que permitem a troca dos dados ao nível da aplicação como é o caso do *Hyper Transfer Protocol* (HTTP). O HTTP é o protocolo implementado na *web* que estende para a rede mundial a arquitectura *cliente-servidor* agregando serviços previamente existentes. Uma aplicação (*web browser*) faz pedidos a servidores que disponibilizam recursos (páginas e conteúdo *web*) identificados por URI. Como complemento, a linguagem *HyperText Markup Language* (HTML) permite a codificação da informação, isto é, facilita a visualização dos ficheiros hipertexto do lado do cliente.

O sucesso desta arquitectura coloca-a entre as melhores candidatas para a integração da IoT ao nível da aplicação. A ideia principal é aceder aos conteúdos da IoT (serviços/recursos físicos), da mesma maneira que se acedem aos recursos puramente virtuais, utilizando as mesmas ferramentas e aplicações [GTW10], [GTPL09]. Por exemplo, utilizando uma ferramenta *web*, como um *browser*, o cliente pode aceder a um servidor associado à sub-rede de dispositivos inteligentes, à memória dos próprios dispositivos ou até mesmo à leitura em tempo real de um sensor, se este estiver identificado por um URI específico e se forem implementados métodos tradicionais de http. Com a implementação da arquitectura REST em *smart phones*, estes recursos físicos podem ser facilmente acedidos independentemente do *hardware*.

As quatro características fundamentais do REST são:

- Utiliza o modelo Cliente-Servidor;
- Realiza ligações sem gravação de estado;
- Utiliza *caches*;
- É composto por uma interface uniforme.

Alguns desafios da WoT provêm dos desafios de camadas inferiores:

- Segurança e encriptação dos dados;
- Atraso do tempo de resposta;
- Elevada disponibilidade;
- Elevada escalabilidade;
- Auto-organização dos dados entre objectos e servidores.

Tem vindo a ser desenvolvida uma arquitectura REST para redes muito restritas e limitadas ao nível de recursos energéicos e espectrais. Estas redes caracterizam-se por comunicarem pacotes comprimentos bastante reduzidos, com elevadas probabilidades de perda de pacotes e consumos de energia extremamente reduzidos. Esta arquitectura é denominada *Constrained RESTful Environment* (CoRE) e está a ser desenhada exclusivamente para as aplicações IoT. O tipo de redes onde é executado o CoRE é na próxima secção.

2.2 Comunicação *Machine-to-Machine*

As comunicações *Machine-to-Machine* (M2M) nasceram, como o próprio nome indica, das tecnologias desenvolvidas para possibilitar a comunicação totalmente autónoma entre dispositivos e equipamentos sem qualquer intervenção humana. Este novo conceito tem sido aplicado a projectos e aplicações reais onde se estão presentes os ambientes inteligentes. As aplicações pioneiras foram a telemetria e o sistema de monitorização industrial SCADA.

Inicialmente, o conceito M2M foi utilizado em aplicações direccionadas para vigilância de espaços privados e segurança de espaços públicos, monitorização e *tracking* (rastreamento) de veículos ou objectos em movimento, controlo industrial e logística, negócios e compra de bens, *smart grid & metering*.

Existem oportunidades de mercado com potencial na área M2M à medida que os sistemas inteligentes se vão tornando mais concretos e realizáveis. A competitividade, a criatividade dos negócios e a sucessiva normalização de tecnologias pode abrir oportunidades para realização de novas soluções comerciais [HARBOR].

As comunicações M2M estão fortemente incluídas nas redes cabladas, mas este é um cenário que tem vindo a mudar e a transitar fundamentalmente para redes sem fios. Inicialmente, os dispositivos M2M utilizavam tecnologias como o GSM, na troca de informação simples através de serviços *Short Message Service* (SMS). Em 2003, aproximadamente 20 milhões de SMS e 30 milhões de Multimedia Messaging Service (MMS) foram enviadas por estes dispositivos M2M, mundialmente [WA09]. Actualmente, existem inúmeros projectos e negócios relacionados com a comunicação M2M, nomeadamente a telemétrica veicular, *smart grids* e saúde electrónica, através das tecnologias *Geographic Position System* (GPS), 3GPP, WLANs e redes cabladas *Controller Area Network* (CAN), *Digiral Subscriber Line* (DSL), Ethernet e fibra óptica. O objectivo principal é integrar estas redes numa só, convergindo para redes totalmente heterogéeas.

O conceito básico de uma aplicação M2M combina dois componentes principais: *Data End Point* (DEP) e *Data Integration Point* (DIP). O primeiro é o dispositivo M2M com capacidade para recolha, processamento e envio de dados. DIP é a entidade cliente no final do processo que agrega e utiliza os dados requeridos da aplicação, formado por um servidor ou um *software* [WA09]. A Figura 2.6 apresenta um exemplo deste conceito M2M.

Existe no mercado uma diversidade de equipamentos para diferentes aplicações M2M neste contexto. Por exemplo, algumas soluções para aplicações de *Tracking* integram módulos de GPS e GPRS nos equipamentos (DEP) cuja missão é enviar dados sobre a posição de veículos para um centro de monitorização (DIP) [ADV], [ADM2M].

Estes dispositivos M2M são geralmente constituídos por três módulos principais: recolha de dados, plataforma de processamento/comunicação e fornecedor de informação [SS10]. O módulo de "Recolha de Dados" é constituído por dispositivos equipados com sensores. Estes dispositivos são as principais fontes geradoras de dados sobre a aplicação. A "Plataforma de Processamento/Comunicação" é o módulo *hardware* composto por microprocessador e rádio emissor/receptor, cujas funções são o processamento dos dados e a comunicação dos mesmos para outros dispositivos ou outras redes. A tarefa principal do módulo "Fornecedor de Informação" é dar suporte à área negocial, realizando a análise e o armazenamento da informação gerada a partir dos dados.

O microprocessador é composto por módulos básicos, tal como memória, CPU, sistema operativo e identificação da plataforma. O módulo de comunicação pode ser composto por diferentes interfaces e tecnologias de comunicação. É comum que aplicações de monitorização e controlo partilhem um número elevado de tramas de dados em rajada. Mesmo que sejam tramas com comprimentos reduzidos, é necessário avaliar o impacto destas transmissões em rajada nas redes de comunicação.

Figura 2.6 - Visão básica das comunicações M2M.

A IoT e a M2M unem-se no mesmo sentido para criar condições de partilha de informação proveniente de dispositivos embutidos no ambiente, ao mesmo tempo que existe auto-organização das sub-redes, comunicação ubíqua e reutilização de arquitecturas e protocolos existentes, além das actualizações e possível introdução de novas tecnologias. No fundo, a IoT é a base principal para a interligação de várias redes de comunicação M2M entre objectos, fornecendo, ao mesmo tempo, novos serviços às pessoas. Por isso, as siglas IoT e M2M estão fortemente ligadas e representam virtualmente a mesma missão e o mesmo objectivo.

Existem alguns desafios que necessitam de ser continuamente ultrapassados. Os desafios considerados mais importantes para a realização eficiente das comunicações M2M e das suas aplicações são os seguintes como apresentados na Figura 2.7:

- Custo de implementação (e manutenção);
- Eficiência energética;
- Eficiência espectral;

- Privacidade e segurança dos dados dos sistemas e das aplicações;
- Gestão e controlo de milhões de dispositivos previstos.

O custo de implementação engloba tanto o custo dos dispositivos durante o desenvolvimento da rede, como o custo de manutenção da mesma. As estimativas indicam que perto de 500 mil milhões de dispositivos estarão ligados à Internet no ano 2020 [OECD12].

Figura 2.7 - Desafios principais da IoT/M2M.

O potencial das aplicações M2M é bastante reconhecido, mas a implementação de milhares de dispositivos em ambientes inteligentes tem custos que necessitam de ser contabilizados no sentido viabilizar economicamente os investimentos em aplicações M2M no âmbito do mercado e do negócio. Como indicado no Capítulo anterior, a miniaturização e redução do custo do *hardware* dos dispositivos facilita a ultrapassagem deste desafio. Na mesma medida, o custo é influenciado pela normalização de tecnologias e protocolos, bem como pela eficiência e a consistência do seu funcionamento. Como em qualquer outro tipo de mercado, as soluções necessitam de ser viáveis, tanto ao nível do funcionamento, como ao nível das necessidades ou desejos dos utilizadores finais. A aposta ao nível do negócio de cada aplicação obriga a um estudo profundo relativamente ao impacto no mercado, detalhando os benefícios e os custos das implementações.

A questão da eficiência energética é complexa e extremamente discutida, motivada pelo impacto ambiental das emissões de CO_2, desperdício de energia e redução dos custos associados. É cada vez mais imperativo o aparecimento e adopção de soluções que englobem tecnologias com consumos eficientes, potências reduzidas e tempos de vida úteis significativos.

Inúmeras tecnologias de redes sem fios participam na formação da IoT e nas comunicações M2M. Estas tecnologias possuem as suas próprias características de funcionamento, com

centenas de dispositivos agrupados em redes locais. O desafio da eficiência espectral inclui portanto dois aspectos diferentes: interferência espectral e atribuição de canais de ligação ascendente/descendente (*uplink*/*downlink*) que suportem o tráfego de um número extremamente elevado de pequenos pacotes enviados em rajada.

A gestão de milhares de dispositivos autónomos e interligados é um desafio complexo. Os dispositivos devem ter um determinado nível comportamental consoante a aplicação que sustentam, o ambiente que o envolve e outros dispositivos na sua vizinhança. A identidade única de cada um torna-os, em certa medida, cientes da sua própria existência e da dos outros dispositivos e outras aplicações. A auto-organização eficiente da rede tanto maior é, quanto maior também forem as capacidades de processamento e comunicação. Existem vários métodos e protocolos propostos para este tipo de auto-organização e auto-descoberta, mas apenas alguns são direccionados para a auto-gestão de milhares de dispositivos em rede. Essencialmente, a resposta a este desafio passa pela organização planeada de grupos de dispositivos, por aplicação ou por contexto. É necessária uma nova geração de processadores de comunicação que mantenham uma gestão consistente e controlada de todos os dispositivos e das suas ligações, bem como capacidade para interligar as suas funcionalidades com os serviços correspondentes [KA12].

A heterogeneidade afecta igualmente os métodos de segurança dos sistemas. A interligação de redes com características completamente distintas levanta sérias questões sobre segurança e protecção dos meios de comunicação. A comunicação entre redes de potência reduzida e redes banda larga (*broadband*) necessita de métodos optimizados de criptografia e sistemas adequados de gestão de chaves para autenticação [RNL11]. A ligação através da Internet requer protocolos de segurança para este tipo interligação de redes sem fios. A privacidade, como já visto, é um dos factores mais importantes no âmbito da segurança. Todos os utilizadores dos serviços Internet e IoT exigem a confidencialidade da sua informação. Os operadores e fornecedores terão obrigatoriamente de adoptar um compromisso de fiabilidade dos seus serviços, garantir a possibilidade de gestão da informação pelos próprios utilizadores e manter políticas de não violação dos direitos dos utilizadores. Estas indispensabilidades englobam-se na solução "*privacy by design*", onde os utilizadores possuem as ferramentas necessárias à gestão dos seus próprios dados [RNL11].

2.2.1 Arquitectura de uma Rede M2M

Essencialmente, uma rede M2M é dividida em duas sub-redes principais: A rede capilar e a rede celular. A rede capilar é composta por dispositivos embutidos no ambiente e/ou objectos, isto é, pelas fontes dos dados recolhidos. Normalmente são altamente escaláveis e utilizam protocolos que garantem o cumprimento dos requisitos mais comuns destes dispositivos, tal como requisitos de eficiência energética. A rede celular é a rede que difunde, processa e apresenta esses dados através de acesso de banda larga. A rede celular é

altamente vantajosa pois possibilita elevada mobilidade, ubiquidade e excelente cobertura. Assim, é possível reutilizar infraestruturas de redes de comunicação já desenvolvidas e implementadas.

Existem vários trabalhos que especificam os objectivos, as características e os requisitos das comunicações M2M, nomeadamente os desenvolvidos pela organização *European Telecommunications Standards Institute* (ETSI) [ETSIM2M].

A arquitectura proposta pelo ETSI [TS102690] tem dois domínios principais: o *Device Domain* (DevD) e o *Network Domain* (NetD). Entre os dois encontra-se o *Gateway Domain* (GatD). De forma análoga, poderá ser classificado o *Application Domain* (AppD), como o domínio onde os dados dos dispositivos M2M são armazenados em massa, ou são directamente acedidos pelos utilizadores. Ambos os domínios estão fortemente ligados com o objectivo de disponibilizar aplicações M2M a clientes desses mesmos serviços.

O DevD contém as redes, preferencialmente de potência e consumo reduzido, onde os dados são gerados por milhares de dispositivos embutidos nos ambientes. Portanto, engloba-se na rede capilar. O ETSI exemplifica dois tipos de acesso ao DevD:

- **Ligação directa entre o NetD e o DevD através da rede de acesso** - Procedimentos como registo, autenticação, autorização e gestão são realizados no DevD, além de disponibilizar outros serviços a outros dispositivos "escondidos" da NetD.
- **Ligação indirecta do DevD com o NetD através de uma *gateway*** - ConFigurado como uma *proxy* de rede. Neste caso, todos os procedimentos exigidos são desencadeados pelo GatD. A *gateway* é diversificada na medida em que recolhe e trata informação/dados de vários tipos.

As capacidades de acesso à rede são definidas no NetD e permitem que o DevD e o GatD (se existir) comuniquem com a rede de suporte central (*core network*). Aqui estão englobas as redes de um sistema celular. A rede de acesso é vista como uma camada superior à rede capilar constituída pelos *domain devices*. É nesta camada que os dados são geridos, do ponto de vista das entidades controladoras, e partilhados com os utilizadores finais.

As redes de acesso sugeridas são normalmente as redes móveis 3G e 4G. A rede central participa activamente com ligações IP, interligação com redes heterogéneas, técnicas de *roaming* e outros serviços habituais das redes móveis. A Figura 2.8 apresenta esquematicamente esta arquitectura sugerida pelo IETF.

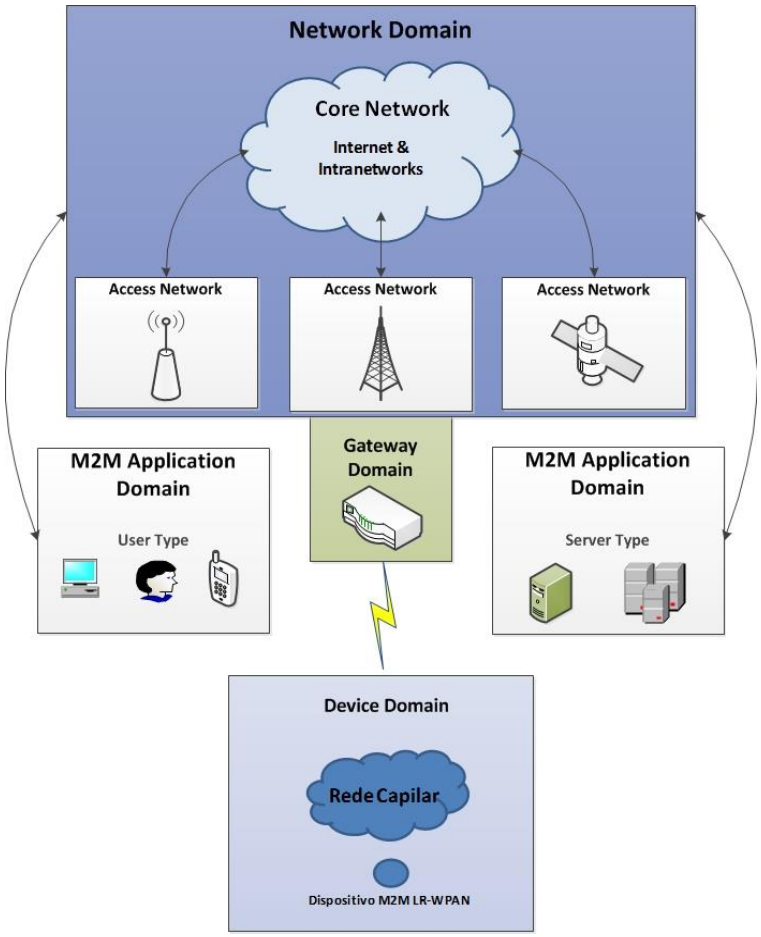

Figura 2.8 - Arquitectura M2M IETF.

Outros projectos tentam dar igualmente resposta a uma arquitectura normalizada, escalável e aceitável, nomeadamente as propostas IEEE 802.16p [IEEEM2M] e as propostas da 3GPP (que inclui o LTE como suporte base das aplicações M2M ao nível da rede de acesso). No *Release 10*, a 3GPP publicou os documentos TS 22.368 e TR 23.888, os quais apresentam e descrevem vários cenários possíveis para os serviços M2M [JPL10]. Os três cenários considerados pela 3GPP são representados nas Figuras 2.9, 2.10 e 2.11. O cenário (1) coloca o servidor M2M dentro do domínio do operador. No cenário (2) o servidor está fora do domínio do operador. O cenário (3) exemplifica a comunicação directa entre dispositivos M2M sem servidores intermédios.

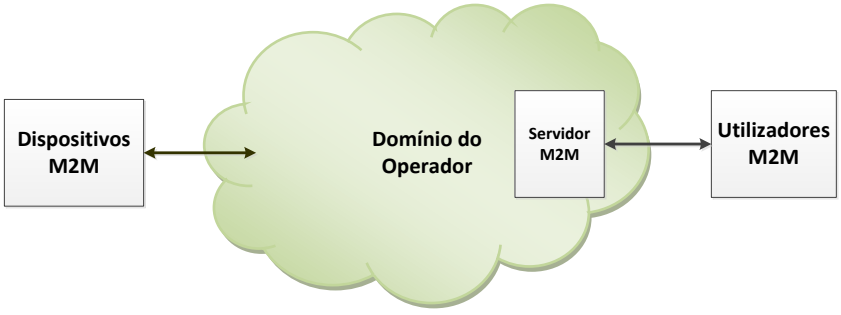

Figura 2.9 - Cenário (1).

Figura 2.10 - Cenário (2).

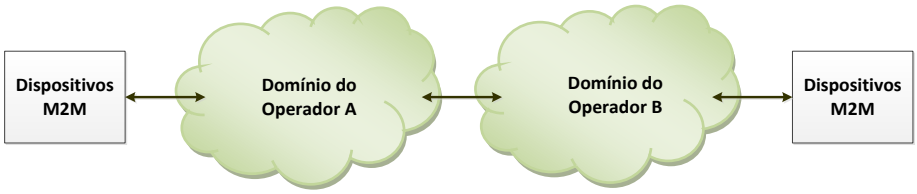

Figura 2.11 - Cenário (3).

2.2.2 Rede Capilar

Os dispositivos inteligentes embutidos no ambiente formam a rede capilar. As componentes destes dispositivos podem variar fortemente, dependendo do contexto em que se inserem. No entanto, a arquitectura é basicamente semelhante, bem como os requisitos para o seu funcionamento. Os requisitos são:

- Comunicação de curto alcance para dispositivos sem fios (*wireless*):
  - Gama de frequências *Industrial, Scientific and Medical* (ISM);
  - Redes sem fios de área pessoal e área corporal com ritmos de transferência reduzidos – *Low-Rate Wireless Personal/Body Area Network* (LR-WPAN/WBAN);
  - Capacidade para superar interferências;
  - Ligações intermitentes e com diferentes dinâmicas entre dispositivos.
- Consumos reduzidos de energia:
  - Componentes *hardware* de potência reduzida;
  - Eficiência energética dos protocolos;
  - Capacidade de recolha de energia do ambiente;
  - Transição entre estados de baixo consumo de energia.
- Baixo custo de desenvolvimento:
  - Capacidades de processamento e comunicação reduzidas mas eficientemente focadas em objectivos específicos;
  - Tamanho e custo reduzido das plataformas *hardware*;
  - Altamente auto-sustentáveis e auto-organizáveis;
  - Elevada autonomia e longevidade;
  - Elevada escalabilidade.

Espera-se que existam milhões de dispositivos, o que só será economicamente viável se existir um custo de desenvolvimento reduzido. A gestão dos mesmos torna-se insustentável se não existirem boas soluções que ofereçam autonomia e longevidade elevada.

A visão sobre a implementação M2M nesta dissertação descreve as tecnologias das redes móveis como sendo as principais candidatas para a concepção da espinha dorsal (*backbone*) principal de qualquer aplicação M2M, eficiente, sustentável e suficientemente robusta para partilhar milhões de bytes de informação, gerados pelos ambientes inteligentes. A rede capilar por sua vez, devido às suas características intrínsecas, necessita de tecnologias menos dispendiosas e mesmo assim eficientes do ponto de vista energético, espectral e económico. Inúmeras tecnologias e normas de comunicação englobam e especificam as características das redes LR-WPAN, necessárias à construção de redes capilares sustentáveis. Dependendo da aplicação desejada, podem ser sugeridas diferentes tecnologias com requisitos bastante semelhantes [HBE12].

- ***PowerLine Communication*** - As *PowerLine Communications* (PLCs) são uma tecnologia de comunicação cablada. Toda a comunicação é realizada através da rede de distribuição eléctrica, ao contrário das redes cabladas tradicionais. Esta forma inovadora de transmissão de dados enfrenta fortes desafios técnicos, pois utiliza um meio não concebido para a transmissão de dados. No entanto, para aplicações *smart grids*, esta tecnologia é considerada a melhor candidata [HBE12].

  Apesar das limitações iniciais das frequências suportadas pela norma CENELEC (entre 3 kHz e 148.5 kHz) com *bit rates* limitados com modulação *Frequency-shift keying* (FSK), o recente aparecimento de melhores técnicas de modulação (por exemplo, o método *Orthogonal Frequency-Division Multiplexing* (OFDM)) permite melhorar a fiabilidade das comunicações e a taxa de transferência, evoluindo para uma tecnologia de banda larga. Com a integração de pilhas protocolares, as aplicações de vídeo, o acesso à Internet e as aplicações inteligentes são possíveis através de PLC.

  As alianças industriais Homeplug e Homegrid possuem vários *white papers* que caminham para comunicações *powerline* capazes de suportar ritmos de transferências elevados, utilizando OFDM em bandas de frequências superiores a 2 MHz. Algumas tecnologias desenvolvidas para comunicações PLC atingem ritmos de transmissão de 400-600 Mb/s, como por exemplo o *HomePlug AV2* (HP AV2) em conjunto com a norma IEEE 1901.

  O consumo energético dos *modems* desenvolvidos com esta tecnologia é reduzido quando comparado com equipamentos eléctricos como ar-condicionado ou máquinas de lavar. Um *modem* de 200 Mb/s pode, por exemplo, consumir entre 4 W e 6 W. No entanto, com a larga adopção destes dispositivos em edifícios, nas casas e na

indústria, espera-se que este consumo adicional de energia tenha impactos significativos na energia produzida e consumida mundialmente.

Outro desafio são as perturbações das ligações devido às características dos equipamentos elétricos ligados à grelha que introduzem ruído e absorvem sinal. Por esta razão, esta tecnologia é reconhecida como sendo uma rede de comunicação com muitas perdas e ligações assimétricas, e por isso com requisitos e restrições semelhantes às das redes sem fios de potência reduzida.

- ***M-Bus e Wireless M*-Bus -** Os sistemas de comunicação em série são redes cabladas de área local (LAN) que interligam dispositivos e componentes da rede. As ligações *serial bus* possuem uma topologia na qual todos os componentes são ligados através de uma linha de transmissão comum entre todos, com transmissão sequencial de bits em série. Este tipo de topologia é considerada a mais fiável para interligar contadores inteligentes de gás e electricidade. A ligação *bus* de centenas ou milhares destes dispositivos tem um elevado nível de integridade, com a característica obrigatória de ser excepcionalmente insensível a interferências externas. As mesmas atingem ritmos de transferência elevados para quantidades de informação muito reduzidas, provenientes de múltiplas fontes [MBUS].

  A tecnologia *Meter-Bus* (M-Bus) foi concebida especialmente para responder a estes requisitos. O M-Bus não é, na sua essência, uma rede. Não necessita das camadas de transporte e sessão do modelo OSI. Utiliza um sistema hierárquico onde toda a comunicação é controlada por um módulo lógico de atribuição central, conhecido como nó mestre. Todos os dispositivos (contadores inteligentes) são nós "escravos", ligados paralelamente ao meio de transmissão através de um *Plug*. O meio de comunicação é composto por dois fios condutores. A norma especifíca *baud rates* entre 300 e 9600 Baud e o número máximo de 250 "escravos" por ligação. Estas características restringem a distância máxima entre um nó escravo e um nó repetidor para apenas 350 m.

  O *Wireless M-Bus* está especificado na norma EN13757-4:2005 como sendo a versão sem fios do M-Bus. A frequência de operação é a banda ISM 868 MHz. Basicamente a comunicação é realizada entre o contador e um ponto de agregação de dados. Os nós possuem modos de operação como o modo SLEEP (excepto para *duty cycles* de transmissão reduzidos) de forma a optimizar o tempo de vida da bateria dos dispositivos.

- **IEEE 802.11 -** A norma 802.11 (Wi-Fi) [IEEE80211] teve um enorme sucesso na área das redes de área local sem fios. O Wi-Fi utiliza 14 canais diferentes com as técnicas de modulação *Direct-Sequence Spread Spectrum* (DSSS), *Complementary Code Keying*

(CCK) ou OFDM (para 802.11g/n). A largura de banda de cada canal é 20 MHz, sendo a separação entre cada um apenas 5 MHz. Isto significa que, por exemplo, tendo como referência o canal 1 (Tabela 2.2), os quatro canais adjacentes seguintes sobrepõem-se ao primeiro, causando interferência de sinais. Este facto leva à utilização comum de apenas três conjuntos de canais de cada vez (1, 6 e 11, ou 2, 7 e 12, ou 3, 8 e 13, ou 4/5, 9/10 e 14, se a utilização deste último for permitida), com um total de cinco combinações possíveis. Estas combinações não são fixas. Quanto mais afastados estiverem os canais em utilização, maior a garantia de que não ocorrerão interferências, dado que alguma energia pode eventualmente espalhar-se para lá da largura de banda assumida de 22 MHz.

Tabela 2.2 - Canais e banda de frequência para IEEE 802.11.

Canal	Frequência Baixa (MHz)	Frequência Média (MHz)	Frequência Alta (MHz)
1	2401	2412	2423
2	2404	2417	2428
3	2411	2422	2433
4	2416	2427	2438
5	2421	2432	2443
6	2426	2437	2448
7	2431	2442	2453
8	2436	2447	2458
9	2441	2452	2463
10	2451	2457	2468
11	2451	2462	2473
12	2456	2467	2478
13	2461	2472	2483
14	2473	2484	2495

Tipicamente, as potências de emissão estão entre os 14 e 18 dBm. Para as normas IEEE 802.11b e 802.11n, a sensibilidade de recepção encontra-se no intervalo [-88 dBm, -82 dBm] e no intervalo [-88 dBm, -68 dBm], enquanto que o ritmo de transmissão máximo é igual a 11 Mb/s e 54 Mb/s, respectivamente. O tamanho máximo dos pacotes é restricto a 1500 bytes.

Alguns trabalhos foram desenvolvidos para transportar a tecnologia Wi-Fi para as redes de sensores sem fios [MYM11], [LXKK11]. Para respeitar os requisitos sobre o consumo de energia e complexidade destas redes, foram concebidos módulos de comunicação Wi-Fi de potência reduzida para aplicações de monitorização.

Estes módulos Wi-Fi (por exemplo, o WSN802G [WSN802G]) integram a pilha protocolar TCP/IP, possuem ritmos de transferência entre 1 Mb/s e 11 Mb/s e operam na banda de frequência 2.4 GHz. Estes módulos são no entanto bastante recentes. Por isso, são necessários mais desenvolvimentos concretos de plataformas *hardware* e protocolos MAC para redes de sensores sem fios baseadas na tecnologia Wi-Fi. Prevê-se que no futuro apareçam novas soluções.

- **IEEE 802.15.1 -** A norma 802.15.1 [IEEE802151], referente ao Bluetooth, ocupa no total 79 canais diferentes, não sobrepostos, na gama de frequência entre 2400 e 2480 MHz. Cada canal tem 1 MHz de largura de banda e estão espaçados por 2 MHz (*lower guard band*) e 3.5 MHz (*upper guard band*). A técnica de modulação utilizada é *Frequency-Hopping Spread Spectrum* (FHSS), evitando interferências com outras redes e dispositivos ao ir trocando de canal de comunicação ao longo do tempo. O modo *Adaptive Frequency Hopping* (AFH) foi desenvolvido para melhorar a eficiência das comunicações, exercendo saltos entre frequências (*frequency hopping*) se detectar que o canal em utilização está ocupado.

  As potências de emissão estão dependentes das aplicações e das distâncias pretendidas. Visto isto, consideram-se três classes quanto a esta matéria. A classe 1 atinge uma potência máxima de 20 dBm (100 metros); a classe 2 uma potência máxima de 6 dBm (10 metros); e a classe 3 uma potência máxima de 0 dBm (10 cm).

- **IEEE 802.15.3 -** A norma 802.15.3 [IEEE802153] permite atingir ritmos de transferência elevados em WPAN para serviços com elevados níveis de QoS, como vídeo e música em tempo real. A norma é caracterizada como sendo a escolha ideal para uma rede multimédia doméstica sem fios, ou aplicações de videovigilância baseadas na arquitectura M2M. A norma opera na banda 2.4 GHz com um *data rate* mínimo de 11 Mb/s para o esquema de modulação digital *Quadrature Phase-Shift Keying* (QPSK), e um *data rate* máximo de 55 Mb/s para o esquema de modulação *Quadrature Amplitude Modulation 64* (64-QAM). Tem 5 canais atribuídos, com largura de banda igual a 15 MHz, 3 dos quais em coexistência com a norma IEEE 802.11b.

- **IEEE 802.15.4** – Em 2003 o *Institute of Electrical and Electronics Engineers* (IEEE) definiu a norma IEEE 802.15.4 [IEEE802154]. A norma IEEE 802.15.4 define um conjunto de regras, restrições e técnicas protocolares em redes de comunicação sem fios para a camada física e camada de ligação de dados de dispositivos LR-WPAN. Esta norma serviu de base para o desenvolvimento das RSSF (cujo desenvolvimento provinha dos anos 90). A principal vantagem na utilização do IEEE 802.15.4 em redes capilares é precisamente o gigantesco número de trabalhos relacionados com a norma, além da mesma englobar todas as características necessárias para a criação de redes capilares. Assim, as RSSF, em conjunto com a norma IEEE 802.15.4, são

candidatas importantes para participarem na rede capilar. A norma é descrita no Capítulo 3.

Para que seja possível a integração destas redes com a Internet, foram propostos protocolos coerentes ao nível das camadas de rede e aplicação. Estes protocolos são brevemente descritos a seguir.

- **6LoWPAN -** O 6LoWPAN [RFC4944], proposto pelo IETF, é um protocolo de adaptação implementado entre as camadas física/ligação de dados e a camada de rede para tecnologias e normas previamente existentes como o PLC e o IEEE 802.15.4. O 6LoWPAN permite a atribuição de endereços lógicos IPv6 em redes de potência reduzida e ritmos de transferência reduzidos. Este protocolo de adaptação realiza a compressão do cabeçalho IPv6 de forma a diminuir o *overhead* do mesmo. Através do 6LoWPAN, a atribuição do IPv6 torna-se globalmente possível, e consequentemente, é possível fazer a interligação dos dispositivos IoT/M2M com a Internet. As suas características são aprofundadas no Capítulo 5.

- **RPL -** O IETF *Routing Over Low-power and Lossy Networks* (ROLL) *Working Group* propôs o *Routing Protocol for Low power and Lossy networks* (RPL) [RFC6550] como um protocolo de encaminhamento eficientemente energético para as LR-WPAN, capaz de suportar ritmos de transferência reduzidos mesmo com ligações instáveis e assimétricas. O RPL é implementado na camada de rede, desenhado especialmente para funcionar em conjunto com o 6LoWPAN. O protocolo é detalhadamente descrito no Capítulo 5.

- **CoAP -** O *Constrained Application Protocol* (CoAP) [DRAFT18], proposto pelo IETF *Constrained RESTful Environment* (CoRE) *Working Group*, é um protocolo da camada de aplicação que permite o acesso aos recursos físicos na WoT. Basicamente, este protocolo reutiliza a arquitectura REST, aproveitando todas as suas características e implementando as mesmas funções do HTTP, isto é, é um protocolo de transferência aplicacional dos recursos *web*. A diferença é que o CoAP permite a execução das funções HTTP em redes com recursos muito limitados, reduzindo o *overhead* original do HTTP ao comprimir o cabeçalho da camada de aplicação para apenas 4 bytes (excluindo os campos opcionais, que ocupam 4 bits cada um). O CoAP utiliza os quatro métodos tradicionalmente utilizados pelo HTTP: GET, PUT, POST e DELETE. O CoAP reutiliza o esquema URI para identificar os recursos físicos. O esquema URI do CoAP é algo como "*coap://site*".

2.2.3 Rede Celular

As redes celulares surgem como a melhor solução para a rede de acesso das aplicações IoT/M2M e, portanto, para a partilha dos dados provenientes das redes capilares entre servidores, *data centres* ou directamente entre utilizadores finais. A motivação para a utilização destas redes parte do facto do tráfego de dados anual cresce de ano para ano de forma vertiginosa. Espera-se que no ano 2015 o tráfego global supere a barreira dos 40 Exabytes. Com a inclusão das redes capilares, a previsão é que o tráfego anual em 2020 seja acima dos 120 Exabytes. Para que este aumento de tráfego seja possível, são necessárias tecnologias capazes de suportar este crescimento.

Por esta razão, as redes móveis integradas com as redes cabladas, são o principal suporte para a rede de espinha dorsal. A evolução das redes móveis é apresentada na Figura 2.12.

As vantagens das redes móveis vão para além da capacidade de suportar o crescimento do tráfego anual. Estas oferecem cada vez melhores coberturas, ritmos de transferência e rendimentos binários. Têm as características necessárias para atingir a ubiquidade exigida, tal como suporte para mobilidade elevada, técnicas optimizadas de gestão de interferência, suporte para *roaming* entre diferentes redes e infraestruturas consistentes. A evolução destas tecnologias e dos equipamentos dos utilizadores finais possibilita a inteira adopção das redes celulares actuais como espinha dorsal da IoT. O número de clientes das redes móveis em 2013 é 6.8 mil milhões, mundialmente. Além destes factores, a plena reutilização das infraestruturas criadas para estas tecnologias torna-se mais uma fortíssima motivação.

As actividades M2M estão a ser desencadeadas não só pelo ETSI mas também pelos grupos 3GPP e IEEE. Estas actividades são exigentes e enfrentam vários desafios. Muitas das características actuais das redes móveis não são compatíveis com as exigências das redes M2M no futuro. A Tabela 2.3 apresenta algumas das diferenças que terão de ser necessariamente resolvidas para existir compatibilidade entre as tecnologias móveis e as comunicações M2M.

A 3GPP tem vindo a identificar quais as características e melhorias necessárias para a integração das aplicações M2M nas suas tecnologias 4G.

Os documentos 3GPP, nomeadamente o TR 22.868, TR 33.812, TS 22.368 e TR 23.888, especificam os requisitos e as melhorias que necessitam de ser realizadas para suportar eficientemente as aplicações M2M nas redes móveis actuais e futuras. Paralelamente, estão a ser realizados outros estudos para o mesmo fim, bem como outras propostas de melhoria.

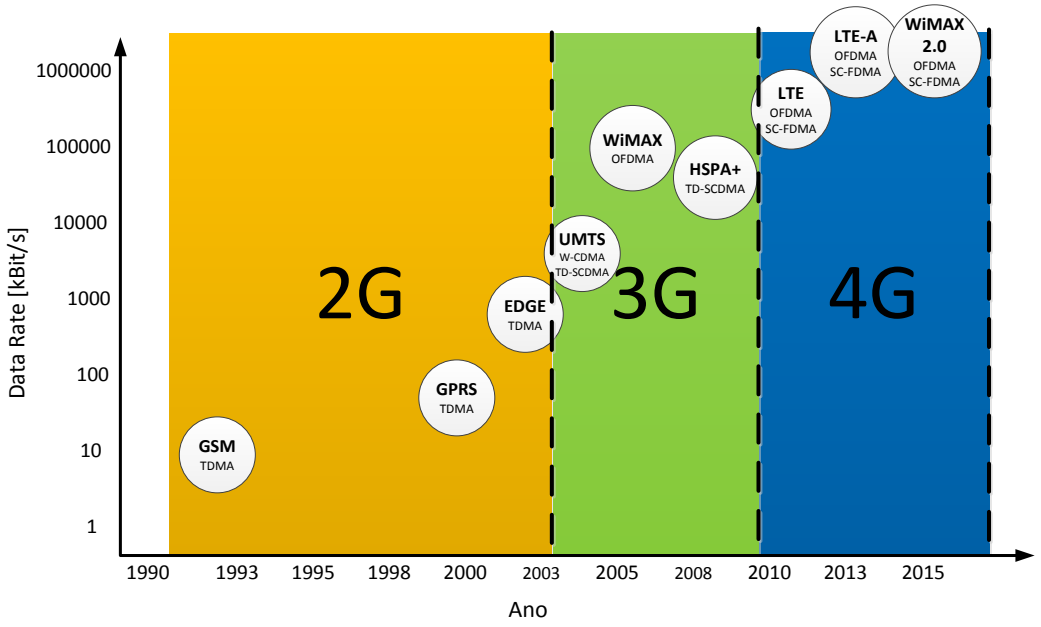

Figura 2.12 - Evolução das Redes Móveis.

Tabela 2.3 - Comparação das características actuais e M2M em redes móveis.

Características actuais das redes móveis	Características M2M
Poucos utilizadores	Muitos nós
Elevada utilização das ligações descendentes	Elevada utilização das ligações ascendentes
Dispositivos recarregáveis diariamente pelo utilizador	Dispositivos com elevada autonomia e tempo de vida útil
Gestão Centralizada (Operadoras)	Inclusão da Gestão Descentralizada (*Privacy by Design*)
Aplicações toleráveis a atrasos, incluindo ligações por voz	Grande parte das aplicações intolerantes a atrasos
Reduzida necessidade de segurança	Mecanismos de segurança autómatos
Tráfegos com características semelhantes	Coexistência de tráfego altamente diversificado e eficiente gestão de interferência co-canal
Pacotes de dados com comprimentos elevados	Pacotes de dados com comprimentos reduzidos mas em rajada

De todas melhorias que são necessárias para tornar mais eficiente a utilização dos recursos de rádio em M2M, provavelmente os dois maiores desafios identificados são a falta de eficiência que as redes 4G possuem para tráfegos de pacotes de dados pequenos, e a previsível congestão do tráfego proveniente dos milhares de dispositivos M2M. Por exemplo, para o primeiro desafio os sistemas e dispositivos LTE deverão suportar taxas de 118 kb/s em canais descendentes e 59 kb/s em canais ascendentes como suportam os dispositivos E-GPRS *multi-*

slot class 2 [B12]. Simultaneamente, o impacto deste tráfego de pacotes de dados pequenos em rajada não deverá degradar o tráfego actual, proveniente dos equipamentos dos utilizadores (por exemplo, *smartphones* LTE). Esta coexistência tem de ser conseguida, tanto no âmbito do tráfego, como na utilização dos mesmos portadores de frequência.

O segundo desafio, o congestionamento da rede, existe devido à forte concorrência entre os dispositivos que tentam aceder simultaneamente à rede de acesso, levando a picos de tráfego de dados e sinalização que penalizam os dispositivos tradicionais das redes móveis [A11]. Do mesmo ponto de vista, este congestionamento futuro do tráfego ocorre devido ao elevado número de dispositivos M2M que gera informação destinada a servidores no mesmo período de tempo. Três tipos de congestionamento de tráfego são considerados: congestionamento *Radio Network*, congestionamento *Core Network* e congestionamento *Signaling Network* [JPL10]. O primeiro tipo de congestionamento ocorre principalmente nas estações base (por exemplo, eNodeB para LTE) quando muitos dispositivos estão ligados à mesma estação base e utilizam consequentemente os mesmos canais. O segundo tipo de congestionamento ocorre nas diferentes entidades da rede responsáveis por gerir e transportar o tráfego. Um número muito elevado de dispositivos ligados à rede celular corresponde à utilização de muitos *bearers* por períodos de tempo reduzidos, levando a *overheads* críticos na gestão dos mesmos. O terceiro tipo de congestionamento ocorre no plano de controlo e, portanto, em toda a arquitectura da rede devido ao facto dos dispositivos gerarem continuamente sinalização, mesmo na presença de pacotes de dados pequenos.

Em [LKY11] os autores propõem dois métodos para gerir e atribuir *Random Access* (RA) *preamble* de maneira a acomodar o tráfego M2M adicional. O primeiro método consiste em dividir o conjunto disponível de RA *preambles* em dois subconjuntos distintos: um paras as comunicações tradicionais *humano-para-humano* (H2H), outro para as comunicações M2M. O segundo método divide igualmente o RA *preamble* em dois subconjuntos mas com a diferença de que um é exclusivo para comunicações H2H e o outro é partilhado para as comunicações H2H e M2M. Os resultados confirmados através de uma análise analítica indicam que abaixo de uma determinada carga RA, o método 2 (H2H) tem melhor desempenho que o método 1 (M2M/H2H), além de que um número cuidadosamente escolhido de *preambles* no conjunto partilhado do método 2 (M2M/H2H) pode representar a solução mais viável.

Os mecanismos LTE de calendarização de pacotes nos canais ascendentes não suportam eficientemente as aplicações M2M nas células LTE [GLA12]. A crescente quantidade de dispositivos M2M participantes está directamente associada à crescente carga de sinalização para realizar as calendarizações das ligações ascendentes, o que pode tornar impossível o suporte de acessos simultâneos nos canais partilhados. Em [R1072277], o grupo 3GPP sugeriu uma calendarização de pacotes genérica na qual todas as atribuições são realizadas consoante métricas de qualidade dos canais, latências e níveis de QoS. Assim, é permitido que os recursos sejam atribuídos dinamicamente segundo diferentes requisitos de QoS. Apesar de ser

um método de calendarização eficiente, os cenários M2M lidam com critérios de QoS extremamente diversificados e, portanto, o agrupamento de fontes de tráfego de diferentes classes de QoS pode originar latências devido a processamentos longos e restrições exigentes de armazenamento nas estações base [GLA12]. Duas soluções foram apresentadas para estas situações onde existem diferentes níveis de QoS:

- **Calendarização baseada em grupos -** grupos/*clusters* de dispositivos M2M são criados com um perfil de QoS associado;
- **Calendarização Semi-persistente** - proposta para lidar com tráfegos com características especiais (por exemplo, tráfego VoIP) e onde as atribuições duram longos períodos de tempo. Devido às características semelhantes entre tráfego VoIP e M2M, este tipo de calendarização é visto como um candidato com bastante potencial para os cenários M2M.

O *IEEE 802.16 Working Group* tem também vindo a desenvolver propostas para dar suporte e normalizar uma arquitectura M2M generalizada (*IEEE 802.16 Machine-to-Machine Task Group*) [IEEEM2M]. O documento [80216P] foi desenvolvido com propostas de aperfeiçoamento da camada física e MAC que possibilitem a comunicação M2M em grandes áreas de cobertura com bandas de frequência licenciadas, suporte para consumos de energia reduzidos, gestão eficiente para um número elevado de dispositivos ligados a uma estação base, suporte eficiente para comunicações em rajada de quantidades reduzidas de informação, e métodos mais eficientes de segurança/autenticação.

Essencialmente, o IEEE 802.16p descreve a criação de zonas M2M compostas por estações base e dispositivos interligados, agrupados em áreas específicas das redes de comunicação sem fios, como uma camada de suporte entre fontes de dados e as redes sem fios centrais para a partilha de informação. A informação M2M é partilhada por uma *Subscriber Station* (SS, dispositivos com funcionalidades M2M) e a rede central através das estações base. Foram propostos uma série de identificadores que identificam grupos M2M e ligações descendentes *multicast* partilhadas por dispositivos pertencentes ao mesmo serviço e ao mesmo grupo/zona. A atribuição e actualização dos serviços e dos identificadores é realizada pela camada superior, composta por estações base atribuídas a um determinado grupo de dispositivos.

As ligações são feitas a partir de negociações realizadas entre os SS e as estações base para que sejam atribuídos canais ascendentes/descendentes disponíveis por um determinado período de tempo, possibilitando a existência transmissões dedicadas. O documento especifica a atribuição de largura de banda suficiente para transmissões de tramas de dados em rajada, agrupadas numa trama principal, atribuída a um grupo específico de dispositivos.

Em [AGK11] é proposto um esquema prático de *relaying* eficientemente energético baseado na norma IEEE 802.16 para comunicações M2M. Os dispositivos M2M actuam como pontos de agregação de dados provenientes de fontes embutidas no ambiente. Estes pontos de agregação são a ponte de acesso para as redes móveis, comunicando directamente com as estações base do mesmo grupo. Os autores apresentam um exemplo baseado na atribuição IEEE 802.16p de tramas ascendentes/descendentes, considerando ligações com boa e má qualidade. Para casos onde existem ligações com qualidade reduzida entre dispositivos e estações base, os autores comprovam ganhos significativos ao nível do desempenho com o esquema proposto para dispositivos intermédios. A Figura 2.13 apresenta o mesmo exemplo, simplificado, para dispositivos M2M que actuem como *relays* de dados em rajada de outros dispositivos M2M, dentro do mesmo grupo/sub-rede M2M, com ligações de má qualidade com a estação base.

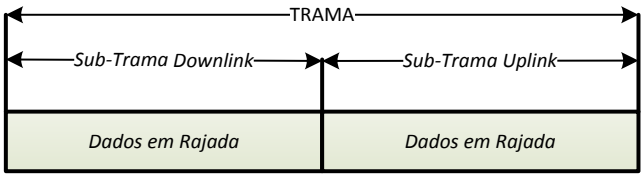

Figura 2.13 - Trama sugerida para descendentes e ascendentes.

A ligação entre diferentes redes e sistemas pode motivar o aparecimento de novas soluções e serviços. Em [MAI11] os autores propõem a inclusão de *cloud data centers* para dar resposta à procura elevada de processamento adicional devido à integração de milhares de dispositivos. Três problemas são descritos na interligação destes sistemas com os servidores da *cloud*:

- **Escalabilidade extremo-a-extremo** – A convergência de todo o tráfego gerado pode prejudicar a utilização eficiente dos recursos das redes e servidores;
- **Latência** – O tempo cíclico entre a recolha, processamento e distribuição dos dados pode ser longo demais com a inclusão dos *data centres* como centros de tratamento de dados.
- **Desperdício dos Recursos do Sistema** – Uma forma de reduzir a latência do processamento e da análise dos dados é aumentar a largura de banda e o custo das ligações da rede.

A resolução destes problemas passa pelo desenvolvimento de uma arquitectura baseada em serviços, denominada *Application Assist Network* (AAN). Esta arquitectura consiste em quatro camadas e uma entidade controladora que calcula como o sistema pode ser optimizado segundo alguns critérios. As duas camadas principais são a camada física (composta pelos dispositivos físicos do sistema) e a camada de aplicação (composta pelas aplicações do lado do servidor e do lado do cliente/dispositivos). Entre ambas situam-se a camada virtual (nós e

ligações lógicas) e a camada de rede (elementos da rede que possibilitam o processamento de dados).

2.2.4 *Gateways* em comunicações M2M

As *gateways* são elementos essenciais nas redes de comunicação M2M. As g*ateways* desenvolvidas para aplicações M2M agregam várias tecnologias de comunicação num só equipamento. Assim, é possível alcançar a convergência de várias redes de comunicação de forma bastante simples. A vantagem principal das *gateways* M2M é essencialmente a capacidade de agrupar aplicações e diferentes pilhas protocolares num só equipamento. Além disso, uma plataforma de confiança entre redes é um factor chave para a prevenção de ataques virtuais, roubo de dados privados e invasão de privacidade.

No entanto, existem inúmeras vantagens na utilização de *gateways* M2M [FSM2M], [OSGM2M]:

- Podem ser aplicadas a uma vasta variedade de aplicações distintas;
- Combinam vários serviços para criar novos serviços;
- Evita a duplicação de *gateways* ao agregar várias tecnologias num só equipamento;
- Podem converter dados provenientes das redes capilares em informação útil;
- Realizam o *bridging* entre redes de banda larga e espinha dorsal, e redes sem fios de área local, com segurança adicional;
- Permitem a convergência entre várias aplicações e serviços;
- Aumentam a escalabilidade das múltiplas redes convergidas, aliviando o congestionamento de tráfego em rajada proveniente das redes capilares para as redes celulares;
- Seleccionam dados úteis e filtram dados desnecessários;
- Realizam a gestão dos nós das redes capilares locais, assumindo a função de coordenadores das redes.

Existem várias soluções disponíveis no mercado que agregam parcialmente todas estas características. Por exemplo, a Digi apresenta alguns equipamentos com tecnologias 2G/3G, Wi-Fi, IEEE 802.15.4 e Gigabit Ethernet incorporadas num único dispositivo, possibilitando a convergência entre diferentes redes, de forma completamente transparente [DigiM2M].

Um outro exemplo, é a *gateway* Meshlium Xtreme da Libelium [MeshX]. Esta *gateway* pode conter até 5 interfaces de rede diferentes como Wi-Fi 2.4 GHz, Wi-Fi 5 GHz, 3G/GPRS, Bluetooth e ZigBee (redes de sensores sem fios). Adicionalmente pode ter agragada tecnologia GPS para aplicações móveis e veiculares. A *gateway* Meshlium permite acesso directo por *Secure Shell* (SSH), tem vários métodos de autenticação (WEP, WPA-PSK, etc) e possibilita a integração de sistemas de alimentação por recolha de energia solar.

2.3 Aplicações Reais e sua Caracterização

A IoT e as comunicações M2M possibilitam o aparecimento de um vasto número de aplicações extremamente diferentes umas das outras. Vários trabalhos descrevem estas aplicações, algumas actualmente desenvolvidas, nomeadamente [VS10], [KA12], [B06], [BGZR11], [OECD12].

- ***Automação Doméstica e de Edifícios*** **-** Este tipo de aplicações aplicam-se essencialmente para a automação de todo o funcionamento de casas e edifícios, sendo mais comuns na implementação de autonomia na ventilação, no aquecimento e na iluminação. No entanto, o conceito de IoT/M2M vai mais além, disponibilizando serviços extra ao incluir electrodomésticos, televisão e dispositivos multimédia, objectos com os quais temos contacto diariamente. A interligação destes objectos poderá informar quando um produto está em falta, onde se encontra uma chave perdida ou desligar um fogão quando não existe um utensílio em aquecimento. Algumas aplicações sucedem sem qualquer interacção humana, com capacidade de autonomia de uma forma transparente. Outras poderão fornecer dados úteis directamente aos clientes.

- ***Smart Grid/Metering*** - O conceito de *Smart Grid* está fundamentalmente ligado à eficiência energética de edifícios. Refere-se essencialmente à utilização de tecnologias de comunicação, processamento e controlo remoto nas redes eléctricas com o objectivo de optimizar a eficiência na geração e distribuição da energia. A optimização através desta tecnologia inclui a monitorização da rede com um sistema de supervisionamento semelhante ao SCADA utilizado na indústria [Fa10]. As aplicações *Smart Grid/Metering* são exemplos perfeitos de redes M2M, que futuramente podem ajudar a reduzir o consumo/desperdício de energia e manter actualizada a informação decorrente sobre o consumo de energia e outros recursos. Estes são os desafios que as *Smart Grids* pretendem ultrapassar. Adicionalmente inclui-se a optimização da distribuição de energia gerada por meios renováveis a qual sofre variações dependentes do ambiente ao contrário da geração de energia tradicional [MLK11].

  Relacionadas de forma muito próxima estão as aplicações pertencentes ao conceito de *Smart Metering*. Uma futura utilização mais ampla dos veículos eléctricos/híbridos também representa um peso adicional na rede eléctrica que é necessário avaliar o que é preciso gerir eficientemente. Aplicações reais monitorizam e fornecem informação útil sobre os consumos mais comuns em edifícios, casas e até ambientes industriais, como água, gás e electricidade. Portanto, a gestão inteligente e a informação sobre a rede elétrica é um objectivo fundamental. Os dispositivos que realizam a monitorização e partilham os dados recolhidos são comumente conhecidos como *Smart Meters*. Os primeiros dispositivos deste tipo eram os *Advanced Meter*

Reading (AMR) que recolhiam e dispunham os dados. Com a evolução do conceito, surgiu o *Advanced Meter Infrastruture* (AMI) capaz de analisar e processar os dados recolhidos, além de comunicar com outros *Smart Meters*, realizar tomadas de decisão, pedir dados e disseminar informação pelos nós vizinhos. O *Plogg* é um *Smart Meter* exemplar criado pela ByteSnap Design para aplicações ao nível do consumo de energia de dispositivos ligados à rede eléctrica. Possui um microcontrolador e um sistema de comunicação sem fios compatível com tecnologia ZigBee e Bluetooth. Pode ser controlado remotamente, controlando autonomamente o seu próprio consumo. Disponibiliza a informação sobre o consumo de energia dos equipamentos a ele ligados. Na sua essência, uma rede de *Ploggs* pode ser tratada como uma rede de sensores sem fios.

- ***Cidades*** - As aplicações das cidades inteligentes surgem para dar resposta a uma série de factores no modo como a sociedade cada vez mais concentrada em áreas urbanas lida diariamente com a rotina citadina. Inclui-se o Parqueamento Inteligente para monitorização e gestão de parques de estacionamento públicos, disponibilizando informação a cidadãos condutores sobre as condições presentes nos seus locais de destino. Outras aplicações maioritariamente de cariz estático, incluem a gestão da iluminação duma cidade, optimização da recolha do lixo, monitorização da poluição (poluição do ar, da água e sonora) e a monitorização de infraestruturas (pontes, edifícios, monumentos). Esta última aplicação pode estar inserida em ambientes de risco pouco relacionados com as cidades, tal como minas.

  Aplicações dinâmicas incluem a mobilidade dos habitantes das cidades. O transporte inteligente pretende otimizar os sistemas de transporte público, fazendo o *tracking* dos próprios transportes ou monitorizando a quantidade de utilizadores num dado veículo, disponibilizando mais sempre que necessário ou retirando da circulação transportes desnecessários. A gestão do congestionamento do tráfego optimiza a distribuição de veículos e até de pedestres, fornecendo informação em tempo real sobre os locais mais/menos congestionados e esperando que a mobilidade veicular seja contrabalançada consoante as necessidades e os locais de destino de cada condutor/pedestre. Neste âmbito, insere-se a telemática veicular onde existe troca de informação entre veículos e onde essa informação é disponibilizada de forma interactiva ao condutor e/ou passageiros. Esta informação centra-se nas condições rodoviárias referentes às condições climatéricas e nas melhores práticas a manter durante a condução diante de situações particulares de incidentes e eventos incomuns tais como acidentes rodoviários ou trânsito intenso.

- ***Agricultura*** - A automação e monitorização da agricultura visa facilitar as tarefas necessárias a realizar na produção e cultivo agrícola. Incluem-se sistemas de irrigação automáticos consoante os níveis concentrados de água, humidade e nutrientes,

prevenção de contaminações e controlo de microclimas em estufas, para optimizar a quantidade e a qualidade dos alimentos produzidos.

Outra aplicação que é reconhecida por alguns como necessária para sustentar a nossa capacidade de produzir alimento é o *Vertical Farming* ou Agricultura Vertical. A agricultura vertical pretende dar rápidas respostas na produção e distribuição de alimentos maioritariamente em áreas urbanas. O conceito insere a agricultura como algo possível dentro de cidades, embutida em edifícios desenhados e pré-concebidos para esse efeito e até dentro de apartamentos já construídos. O tipo de controlo e monitorização é semelhante às técnicas concebidas e utilizadas em estufas traficionais, mas com uma certa racionalização do espaço ocupado, dispondo inúmeras áreas agrícolas na vertical em vez de na horizontal. Inclui-se além do controlo da temperatura, humidade e luminosidade, a detecção de elevadas concentrações de poluição e renovação da qualidade do ar. A implementação destes campos agrícolas verticais dentro de áreas urbanas sugere igualmente uma forte eficiência na distribuição dos alimentos produzidos em edifícios vizinhos.

- ***Saúde -*** A saúde é uma das áreas de interesse com maior potencial para a realização de ambientes inteligentes interactivos e ubíquos atraindo o interesse da maioria das pessoas. Estão fundamentalmente inseridos em redes *Wireless Body Area Network* (WBAN), em aplicações de monitorização corporal tal como pressão arterial, níveis de açúcar e batimentos cardíacos. Aplicações parcialmente distintas podem ser agregadas como por exemplo a actividade social e os movimentos de um paciente idoso e/ou com capacidades reduzidas que habite sozinho na sua residência. A monitorização em tempo-real dos pacientes pretende, como na maioria das aplicações em ambientes inteligentes, armazenar e disseminar os dados relevantes dos pacientes para centros de apoio e hospitalares, mantendo históricos médicos e acompanhando o estado de saúde diário de cada pessoa. Este conceito é comumente denominado *Personal Health Record* (PHR) e permite essencialmente que tanto o paciente como o seu médico tenham acesso aos parâmetros actualizados da monitorização. Qualquer irregularidade detectada poderá ser alertada no próprio momento, bem como seja mantida e visualizada a informação por ambas as partes.

  Este novo conceito de saúde informatizada é denominado por *e-Health*. Inclui tanto a parte do processamento electrónico como as próprias comunicações realizadas por diferentes redes de maneira a dar suporte a estas aplicações. Incluído está o conceito *m-Health* que pormenoriza a utilização de dispositivos de redes móveis para ter acesso e visualizar a informação, incluindo *smart phones*, PDAs e *tablets*. Este é puramente um conceito M2M.

  A telemedicina é um ramo do *e-Health* e *m-Health*. A telemedicina visa derrubar barreiras como a distância entre pacientes e médicos/hospitais, e garantir o acesso às necessidades medicinais de lugares isolados/distanciados ou em situações de

emergência. A telemedicina foca-se mais em aplicações de videochamadas para consultas e/ou diagnósticos médicos que aproveitam a informação recolhida localmente no paciente. Outro exemplo de aplicações mais avançadas é a telecirurgia que integra elementos de robótica e tecnologias de comunicações avançadas com elevadas capacidades de transferência de dados, ligações seguras e praticamente a inexistência de latências e erros nas comunicações.

- ***Indústria*** - A indústria foi sem dúvida pioneira no sentido de criar sistemas autónomos e de supervisão/monitorização de processos. O melhor exemplo são os sistemas SCADA. A intensa competitividade industrial e consequente demanda por processos mais eficientes impulsionou a criação destes sistemas autossuficientes e monitorizáveis. Nesse âmbito, a natureza das redes sem fios de monitorização (ou seja, redes de sensores sem fios) abarcam várias vantagens sobre o controlo de sistemas industrias através de redes cabladas, tal como auto-organização, fácil implementação/manutenção, flexibilidade e actualmente uma elevada capacidade de processamento [GH09]. Estas características oferecem tempo de resposta melhorado a eventos proporcionando o desencadear das acções mais apropriadas em tempo real. No entanto, aplicações em ambientes industriais levantam alguns desafios e problemas. Primeiro, as condições severas e rigorosas presentes em fábricas. Os nós das redes estarão sujeitos a elevados níveis de humidade, temperatura, vibração, poeira, e outras condições mais corrosivas. Segundo, a elevada interferência de radiofrequência e ruído. Terceiro, segurança contra ataques e espionagem. Durante muito tempo a indústria reconhecia que a tecnologia e normas actuais não respondiam aos requisitos das suas aplicações de monitorização. No entanto, nos últimos anos têm vindo a ser definidas duas normas independentes, e em paralelo, que respondem aos desafios da industria: *WirelessHart* e *ISA100.11a* [PC11]. Ambas as normas estão implementadas em conjunto com as actualizações 2011/2012 da norma IEEE 802.15.4. As aplicações industriais mais gerais são a monitorização de falhas em equipamentos, optimização de processos (redução da intervenção humana), identificação de deficiências em linhas de produção, aumento da segurança dos empregados e geração de relatórios relativamente às actividades e produções diárias [SMZ07].

2.4 Sumário e Conclusões

A IoT pode ser descrita de várias maneiras e em vários sentidos. O seu potencial é imenso pois cobre uma vasta diversidade de aplicações, alvo de investigação, e que ganham cada vez mais a atenção do público. Consequentemente, novas áreas de negócio continuam a aparecer e a crescer neste âmbito.

Neste Capítulo foram abordados os vários conceitos que dão forma à Internet tradicional e à IoT. Protocolos e arquitecturas que tornam possível a existência da "Internet das Pessoas", são reaproveitados para construir a "Internet das Coisas". Nomeadamente, o protocolo IP que possibilitará a integração dos objectos do dia-a-dia com a Internet ao fornecer-lhes um endereço (identidade), e a arquitectura REST que tornará possível localizá-los na web como serviços e recursos físicos, como se se tratassem de serviços virtuais, hoje disponíveis na actual Web 2.0.

Por outro lado, as comunicações M2M inserem-se neste âmbito em aplicações onde objectos e todo o tipo de dispositivos comunicam e trocam informação entre si, em aplicações de controlo e automação sem qualquer intervenção humana. Juntas, as comunicações M2M e a IoT, criam um paradigma novo que pouco a pouco revolucionam a forma como o brutal crescimento de informação disponível é tratado. Enquanto as comunicações M2M elevam a capacidade de como os objectos se auto-organizam e auto-informam dos seus estados actuais e futuros, a IoT disponibilizará a infraestrutura necessária para essa troca constante e omnipresente de informação, disponibilizando-a tanto entre os objectos, como para as pessoas.

A estratégia adoptada para este novo paradigma foca-se na reutilização das tecnologias já existentes, desde as tecnologias de comunicação celulares, até às tecnologias capilares. As tecnologias que compõe as comunicações capilares inserem-se nos próprios objectos e dispositivos embutidos nos ambientes, para realizem a aquisição de dados e transmissão dos mesmos para pontos de acesso integrados na IoT. Depois dos dados estarem presentes na espinha dorsal das redes de comunicação, são tratados pelas comunicações celulares devido às suas importantes características de "omnipresença", larga adopção pelas sociedades humanas, velocidades de transmissão e eficiência elevada, tentando alcançar a sensação de estarem *"always online"*.

No entanto, necessitam de ser ultrapassados alguns desafios importantes. Apesar deste novo paradigma da informação conter dados de dimensões reduzidas, são em quantidades excecionalmente elevadas, levando a comunicações e transmissões em rajada de imensos dados. Além disso, prevê-se que sejam milhões de dispositivos a comunicarem estes dados em rajada, conduzindo a elevados níveis de congestionamento da rede. Por isso, é necessário uma eficiência espectral mais elevada, visto que não se pretende degradar os serviços celulares actualmente disponíveis. Englobados nestes desafios estão outros, como a diminuição da latência extremo-a-extremo entre dispositivos e utilizadores, melhoria da cobertura das ligações sem fios, gestão mais eficiente das interferências, elevada fiabilidade em conjunto com o elevado grau de mobilidade para aplicações móveis, e convergência entre redes completamente heterogéneas, alcançando a aclamada computação ubíqua. Alguns grupos de trabalho focam-se cada vez mais na normalização de técnicas e tecnologias que respondam aos desafios descritos, nomeadamente a 3GPP e o IEEE.

Capítulo 3

Adopção de uma Rede Capilar

As redes de sensores sem fios (RSSF) são um tipo de LR-WPAN com as características desejáveis de uma rede capilar [RL09] [P06]. Classificam-se como redes embutidas cuja maioria dos dispositivos suporta sensores e/ou actuadores, com capacidade de processamento e comunicação com consumos de energia muito reduzidos.

O grande número de trabalhos relacionados com as RSSF levou ao aparecimento de soluções bastante viáveis e implementações robustas. Actualmente existem oportunidades vastas no mercado para a realização de projectos de controlo/monitorização industrial, citadino, comercial, automação de edifícios e casas, *smart grids*. Inclui-se a miniaturização e aperfeiçoamento de todas as gamas de sensores. O acréscimo de capacidades *hardware* e *software* em componentes igualmente miniaturizados, proporcionou desempenhos superiores mesmo que limitados. Simultaneamente, a implementação do *software* e de camadas *middleware* tem vindo a ser facilitada através de sistemas operativos mais transparentes para este tipo de dispositivos de potência reduzida.

As plataformas *hardware* possuem vários estados de funcionamento. Os dois principais são o estado *activo*, em que todo ou parte do *hardware* está em funcionamento, e o estado SLEEP onde a maioria está desligada ou em *stand-by*. Alguns dispositivos permitem seleccionar os componentes *hardware* que se pretendem desligar durante determinados períodos de tempo. Outros simplesmente desligam alguns componentes, deixando outros sempre activos para várias funções. Esta é uma característica extremamente importante pois os nós sensores são alimentados por baterias. Quanto mais componentes desligarem durante mais tempo e intermitentemente, menores serão os consumos. No entanto, maior será a probabilidade de ocorrem faltas e erros nas comunicações.

Quanto menor for o consumo energético maior é o tempo de vida de um nó sensor. Isto só será válido se o nó não perder pontos ao nível do desempenho. Se a capacidade de processamento for fraca, mais dados ficarão por ser tratados. Se a comunicação não for eficiente, existirão atrasos, corrupção de dados e retransmissões. Se um nó não tiver um mecanismo sólido de transição entre estados e permanecer muito tempo no estado de SLEEP, poderá não ser capaz de capturar eventos importantes e falhar o sincronismo com a sua rede de comunicação. O tempo de vida da rede é uma questão fundamental quando se pretende que uma rede de sensores sem fios funcione com bom desempenho.

Para obter os resultados pretendidos, os componentes *hardware* dos nós têm de possuir consumos de energia reduzidos e níveis de desempenho aceitáveis. Estes componentes não podem ser muito complexos mas deverão ser suficientes para a maioria das aplicações. É, por isso, necessário ponderar sobre quais os aspectos mais importantes duma determinada aplicação. Cada aplicação tenderá a ter no final as suas próprias características individuais.

As aplicações em RSSF são bastante diversificadas [SMZ07]**:**

- **Baseada em Eventos** – detecção de eventos como vigilância, inundações, acidentes;
- **Baseada em Monitorização e *Tracking*** – monitorização em saúde, atmosfera, agricultura, vigilância, *tracking* de veículos e crianças;
- **Baseada em Acções** - Automação;
- **Baseada em Perigo** – Detecção e/ou monitorização de eventos que representem perigo e necessitem de ser alertados o mais rapidamente possível;
- **Baseado em Movimento** – Robótica e máquinas com mobilidade.

Os sensores são parte fundamental para que haja percetibilidade sobre o ambiente. Sensores passivos são preferíveis comparativamente a sensores activos devido aos diferentes consumos de energia entre ambos. Actualmente, os sensores passivos contribuem satisfatoriamente para a grande maioria das aplicações. Para que as aplicações propostas para as RSSF possam ser realizáveis, os nós sensores terão de ser capazes de fazer essencialmente três tipos de medições: físicas, biológicas e eventos [SMZ07]. A Tabela 3.1 relaciona os tipos de medição com as aplicações e os sensores utilizados.

Tabela 3.1 - Sensores e aplicações.

Tipos	Descrição	Aplicações	Exemplos
Medições Físicas	Capacidade para sentir vários factores sobre o ambiente.	*Tracking*; monitorização de condições físicas do ambiente, relacionados com níveis de temperatura, condições mecânicas de máquinas e infraestruturas.	Termístores, sensores de carga e força¸ GPS, *Encoder*, acústicos, vibração, acelerómetro
Medições Biológicas	Capacidade para medir níveis de concentração de substâncias e produtos.	Monitorização das condições biológicas do ambiente, objectos ou pessoas; controlo industrial, agrícola e citadino.	Sensores de gases, pressão atmosférica, humidade, corporais.
Detecção de Eventos	Capacidade para reconhecer ocorrências e eventos naturais ou humanos.	Monitorização de eventos relacionados com vigilância, acidentes, incêndios e inundações.	IR, detector de chamas e inundações, *photo resistors*, ultrassónico, imagem.

É apresentado um exemplo bastante simples e genérico sobre monitorização de temperatura. Para este exemplo foram utilizados dois módulos de comunicação XBee ligados através de microcontroladores Arduino. As Figuras 3.1 e 3.2 mostram um emissor e um receptor, respectivamente. O emissor adquire valores de temperatura a partir de um sensor LM35 e emite esses dados. O receptor por outro lado, ligado a um computador (uma estação base genérica), recebe os dados e envia-os por porta série, apresentando-os ao utilizador final através de uma simples aplicação, como mostra a Figura 3.3. Posteriormente, esta informação é armazenada numa base de dados associada à aplicação.

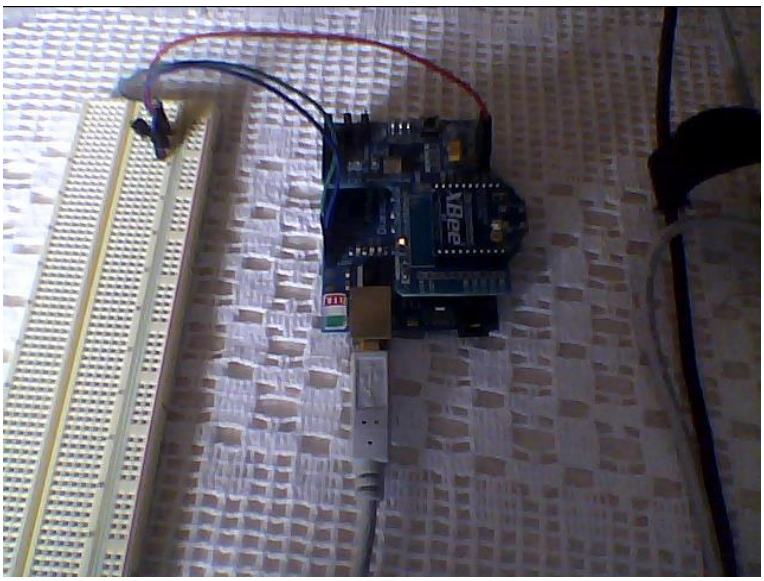

Figura 3.1 – Aquisição e emissão de dados de um nó XBee.

Figura 3.2 – Nó receptor XBee.

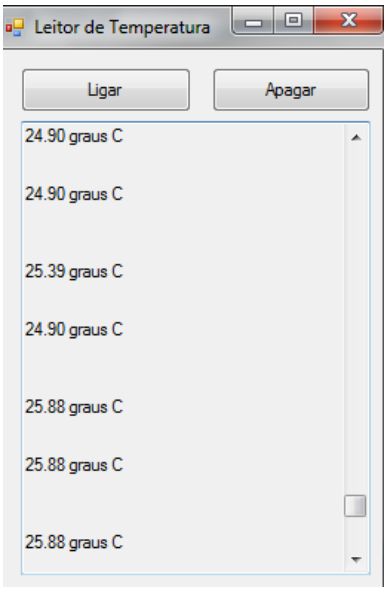

Figura 3.3 – Aplicação "Leitor de Temperatura".

Genericamente, estas redes são denominadas por *Redes de Sensores Sem Fios*, devido à maioria das aplicações serem compostas por sensores e aquisição de dados. No entanto, em ambientes inteligentes, a vertente da automação tem uma forte presença para além da monitorização e actualização de informação. Um ambiente inteligente pretende, além de tornar ubíqua e constante a actualização de informação sobre si mesmo, automatizar funções diárias, das mais básicas às mais complexas. Por essa razão, os nós também são equipados com actuadores. Na totalidade e num contexto mais íntegro dos ambientes inteligentes, estas são *Redes de Sensores e Actuadores Sem Fios*.

As RSSF estão sujeitas a falhas muito comuns e previsíveis dadas as suas características, que devem ser contornadas durante desenvolvimento da rede. Primeiro, muitas das aplicações são executadas em redes com muitos nós, levando a volumes de tráfego bastante densos, o que aumenta a degradação da QoS da rede. Além disso, devido às limitações do *hardware*, com volumes de tráfego mais elevados, isto é, maiores quantidades de dados para emitir, receber e processar, o tempo de vida de toda a rede pode ficar bastante comprometido. A pilha protocolar deve por isso ser simples mas eficiente, actualizada consoante o tipo de aplicação e equilibrando as métricas mais importantes para o desempenho esperado. De acordo com este raciocínio, os problemas mais comuns em aplicações RSSF são:

1. ***Colisões*** - Ocorrem quando existe interferência entre as ligações, ou melhor, quando vários nós comunicam ao mesmo tempo. Em redes sem fios, a informação é partilhada através da propagação electromagnética. O canal de comunicação é um meio partilhado por todos os que se encontram na mesma área de cobertura. Quando ocorrem colisões entre tramas há perdas e, portanto, a troca de dados não é concluída com êxito. Portanto, se existir um mecanismo de controlo que alerte o nó

emissor que os dados não foram entregues, o nó tem de retransmitir o sinal, consumindo mais energia para a mesma tarefa. Para que estas situações não ocorram demasiadas vezes, os nós necessitam de garantir que não existirá interferência e que não ocorrerão colisões, através de sinalização e estratégias controlo;

2. ***Idle Listening*** - Para poderem adquirir dados e transmiti-los, os nós necessitam de estar activamente ligados. No entanto, e dada a natureza da maioria das aplicações, os nós passam mais tempo sem fazer nada em concreto do que a realizar as suas funções habituais. Sendo assim, os nós permanecem ligados desnecessariamente a maior parte do tempo. A este estado denomina-se *Idle Listening* onde há desperdício de energia ao manter ligados circuitos que estão na prática, parados. Pode-se concluir que, nestes casos, o melhor é desligar esses circuitos, mantendo o nó num nível de potência ainda mais reduzido, com valores que podem rondar os nanowatts;
3. ***Overhearing*** - Quando um nó emissor comunica com outro nó receptor, outros nós dentro da mesma área de cobertura que tenham o rádio ligado recebem os mesmos sinais, mesmo que os dados não sejam direccionados para eles. O consumo de energia ao receber dados irrelevantes para esses nós é extremamente ineficiente do ponto de vista energético. Mais uma vez, conseguir desligar os circuitos desses nós, principalmente o rádio, durante essas comunicações, torna-se a melhor solução;
4. ***Overhead*** - Ocorre quando os nós trocam sinalização e tramas de controlo para evitar colisões e atrasos, ou quando se pretende aumentar a fiabilidade das comunicações. A sinalização é feita através de troca de pacotes que, mesmo com tamanhos reduzidos, deterioram a capacidade dos canais de comunicação e podem causar excessivos desperdícios de energia. O *overhead* induz muita discussão e realização de vários estudos para tentar perceber quais os melhores equilíbrios (*tradeoffs*) para diferentes aplicações. Para que as redes sem fios funcionem minimamente bem, o controlo terá sempre de existir. Poderá existir mais ou menos *overhead*, dependendo do que se pretende. Por exemplo, em casos em que os cenários são críticos, é importante que não haja latências e interferências, e, portanto, é importante haver alguma sinalização. Outras aplicações onde o atraso da entrega dos dados não é necessariamente importante, a longevidade pode ser aumentada em toda a rede, diminuindo a quantidade de sinalização imposta pelos protocolos.

3.1 Pilhas Protocolares em RSSF

A norma IEEE 802.15.4 especifica o funcionamento das duas primeiras camadas do modelo TCP/IP para LR-WPAN, a camada física e a camada de ligação de dados, mais especificamente a subcamada de *Medium Access Control* (MAC). Originalmente, as camadas superiores foram definidas pela Aliança ZigBee que se concentra prioritariamente em protocolos de encaminhamento eficientemente energéticos para redes de múltiplo salto (*multi-hop*), e para aumentar o desempenho geral dos dispositivos [F08]. A Figura 3.4 apresenta a pilha protocolar.

Transversal às camadas da pilha protocolar 802.15.4/ZigBee são descritas várias características e funcionalidades necessárias para que se atinjam os melhores desempenhos. Estas funcionalidades verticais são comumente denominadas como *Cross-Layer Functionalities* pois são partilhadas verticalmente por todas as camadas da pilha [Ram01]:

- **Gestão do Consumo de Energia** - Optimização energética dos recursos em todos os níveis;
- **Gestão da Segurança** - Vários métodos de encriptação e de prevenção a ataques;
- **Gestão da Sincronização das Comunicações** - Métodos de sincronização entre emissores e receptores na comunicação de pacotes de dados;
- **Gestão da Localização e Mobilidade** - Reconhecer a localização de um nó em relação aos seus nós vizinhos, tanto para nós estáticos como em constante movimento;
- **Gestão da topologia da rede** - Manter actualizada a topologia onde o nó se insere, tanto em relação aos nós vizinhos, como em relação ao coordenador da rede, bem como a trajectória dos encaminhamentos de pacotes.

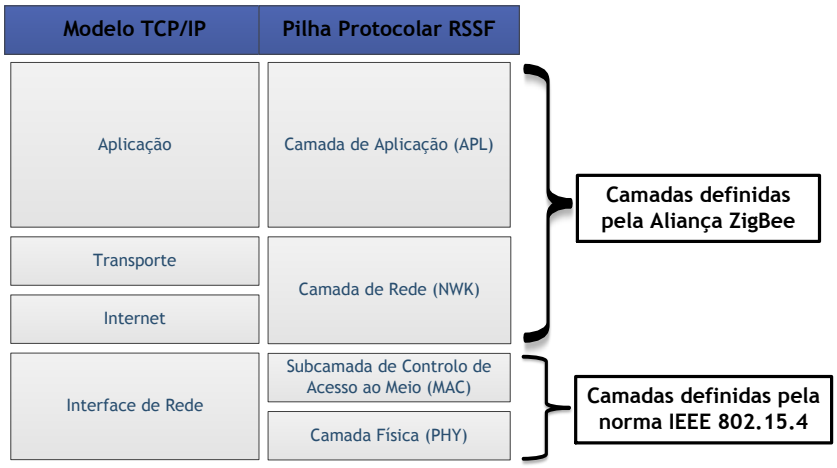

Figura 3.4 - Modelo TCP/IP e RSSF.

Qualquer optimização deverá ser realizada não só numa única camada como por toda a pilha. Da mesma maneira, uma funcionalidade sugerida para resolver os problemas relacionados com uma camada ou optimizar a mesma, deverá ser considerada nas camadas adjacentes, com os mesmos propósitos de optimização, e na resolução de problemas comuns a elas. Os autores em [MHD08] exemplificam esta ideia com o algoritmo *Cross-Layer Power Control* (CLPC), uma funcionalidade vertical cujo objectivo é gerir e balancear o consumo de energia das duas camadas mais profundas (física e MAC).

Os autores de [Ram01] defendem que a implementação do CLPC não pode ser apenas realizada ao nível dos problemas da camada física pois as camadas superiores influenciam fortemente o desempenho da primeira camada. Apesar da abstração natural das camadas e

da independência entre protocolos, alguns aspectos como, por exemplo, o tráfego da rede e os algoritmos ao nível MAC, influenciam o desempenho físico dos dispositivos.

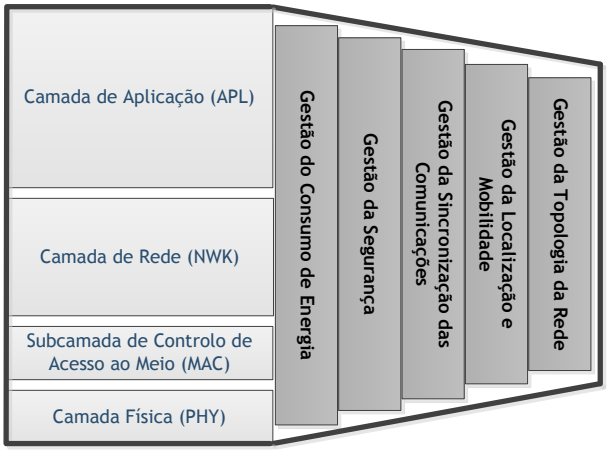

Figura 3.5 – Camadas Verticais e o conceito de *Cross-Layer Design*.

A implementação real das funcionalidades verticais encontra desafios, tal como a possível necessidade de criar interfaces, subcamadas extra e comunicação adicional de informação entre as camadas. Em [KKVBSG12], são mencionados alguns métodos para as interacções entre as camadas. A comunicação pode ser realizada directamente através da informação encapsulada em camadas específicas, acedida pelas camadas adjacentes. Outro método é a construção de uma base de dados (na prática, uma camada extra) partilhada entre as camadas que mantém disponível e actualizada a informação referente às mesmas.

A norma ZigBee está actualmente a redesenhar a sua pilha de protocolos, desde a camada de rede até à camada de aplicação, adicionando poderosos elementos que tenderão a levar as redes LR-WPAN a atingir outros patamares tal como a ligação directa à Internet através do protocolo IP. Este novo projecto, o *ZigBee Smart Energy Profile version 2.0,* visa oferecer um controlo e um funcionamento fortemente baseado em protocolos definidos pelo IETF com bons desempenhos energéticos, como o 6LoWPAN e o RPL [ZigSE]. Estes protocolos IETF da nova geração são os melhores candidatos para a inclusão de uma rede capilar na IoT. Como referenciado em [NR12], o objectivo da construção de uma pilha protocolar ZigBee que integra o protocolo IP abarca inumeros objectivos e aplicações, tal como controlar equipamentos electrónicos remotamente por IP e partilhar e/ou atribuir recursos disponíveis entre dispositivos.

O grupo IETF projectou o protocolo que permite correr IPv6 sobre a norma IEEE 802.15.4. O 6LoWPAN foi criado directamente para as tecnologias de comunicação *low-rate*/*low-power* das redes capilares. Através de técnicas de compressão, o cabeçalho dos pacotes IPv6 pode ser comprimido de 40 bytes para 4 bytes, eliminando 80 % do *overhead*. Esta redução possibilita que pacotes do tipo datagrama (UDP) não violem as tramas 802.15.4 com

comprimento máximo até 127 bytes. Quando a quantidade de dados ultrapassa este limite, o *encoding* do cabeçalho possibilita a fragmentação e o reagrupamento (*reassembly*) dos datagramas. O 6LoWPAN implementa um mecanismo de endereçamento *mesh* que, através de múltiplos saltos, que completam um único salto por IP, possibilitam a entrega de pacotes ao nó destino [CICRSY]. Os fragmentos de um pacote IP podem chegar ao nó destino através de caminhos diferentes, possibilitado pelo mecanismo de endereçamento *mesh*. Após todos os fragmentos serem entregues, o nó destino pode realizar o reagrupamento do pacote IP. A diferença do encaminhamento *mesh* e do encaminhamento *route* é que este último obriga a que os nós intermédios participem como *routers* IP, utilizando a tabela de endereços IP que possuem dos nós vizinhos. Por esta razão, o encaminhamento *route* é o encaminhamento que possibilita a interligação por IP das RSSF com a Internet e outras redes exteriores. O encaminhamento *mesh* por sua vez é realizado apenas localmente. O 6LoWPAN e ambos os encaminhamentos são descritos no Capítulo 5.

Ao nível da camada de rede, o IETF ROLL definiu o protocolo de encaminhamento RPL que organiza hierarquicamente a rede numa topologia em árvore e define os melhores encaminhamentos dentro da rede a partir de uma função de optimização, com métricas de custo, tal como o número de saltos (*hops*), débito binário (*throughput*), latência, energia residual e fiabilidade das comunicações entre os nós.

A camada de transporte reutiliza o protocolo *User Datagram Protocol* (UDP) para criar pacotes de dados. No entanto, o protocolo de transporte *Internet Control Message Protocol* (ICMP) também é considerado, exclusivamente para pacotes de controlo. As principais razões para a não utilização do *Transporte Control Protocol* (TCP) em redes muito limitadas em termos de recursos energéticos são a sua elevada complexidade e a utilização de um cabeçalho com *overhead* bastante elevado. O mecanismo de retransmissão TCP implica desperdiçar recursos em todos os saltos realizados entre o nó de origem e o nó de destino. Ao contrário deste, o protocolo UDP encapsula o mínimo de informação, facilmente comprimível (8 bytes), sem mecanismos de *handshake* DADOS/ACK (a subcamada MAC assume esta responsabilidade) e sem implementar qualquer mecanismo de controlo de fluxo.

Por fim, como descrito no Capítulo 2, ao nível da camada de aplicação, o grupo de trabalho CORE do IETF definiu o protocolo CoAP como o que se melhor insere dentro do contexto das M2M em *serviços web físicos* [BCS12]. Este protocolo de aplicação tem potencialidades importantes na medida em que permite total integração com a arquitectura REST da web ao ser capaz de traduzir o protocolo HTTP. Explicitamente, o CoAP é a versão comprimida do HTTP. Qualquer elemento de qualquer objecto e/ou dispositivo *low-power* pode ser acedido como se fosse um serviço de *browsing*, utilizando URI para encapsular e identificar esse tipo de recursos.

O ContikiOS [ConOS], descrito como o primeiro sistema operativo concebido para a IoT, inclui actualmente estes três protocolos do IETF. A aliança ZigBee está actualmente a desenvolver uma norma baseada em IPv6 [ZB02]. A pilha protocolar de RSSF baseadas em IP está representada na Figura 3.6. As funcionalidades verticais descritas permanecem as mesmas nesta pilha protocolar.

Figura 3.6 – Pilha Protocolar 802.15.4/IETF.

3.2 Arquitectura de um dispositivo de RSSF

Os componentes *hardware* principais de uma plataforma de RSSF são:

- *Sensor/Actuator Board*;
  - Sensores e actuadores;
  - Circuitos de instrumentação.
- Módulo de Comunicação;
  - Rádio Emissor/Receptor;
  - Antena;
- Microcontrolador;
  - Microprocessador;
  - Memória RAM (SRAM/SDRAM);
  - Memória EEPROM;
  - Memória Flash;
  - ADC;
- Sistema de Alimentação;
  - Pilhas, baterias e supercondensadores;
  - Regulador de Tensão;
  - Tecnologias de recolha de energia.

Ao nível do *software*, uma plataforma tem basicamente 5 subsistemas:

- *Sistema Operativo Real-Time* (RTOS);
- *Drivers* de sensores;
- Gestor dos Procedimentos das Comunicações;
- *Drivers* para camada física do módulo de comunicação;
- Aplicações de processamento de dados e informação.

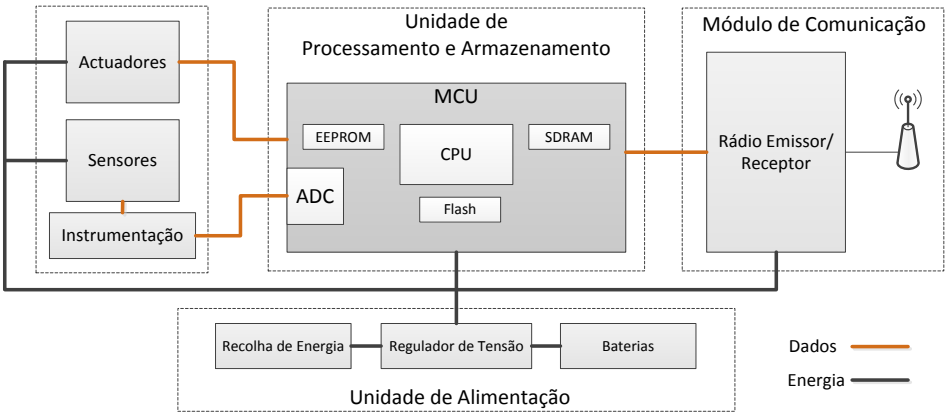

Figura 3.7 – Componentes *Hardware*.

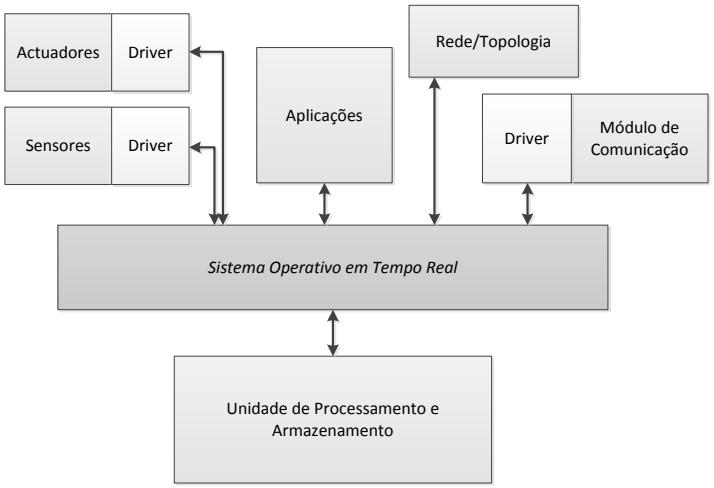

Figura 3.8 – Componentes *Software*.

3.3 IEEE 802.15.4

A primeira versão na norma IEEE 802.15.4 foi publicada em Outobro de 2003. A norma é especialmente direccionada para comunicações sem fios de curto alcance, ritmos de transmissão reduzidos, consumo de energia reduzido e complexidade reduzida. Especificamente criada para aplicações simples de sistemas embutidos, o IEEE 802.15.4 reúne as condições necessárias para arquitecturas com custos e consumos de energia extremamente

reduzidos. Em suma, a norma tenta dar resposta às necessidades e limitações que as LR-WPAN.

Após a publicação da norma em 2003, novas revisões e actualizações foram realizadas, nomeadamente o IEEE 802.15.4-2006, cujo objectivo foi indicar e remover alguns problemas surgidos das primeiras implementações em dispositivos reais, resolver ambiguidades, reduzir a excessiva complexidade da norma e melhorar o sistema de chaves de segurança. Em 2007 foi publicada uma nova reforma da norma (IEEE 802.15.4-2007) que adiciona camadas físicas alternativas para diferentes bandas de frequência. Estas camadas físicas alternativas são utilizadas consoante a região do planeta, os seus regulamentos e o tipo de aplicações. Em 2009 duas novas reformas foram realizadas com o intuito de dar resposta aos diferentes regulamentos da China e do Japão, reforma 2 e reforma 3, respectivamente. Uma nova actualização foi posteriormente publicada em 2011 (IEEE 802.15.4-2011) que remove outras ambiguidades e adiciona melhorias realizadas durante implementações reais da versão 2006. Em 2012 três novas reformas foram publicadas. As três aperfeiçoam o funcionamento das duas primeiras camadas da pilha protocolar. A reforma 1 de 2012 (IEEE 802.15.4e-2012) melhora e adiciona novas funcionalidades à camada MAC, nomeadamente a inclusão de técnicas Múltiplo-Canal, atribuição e sincronização de *slots* temporais, utilização simultânea de múltiplas *Superframes*, inclusão de um esquema *Slotframe* (alternativa à *Superframe*), novos protocolos mais robustos, entre outras. A reforma 2 (IEEE 802.15.4f-2012) define uma nova camada física para a implementação em massa de sistemas RFID, enquanto que a reforma 3 (IEEE 802.15.4g-2012) aplica novas definições.

Como referenciado anteriormente, a norma apenas tem o intuito de especificar o funcionamento das duas primeiras camadas da pilha protocolar: Física (PHY) e de Ligação de Dados (mais precisamente a subcamada MAC). As duas próximas Secções descrevem estas duas camadas.

3.3.1 Camada PHY

A camada física é a camada mais baixa de qualquer pilha protocolar, responsável por estabelecer uma interface entre os dispositivos e o meio de comunicação. Entre todas as suas funções, nomeia-se o controlo do rádio emissor/receptor, detecção e avaliação do nível de energia presente no meio (nível de interferência do meio), avaliação da qualidade do canal, modulação do sinal, selecção do canal, emissão e recepção de tramas no meio de comunicação.

Definida para operar nas bandas de frequência IMS e dependendo da localização geográfica, a camada física possui diferentes técnicas de modulação do sinal, diferentes ritmos de transferência e diferentes conjuntos de canais que podem ser utilizados. A Tabela 3.2 apresenta os diferentes tipos de operação.

Tabela 3.2 – Técnicas de modulação e canais da norma IEEE 802.15.4.

Banda de Frequência	Técnica de Modulação	*Chip Rate* [kchips/s]	*Data Rate* [kb/s]	*Symbol Rate* [ksymbols/s]	Zona Permitida	Nº Canais
868 MHz	BPSK	300	20	20	Europa	1
915 MHz	BPSK	600	40	40	América	10
2.4 GHz	O-QPSK	2000	250	62.5	Mundialmente	15

A utilização da banda 2.4 GHz possibilita, juntamente com a modulação *Offset Quadrature Phase-Shift Keying* (O-QPSK), atingir uma uma *bit rate* máxima de 250 kb/s. Mundialmente, esta é estratégia mais adoptada pelas claras vantagens.

Na modulação O-QPKS, cada byte é dividido em dois símbolos (grupos de 4 bits). Cada símbolo é mapeado para 32 *chips* em 16 sequências pseudo-aleatórias. Cada grupo de *chips* é transmitido a 2 MChip/s. Com um *chip rate* de 2000 kchips/s, o *symbol rate* é 32 vezes mais lento, originando uma *bit rate* de 250 kb/s. O primeiro símbolo a ser transmitido é sempre o menos significativo.

A frequência central dos canais para a banda 2.4 GHz são calculados através da Expressão 3.1, em MHz, e com $k = 11, 12, \ldots, 26$.

$$F = 2405 + 5(k - 11) \tag{3.1}$$

A técnica de modulação utilizada é o DSSS, com espaçamentos de 5 MHz entre bandas de operação que, por sua vez, ocupam 2 MHz. A potência de emissão está normalmente entre 0 e 3 dBm. Alguns rádios emissores/receptores conseguem no entanto atingir mínimos de -17 dBm. Devido a factores energéticos, quanto menor for a potência de emissão, menores serão os consumos de energia. Dependendo da aplicação, da densidade de nós e distâncias entre estes, a potência deve ser ajustada durante o desenvolvimento ou durante a auto-conFiguração da rede.

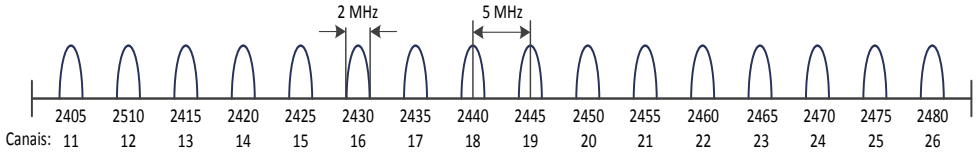

Figura 3.9 - Canais da banda de frequência 2.4 GHz.

No máximo e teoricamente, uma única rede pode ter até 65000 nós, dependendo no entanto, da plataforma *hardware* utilizada.

Tabela 3.3 – Canais 2.4 GHz e DSSS para IEEE 802.15.4.

Número do Canal	Frequência (MHz)
11	2405
12	2410
13	2415
14	2420
15	2425
16	2430
17	2435
18	2440
19	2445
20	2450
21	2455
22	2460
23	2465
24	2470
25	2475
26	2480

Originalmente, a norma suporta três tipos de dispositivos:

- ***Gateway* (*Ponto de Acesso*)** – é o coordenador da rede que regula toda a rede tais como canais a utilizar e a atribuição dos endereços. Teoricamente é um dispositivo com uma alimentação permanente e ilimitada;
- ***Full Function Devices* (FFD)** - que têm capacidade para fazer *relaying* de dados de todos e para todos os dispositivos além das capacidades habituais como processamento e medição;
- ***Reduced Function Devices* (RFD)** - que apenas medem, processam e emitem. Todos possuem endereços físicos e únicos atribuídos pelo coordenador.

A camada física contém vários atributos e procedimentos com objectivos específicos no âmbito da comunicação do próprio nó e da rede: CSMA/CA, CCA, ED, CS, RSSI e LQI.

- ***Carrier Sense Multiple Access Collision Avoidance* (CSMA/CA) -** é o método de acesso múltiplo utilizado em redes sem fios para aceder ao meio. O CSMA/CA previne a ocorrência de colisões no meio de transmissão através da competição pelo canal. O método verifica se o meio está ocupado com comunicações de outros nós. Se estiver desocupado a trama é emitida. Caso contrário, é adicionando um tempo de espera aleatório até realizar nova verificação do canal. Este tempo de espera, organizado por *slots* temporais (conhecidos como períodos de *back-off*), é conhecido como Janela

de Contenção – *Contention Window* (CW). Um período de *back-off* é igual a $aUnitBackoffPeriod = T_{symbol} * 20\ symbols$.

Como $T_{symbol} = \frac{1}{62.5\ ksymbols/s} = 16\ \mu s$, o período é igual $aUnitBackoffPeriod = 320\ \mu s$.

A duração da CW é determinada por um número aleatório de *back-offs* determinado pelo Expoente de *Back-off* – *Backoff Exponent* (BE). Esta variável BE é determinada por meios de uma função de distribuição uniforme, cujo valor mínimo é $macMinBE = 3$ e o valor máximo é $macMaxBE = 5$. A CW é calculada segundo a Expressão 3.2.

$$CW = (2^{BE} - 1) * aUnitBackoffPeriod \tag{3.2}$$

Um nó pode tentar enviar uma trama até 4 vezes. Se ao fim de 4 tentativas o meio permanecer ocupado, é enviada uma notificação da falha para as camadas superiores.

- ***Clear Channel Assessment*** **(CCA)** - é o procedimento que verifica se o canal de transmissão está ocupado ou não. O tempo de verificação é igual a $\mathrm{T_{CCA}} = 128\ \mu\mathrm{s}$ pois a verificação do canal é realizada dentro de 8 *symbols*. O tempo total necessário para realizar o procedimento CCA é dado pela Expressão 3.3.

$$ccaTime = rxSetupTime + T_{CCA} \tag{3.3}$$

O tempo *rxSetupTime* é o tempo que o rádio emissor/receptor demora a transitar entre estados cujo valor deve ser retirado do *datasheet* do módulo de comunicação [BGV13a]. A verificação é realizada através de duas componentes essenciais: *Energy Detection* (ED) e *Carrier Sense* (CS).

- ***Energy Detection*** **(ED)** - A medição ED estima o nível de potência do sinal recebido. Durante o procedimento CCA, se a potência medida for maior do que a variável *EDThreshold*, o canal é classificado como ocupado. O limite mínimo *EDThreshold* deve conseguir medir a potência de recepção para valores menores que 10 dB acima da sensibilidade do rádio emissor/receptor. A norma especifica que a sensibilidade de qualquer plataforma *hardware* tem de ser melhor ou igual que -85 dBm.

- ***Carrier Sense*** **(CS)** - Quando detectado um sinal no canal, o CS verifica se este possui as características e a modulação utilizada na norma IEEE 802.15.4. O CS é utilizado qualquer que seja a potência do sinal recebido, isto é, seja maior ou menor que o *EDThreshold*. O CS identifica portanto qual a natureza do sinal presente no meio.

- ***Received Signal Strength Indication*** **(RSSI)** - O RSSI é uma grandeza extremamente útil. Através da sua medição pode ser verificado qual o nível da potência do sinal

recebido. O RSSI mede os valores instantâneos desta potência e, portanto, o próprio ED é o resultado da média dos valores RSSI ao longo de $128\ \mu s$. O alcance e precisão do RSSI são valores dependentes de cada plataforma *hardware*.

- ***Link Quality Information* (LQI)** - O LQI representa a qualidade do sinal recebido e, consequentemente, a qualidade da ligação entre dois ou vários dispositivos. O LQI é desencadeado para todos os pacotes recebidos. O seu valor mínimo e máximo deve estar directamente associado aos valores mínimos e máximos da qualidade dos sinais detectáveis pelo rádio emissor/receptor, respectivamente.
  Os valores do LQI podem ser correlacionados com o *Packet Error Rate* (PER). O PER é a taxa de pacotes recebidos com erros, em função do número total de pacotes recebidos. Uma taxa igual a zero significa que nenhum pacote foi recebido com erros. Esta medição só é realizada para pacotes com tamanhos maiores do que 20 bytes.

- ***Physical Packet Data Unit* (PPDU)** - O PDU da camada física, denominado *Physical PDU* (PPDU) tem um tamanho máximo de 133 bytes (1064 bits) especificado pela norma e está representado na Figura 3.10. Teoricamente, a norma restringe o comprimento do PDU para o valor máximo de 127 bytes de dados e informação em qualquer trama. O encapsulamento final da camada física adiciona 48 bits (6 bytes) com propósitos de sincronização entre o nó emissor e o nó receptor.
  O *Synchronization Header* (SHR) e o *PHY Header* (PHR) compõe o cabeçalho da camada física que, por sua vez, faz o encapsulamento do *payload* (dados e informação) das camadas mais elevadas. O SHR é composto por 5 bytes e tem o objectivo de sincronizar a entrega dos dados do lado do receptor com os 4 primeiros bytes, terminando com o *Start-of-Frame* (SFD) para indicar o fim do SHR e o começo do PHR. O PHR de 1 byte tem como objectivo informar o tamanho da trama.
  A trama encapsulada proveniente da camada MAC é denominada neste estágio por *PHY Service Data Unit* (PSDU). Esta camada possui duas constantes: A *aMaxPHYPacketSide*, que indica o comprimento máximo duma trama, e a *aTurnaroundTime* (TA) que indica o tempo que o rádio emissor/receptor demora a transitar do estado de recepção (RX) para o estado de emissão (TX), ou vice-versa. Esta constante deve ser menor do que 12 *symbols*, isto é, 192 μs.

Figura 3.10 – Trama PPDU.

Figura 3.11 – Modo de Acesso Básico com resposta ACK.

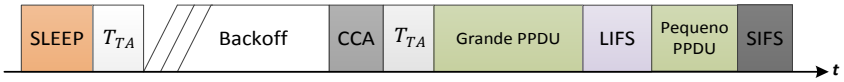

Figura 3.12 – Modo de Acesso Básico sem resposta ACK.

Se após a emissão de uma trama o nó precisar de emitir outra trama, este deve reservar um tempo de espera, antes de emitir a nova trama, denominado *InterFrame Space* (IFS). Na norma IEEE 802.15.4 existem dois tipos de IFS: *Small* IFS (SIFS) e *Long* IFS (LIFS). A primeira é igual a 192 µs (para PPDUs menores que 24 bytes) e a segunda igual a 640 µs (PPDUs acima de 24 bytes).

A norma IEEE 802.15.4 também inclui a emissão de uma trama *Acknoledgment* (ACK) como resposta à recepção com sucesso de uma trama de dados, terminando a comunicação sem erros ou falhas entre emissor e receptor. No entanto, por vários motivos, este mecanismo de resposta pode ser desactivado. As Figuras 3.11 e 3.12 apresentam o modo de acesso básico para transmissões de dados, com tramas de resposta ACK e sem ACK, respectivamente.

O tempo mínimo para transmitir um pacote, receber com sucesso um ACK e transitar de estado, é dado pela Expressão 3.4.

$$D_{min} = \overline{CW} + ccaTime + T_{TA} + T_{DATA} + T_{TA} + T_{ACK} + T_{IFS} \tag{3.4}$$

3.3.2 Subcamada MAC

A camada de ligação de dados é responsável por partilhar tramas de dados ou de sinalização entre dispositivos num meio comum a estes. Essas técnicas especificam como são colocados e recebidos os dados do meio físico, utilizando métodos de controlo de acesso ao meio e detecção de erros.

As pilhas protocolares gozam da capacidade das várias camadas poderem abstrair-se mutuamente, isto é, os encargos e deveres de uma camada não são da responsabilidade da camada seguinte. A camada física e a camada de ligação de dados executam as suas tarefas, independentemente dos protocolos utilizados nas camadas superiores. Por exemplo, a camada de ligação de dados é responsável por permitir que os dados provenientes das camadas superiores acedam directamente ao meio com técnicas do tipo *framing*. Isto possibilita que o mesmo pacote de dados seja transmitido através de diferentes meios físicos e diferentes

interfaces, cuja transferência é da exclusiva responsabilidade das interfaces físicas dos dispositivos.

A camada de ligação de dados é dividida em duas subcamadas:

- ***Logical Link Control*** – A subcamada LLC realiza o encapsulamento de um pacote independentemente dos protocolos utilizados nas camadas superiores;
- ***Medium Access Control*** – A subcamada MAC, após o encapsulamento do pacote, controla e coordena o controlo de acesso ao meio para que sejam partilhadas tramas entre nós vizinhos. O controlo de acesso ao meio irá depender de como esse meio pode ser acedido e como se realizam as ligações entre os nós.

- **Estrutura das Tramas em IEEE 802.15.4**

A Figura 3.13 apresenta a estrutura genérica do encapsulamento da camada MAC e respectivos cabeçalhos. O PSDU é composto pelo encapsulamento da subcamada MAC. O *MAC Header* (MHR) possui informação sobre endereçamento, sequências e segurança. No *MAC Footer* (MFR) é realizada a verificação de erros *Frame Check Sequence* (FCS). O *MAC payload*, proveniente das camadas superiores, é denominado neste estágio como *MAC Service Data Unit* (MSDU). A norma IEEE 802.15.4 prevê a criação de quatro tramas distintas na sub-camada MAC: A trama de dados, a trama *beacon*, a trama de comandos e a trama ACK. A Figura 3.14 mostra a estrutura das três primeiras tramas com o encapsulamento MAC e com diferentes tamanhos. A trama ACK é apresentada na Figura 3.15 separada das outras, pois é apenas formada pelos campos MHR e MFR, sem *MAC payload*. A emissão de tramas ACK, tramas de comando e tramas *beacon* não requerem a utilização prévia do CSMA.

A subcamada MAC cria as tramas com funções bastante específicas. O campo "Controlo da Trama" contém informação sobre as técnicas utilizadas em *framing* e os campos de endereçamento. O "Número da Sequência" evita o processamento de tramas repetidas ao verificar-se o mesmo número de sequência. O *Frame Check Sequence* (FCS) verifica se existem erros na trama após esta ser recebida.

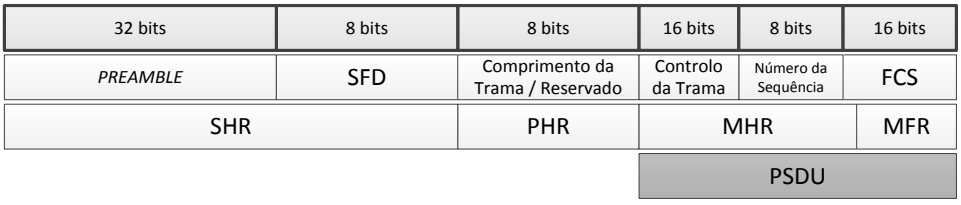

Figura 3.13 - Encapsulamento genérico da camada MAC.

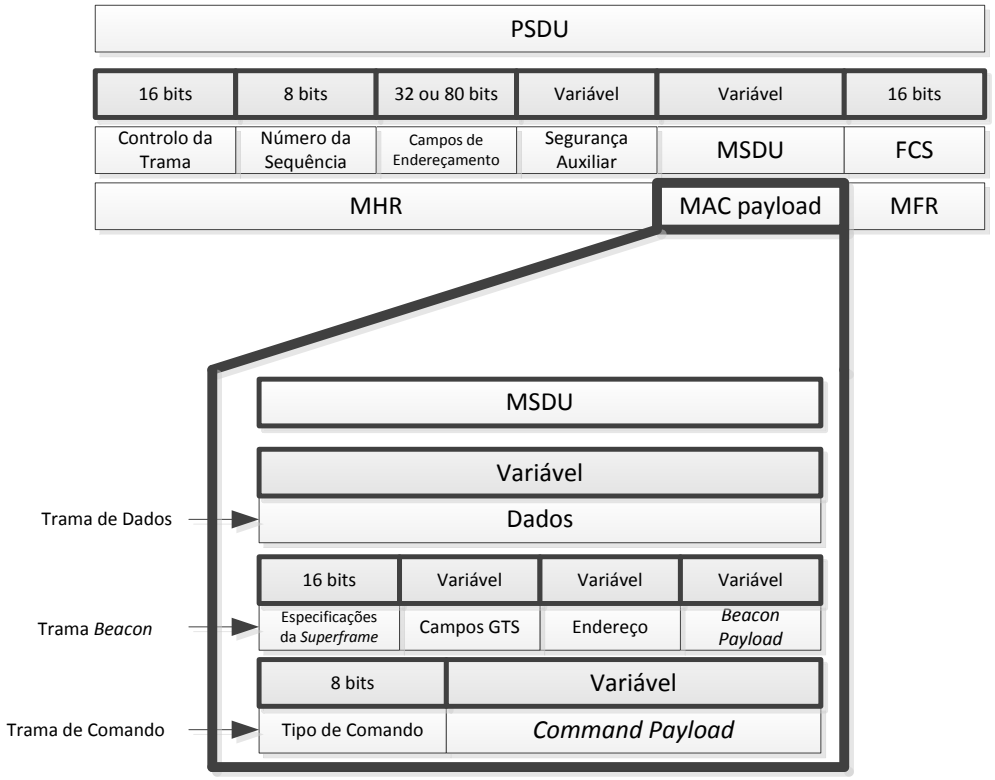

Figura 3.14 - Estrutura das tramas de dados, *beacons* e comandos.

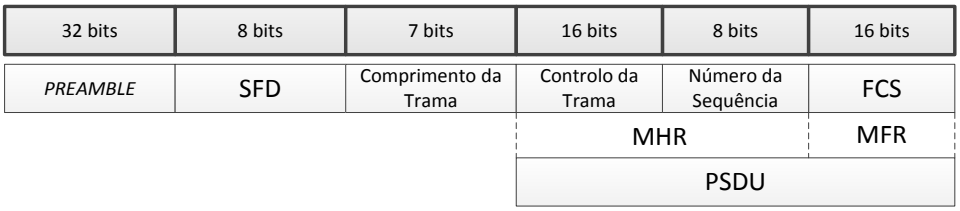

Figura 3.15 - Estrutura da trama ACK.

O endereçamento é uma das tarefas da subcamada MAC. Define um endereço, denominado endereço físico ou MAC, que representa e identifica localmente numa rede, os dispositivos finais no meio físico. O endereço MAC é um identificador global e único atribuído à interface física de cada dispositivo. A grande maioria dos dispositivos utiliza endereços MAC de 48 bits (EUI-48). Contudo, os dispositivos que operam segundo a norma IEEE 802.15.4 utilizam endereços de 64 bits (EUI-64). A Figura 3.16 apresenta a estrutura do endereço MAC EUI-64. A norma define igualmente um endereço substituto do EUI-64, o 16-bit *Short Address*, com 16 bits de comprimento, apenas para comunicações locais.

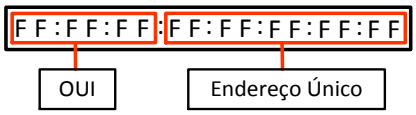

Figura 3.16 - Endereço MAC EUI-64.

O 16-bit *Short Address* é atribuído pelo coordenador da rede. A utilização do endereço de 16 bits nos cabeçalhos das tramas tem vantagens por ter menos 6 bytes que o endereço de 64 bits. Esta redução é importante em redes IEEE 802.15.4 pois reduz o desperdício de energia associado ao *overhead* dos cabeçalhos, além de reduzir o espaço ocupado em memória quando os nós armazenam os endereços dos nós vizinhos para serem utilizados, por exemplo, em protocolos de encaminhamento [IPSO09]. No entanto, a atribuição e utilização deste endereço de 16 bits é limitada. Primeiro, sendo o endereço atribuído pelo coordenador da rede através de uma associação estabelecida localmente, a sua validade e unicidade é limitada pelo tempo de vida dessa mesma associação. A expiração e não renovação da associação pode causar problemas de consistência dos endereçamentos atribuídos, e pode ser pouco eficiente em termos de escalabilidade da rede [RFC4944]. Segundo, estes endereços de 16 bits só podem ser utilizados localmente.

A norma IEEE 802.15.4 especifica um *MAC payload*, com 118 bytes de comprimento máximo. São 118 bytes de dados se nenhum outro protocolo ou informação for encapsulada nas camadas superiores. Se for utilizado o endereço de 64 bits, esse número é reduzido para 106 bytes de dados, resultando em menos 12 bytes de dados (menos 6 bytes no endereço de origem e menos 6 bytes no endereço de destino). Em RSSF, esta pode ser uma diferença determinante. As aplicações em RSSF caracterizam-se por possuírem dados que ocupam muito pouco espaço em memória. No entanto, muitas aplicações geram grandes quantidades destes dados.

Por outro lado, quanto menor for a trama, menor será o tempo necessário para ser emitida e recebida. Consequentemente, menos energia será consumida neste processo e mais cedo estará o meio disponível para outras comunicações.

- **Topologias de Rede**

As RSSF têm diferentes topologias. A topologia de uma rede é a forma como um dispositivo, os seus vizinhos e as ligações com os mesmos são vistas ao nível da segunda camada. Portanto, não é necessariamente a maneira como fisicamente os dispositivos estão distribuídos no espaço, mas sim, como a um determinado nível lógico, as ligações são realizadas. Assim, a maneira como é realizado o controlo de acesso ao meio é determinada pela topologia vista ao nível MAC. A norma IEEE 802.15.4 define dois tipos de topologia: em Estrela e Ponto-a-Ponto.

1. ***Topologia em Estrela***

Um nó coordenador encontra-se no centro da rede, ou sub-rede, rodeado por nós vizinhos. Este nó coordenador (ou PAN), após ser activado, constrói a rede atribuindo inicialmente uma identificação única para si e para a rede, diferente de outras redes próximas. Todos os

dispositivos da rede apenas podem comunicar com o coordenador. Na presença de dispositivos FFDs, alguns protocolos de encaminhamento possibilitam a selecção de outro coordenador diferente do primeiro, após terem decorrido determinados períodos de tempo e/ou após terem ocorrido elevados consumos de energia da parte do coordenador inicial.

2. ***Topologia Ponto-a-Ponto***

Todos os nós sensores podem comunicar uns com os outros se estiverem no seu raio de alcance e fazer *relaying* de dados caso pertençam à classe FFD, em comunicações multi-salto (*multihop*). Existe um nó coordenador FFD cuja escolha pode ser feita de várias maneiras dependendo do protocolo utilizado. A topologia ponto-a-ponto agrupa duas categorias diferentes: *flat* e hierárquica. A categoria *flat*, mais conhecida por topologia *mesh*, caracteriza-se pela capacidade dos FFDs criarem caminhos entre os nós finais e o nó coordenador. Este tipo de topologias requerem níveis elevados de auto-organização e auto-reparação quando esses caminhos sofrem falhas ou se tornam obsoletos. É um tipo de topologia bastante utilizado nas redes *Ad Hoc*.

A topologia mais conhecida da categoria hierárquica é a topologia em árvore. Nesta topologia, os encaminhamentos são inicialmente definidos pelo nó coordenador. Tem algumas semelhanças com a topologia em estrela mas com a profunda diferença dos dispositivos da rede poderem comunicar uns com os outros dentro da mesma árvore. Permite alcançar extensões elevadas sem que o nó coordenador necessite de abranger todos os nós membros da sua árvore. Estas topologias necessitam que seja garantida a sincronização por parte do coordenador, principalmente se existirem técnicas de transição de estado activo para estado SLEEP, de maneira a manter a consistência das comunicações e dos dados. Para tal, são indicados não só vários protocolos MAC como protocolos de rede.

Uma outra designação para esta categoria hierárquica é a topologia *cluster*. Um *cluster*, ou aglomerado, é um grupo de dispositivos hierarquicamente distribuídos por diversos saltos, que encaminham dados do e para um coordenador do grupo (ou vários coordenadores). Este coordenador é denominado por *Cluster Head* (CH) e é escolhido entre os possíveis FFDs presentes na rede. Esta escolha é realizada através de diferentes protocolos de rede bastante revistos na literatura como o LEACH [HCB00], o HEED [YF04] e o EEHRP [LSYZ10], protocolos de rede que organizam topologias de rede para resolução de problemas e/ou melhoria da eficiência da rede e dos recursos dos nós.

- **Protocolos MAC**

A subcamada MAC é de máxima importância para as redes de capacidade reduzida LR-WPAN como as RSSF. Tendo em mente que uma das principais características, se não a mais importante neste tipo de redes, é a reduzida disponibilidade de energia nos nós, o controlo de acesso ao meio e a transferência dos dados tem de ser o mais eficiente possível. No

contexto das redes sem fios em geral, o controlo e coordenação dos nós no acesso ao mesmo meio é um objectivo bem mais difícil e complexo de atingir, com eficiência e fiabilidade, do que nas redes cabladas. O comportamento da comunicação é, já da sua própria natureza, de difusão (*broadcast*). Essa emissão em difusão de dados levanta problemas, como a ocorrência de interferência com outras transmissões no mesmo raio de alcance, e o nível reduzido de segurança quando dispositivos indesejáveis se encontram ao alcance dos nós que geram e partilham dados.

A interferência por si mesma leva a colisões entre tramas ou à distorção dos dados, corrompendo-os. A perda dos dados numa transmissão obriga à retransmissão dos mesmos, o que significa que tem de ser consumida mais energia para que a mesma transmissão seja realizada com sucesso. Se as colisões e os erros não forem tratados com eficácia, os mesmos dados terão de ser retransmitidos vezes sem conta, até que se tenha sucesso, desperdiçando uma quantidade imensa de energia.

Mesmo com o funcionamento eficiente do CSMA, a colisão entre tramas é uma ocorrência bastante comum por vários motivos. O primeiro são os falsos positivos, indicadores de que o meio está livre quando na realidade se encontra ocupado. O segundo e terceiro são os problemas "terminal escondido" e "terminal exposto" [KW05]. A Figura 3.17 ilustra ambos os problemas.

O problema do terminal escondido ocorre quando um nó está fora do alcance de outro nó e ambos comunicam no mesmo instante para um nó dentro do alcance de ambos. Na Figura 3.17, o nó B realiza o procedimento CCA descobrindo que o meio está livre para enviar dados para o nó C. Imagine-se que antes de completar a sua transmissão, o nó A também pretende trocar dados com C. Como A está fora do alcance de B, ao realizar o CCA, não conseguirá ouvir a sinalização de B, classificando erroneamente o canal como estando livre para realizar a sua própria emissão de dados. Se essa transmissão de A para C acontecer quando a transmissão de B para C ainda estiver a ser realizada, ocorrerá uma colisão em C, sem que B e A se apercebam.

O problema do terminal exposto ocorre quando um nó, ao realizar o CCA, é impedido de enviar dados ao detectar o canal ocupado, mas cuja emissão não causaria interferência no nó receptor. Na Figura 3.17, o nó B realiza o CCA e a emissão de dados para o nó A. Como o nó C está na sua área de cobertura, ao realizar o CCA com intensão de enviar dados para o nó D, verifica que o canal se encontra ocupado, sujeitando-se a mais tempo de *back-off*. No entanto, como D está fora do alcance de B, o atraso na comunicação de C é desnecessário e ineficiente.

Foram propostas algumas soluções para a resolução dos dois problemas. Um é o *busy-tone* que utiliza um segundo canal de controlo, no qual o nó receptor emite sinalização aos seus

vizinhos indicando que está a receber dados. Esta sinalização não deverá ser nem muito fraca (para que seja recebida por todos), nem muito forte (de maneira a não forçar desnecessariamente nós longínquos a cessarem as suas emissões), mas sofre da grande desvantagem dos nós terem de funcionar em modo *full-duplex* para conseguirem emitir e receber em simultâneo.

O mecanismo *Request/Clear-to-Send* (RTS/CTS) dá resposta a ambos os problemas. Definido inicialmente para a norma IEEE 802.11, tem como objectivo reservar o canal para ser realizada uma comunicação [KW05], [BDWL10], [RCS05].

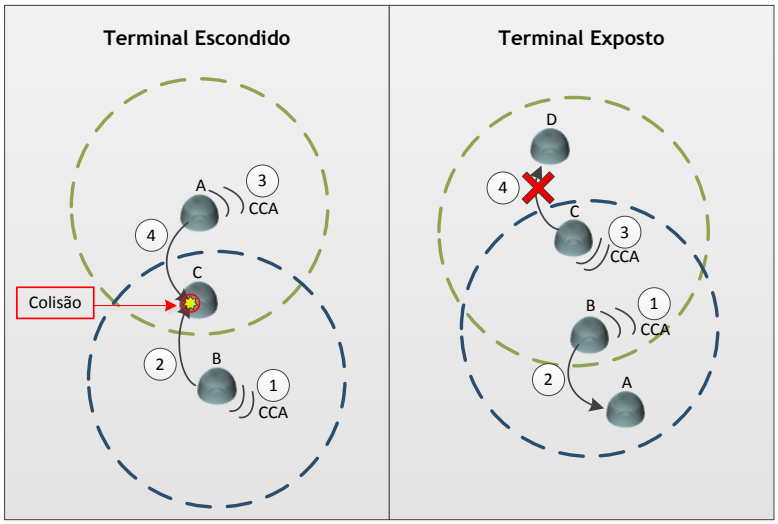

Figura 3.17 - Problema do Terminal Escondido e Terminal Exposto.

Apesar do mecanismo RTS/CTS não estar explicitamente descrito na norma IEEE 802.15.4, a sua adopção é largamente aceite nas RSSF. Após realizar o procedimento CCA e descobrir o canal desocupado, um nó emissor emite primeiro uma trama denominada RTS com um PSDU de apenas 20 bytes (como definido pela norma IEEE 802.11), para o receptor desejado.

Do lado do nó receptor, após receber o RTS sem colisão, realiza-se o CCA. Se se verificar que o canal está de facto desocupado, o receptor responde com um pacote CTS com PSDU com um comprimento típico de 14 bytes. O nó emissor original, após aguardar um breve período de tempo (*aTurnAroundTime*), receberá o CTS que assegurará que o canal está livre e podendo finalmente emitir os seus dados, livre de colisões.

O mecanismo RTS/CTS adiciona um tempo, denominado NAV (*Network Allocation Vector*), a todos os nós vizinhos que possam ser possíveis interferentes. Após receberem RTS, CTS, dados ou ACK, os nós vizinhos estabelecem o temporizador NAV durante o tempo indicado na respectiva trama recebida, onde transitam e permanecem no estado SLEEP até esse temporizador terminar. A Figura 3.18 apresenta o mecanismo RTS/CTS com NAV e ACK para uma comunicação entre A e B.

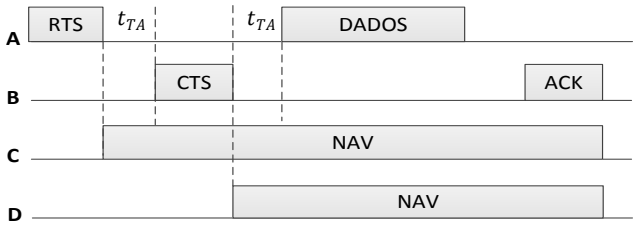

Figura 3.18 - Mecanismo RTS/CTS com NAV e ACK.

Apesar de prevenir colisões, o mecanismo RTS/CTS em RSSF pode introduzir elevado *overhead*, entre 40 % a 75 % da capacidade do canal, pois as tramas de dados têm um comprimento bem mais reduzido quando comparadas com as tramas em IEEE 802.11 [WC01]. Especificamente, mesmo com o comprimento máximo de 127 bytes, a eficiência é bem mais reduzida quando comparada com o IEEE 802.11 cujo comprimento máximo das tramas de dados é 2346 bytes. Pelo mesmo motivo, os autores de [BDWL10] afirmam que além de deteriorar o desempenho da rede, sendo a probabilidade de colisão do RTS igual à probabilidade de colisão de dados (assumindo tramas de dados com comprimentos reduzidos), aumenta o consumo de energia.

Existem esquemas de modulação e codificação que aumentam a fiabilidade da recepção das tramas sem a ocorrência exagerada de erros e oferecem algum dinamismo na potência de transmissão. Estes esquemas mais complexos são utilizados nas redes celulares, tal como o OFDMA para acesso múltiplo por frequência ou o TDMA que divide temporalmente o acesso dos dispositivos atribuindo-lhes *slots* reservados onde podem comunicar. A maioria destas técnicas não são no entanto as mais indicadas para as LR-WPAN por causa das suas características tão peculiares e únicas.

Os protocolos da subcamada MAC nestas redes foram criados para obedecer, não só às suas exigentes restrições, como às restrições das redes de comunicação em geral. Existe um vasto leque de trabalhos na literatura que redesenham esta subcamada com objectivos diferentes mas correlacionais. Dependendo normalmente da aplicação, diferentes equilíbrios (*tradeoffs*) podem ser aplicados referentes principalmente à poupança de energia e ao tráfego reduzido.

Os protocolos MAC podem ser assíncronos ou síncronos. Os protocolos síncronos podem realizar sincronização local para um conjunto de nós vizinhos transitar do estado SLEEP para o estado activo no mesmo intervalo de tempo, ou podem realizar uma sincronização global ao atribuírem-se *slots* temporais aos nós.

De uma forma geral, a sincronização da rede pode ser realizada de dois modos: sincronização das ligações e sincronização dos estados entre nós. A primeira é baseada em reservas – *Time Division Multiple Access* (TDMA) - e a segunda é baseada em competição (CSMA). Os protocolos MAC de sincronização são extremamente diversificados.

O TDMA é um método utilizado por protocolos síncronos globais. Trata cada nó de forma igual, logo há equidade de acesso entre eles, funcionando bem para redes de médias dimensões. Resolve problemas relacionados com interferência e baseia-se na agregação de dados. Contudo, sem a presença de um nó coordenador fixo que actue como ponto de acesso, isto é, em redes descentralizadas, pode haver dificuldade em manter a sincronização. Igualmente, em redes com um elevadíssimo número de nós e onde há necessidade de manter múltiplos calendários, esta pode ser uma tarefa com resultados energeticamente insustentáveis. Caso o tráfego e a distribuição dos nós seja dinâmica, as consequentes e sucessivas alterações dos calendários temporais podem deteriorar toda a rede, assim como podem levar à existência de ciclos activos (*duty cycles*) demasiados longos no estado activo dos nós.

1. *Estrutura Assíncrona*

O ponto forte dos protocolos assíncronos é a não existência de calendarizações dos estados ou das ligações entre nós, e, portanto, o *overhearing* de tramas de controlo é basicamente eliminado. A maior resultante disto é que de que os nós passam a ter períodos de inactivação bastante mais longos do que com os protocolos síncronos, logo menos energia é consumida e desperdiçada no estado *idle*. Para redes pequenas com quantidades de tráfego reduzidas e que sejam tolerantes a atrasos, a estrutura dos protocolos assíncronos é a mais indicada.

O *preamble sampling* é a técnica assíncrona mais conhecida. Sempre que um nó deseja comunicar, emitirá no canal um *preamble* detectável pelos seus vizinhos. Um nó, sempre que acorde e detecte o *preamble*, aguardará até este finalizar e ter sido recebido o pacote de dados do emissor, com o consequente ACK se for requerido. Se por outro lado, o nó acordar e não detectar o canal ocupado, voltará para o estado SLEEP. A duração do *preamble* tem de ser no mínimo igual ao período de tempo entre duas consecutivas transições de dois nós.

Esta técnica foi aperfeiçoada por outros protocolos assíncronos na medida em que podem dinamizar o *duty cycle* dos períodos activos consoante o tráfego, melhorar o procedimento CCA, utilizar diferentes frequências para efectuar o *preamble*, dividir o *preamble* em pacotes com intervalos de tempos reduzidos entre os mesmos, calendarizar as transições de estado dos vizinhos (oferecendo um certo grau de sincronismo), ou o emissor adicionar o endereço do receptor nos pacotes *preamble* de maneira a evitar que outros recebam os dados desnecessariamente.

2. *Sincronização do Estado dos Nós*

A sincronização dos estados dos nós é extremamente utilizada em redes divididas em *clusters*. O objectivo principal é sincronizar as transições de estado entre os nós. Estudos provam que esta é provavelmente a melhor estratégia para aplicações com um grau elevado de

periodicidade, onde o tráfego mantem uma determinada ordem e onde é necessário assegurar a total fiabilidade da rede através de sinalização de controlo. No entanto, não é a melhor solução para aplicações com tráfego irregular.

Entre os principais e mais conhecidos protocolos desta estratégia encontra-se o S-MAC que introduz o pacote de controlo SYNC (que sincroniza os relógios dos nós na mesma *cluster*), RTS, CTS e NAV. Os dados são fragmentados e enviados em pequenos pacotes para evitar retransmissões demasiado longas em situações onde ocorrem colisões. As desvantagens desta estratégia são os *duty cycles* fixos e a necessidade aderente de optimizar esses *duty cycles*. Por um lado, pequenos períodos activos levam a menores *idle listenings* mas, por outro lado, leva à existência de a uma taxa elevada de competição e colisões.

O protocolo T-MAC utiliza a mesma ideia do S-MAC tornando-se mais flexível ao diminuir dinamicamente o tempo activo dos nós que sintam o meio ocupado com dados não direccionados para eles. Igualmente, os nós podem voltar ao estado SLEEP se num determinado *time-out* nenhum evento ocorrer. Além disso, introduz a trama de controlo FRTS para resolver problemas de quebras de sincronização no *cluster* quando os nós adormecem antes do tempo pré-definido. Um nó que deseje ser o próximo emissor emitirá o FRTS indicando que acordará após a comunicação actual terminar, mantendo os possíveis receptores acordados. Outros protocolos MAC foram desenhados para aumentar a eficiência do RTS/CTS nas RSSF, nomeadamente o MACA e o PAMAS

O objectivo dos protocolos baseados nesta estratégia e encontrados na literatura é reduzir tempos de latência no estado SLEEP dos nós, proporcionar capacidade de adaptação a diferentes volumes de tráfego, capacidade de adaptação a diferentes topologias de rede, e capacidade de adaptação a aplicações com mobilidade, sendo este último alvo de estudos que melhorem a fiabilidade das comunicações quando os nós têm possibilidade de se moverem livremente numa determinada área. Outros protocolos atingem desempenhos satisfatórios ao adicionarem modelos estatísticos.

3. *Sincronização das Ligações entre Nós*

Estes protocolos encaixam-se na família de protocolos baseados em reservas. Como utilizam a técnica TDMA, os protocolos podem ser denominados *Frame-Slotted*, pois há atribuição de *slots* temporais nas ligações entre nós. Em redes com tráfegos muito "pesados", onde há grande probabilidade de ocorrerem colisões e a competição pelo meio é bastante elevada, esta é a melhor estratégia. A grande vantagem é que após a calendarização ser concluída sem erros, a probabilidade de ocorrerem colisões é basicamente nula, assim como é diminuído o *overhearing*, e além de se poder minimizar o *idle listening*, optimizando ao máximo o consumo de energia dos nós.

Existem três ramos deste tipo de sincronização: Calendarização da Ligação entre Emissor e Receptor, Calendarização do Emissor e Calendarização do Receptor. Os três tipos têm vantagens e desvantagens uns sobre os outros. O primeiro classifica-se como o que elimina totalmente as colisões e o *overhearing* mas que pode gerar problemas se o tráfego e a estrutura da rede for dinâmica, sobressaindo a necessidade de preparar novos calendários ao longo do tempo. O segundo pode aumentar o *overhearing* dos nós que recebam dados que não são direccionados para eles. O terceiro apenas aumenta a probabilidade de ocorrerem colisões quando vários emissores tentam comunicar com o mesmo receptor.

Os três encaixam-se perfeitamente em aplicações com condições e requisitos distintos. Enquanto que os dois primeiros tipos têm melhores resultados em redes com elevado tráfego e com o mínimo de latência, o terceiro tipo oferece melhores resultados em tráfegos menos "pesados".

Apesar da eficiência elevada, os protocolos construídos para este tipo de sincronização tentam dar a resposta a um número vasto de problemas. Manter uma boa sincronização entre os nós, evitar atribuições de *slots* sobrepostos, arranjar uma maneira de possibilitar comunicação *broadcast* quando requerida, aumentar a flexibilidade mantendo os calendários actualizados consoante as alterações sofridas na rede e optimizar a utilização da memória dos nós necessária para manter as calendarizações.

No geral, o TDMA não é dinâmico mas segundo alguns protocolos, se o tráfego puder ser medido, podem-se alterar os períodos activos dos nós, adaptando-se ao tráfego. Isto pode incluir algum *overhead* e complexidade no processamento por causa da inserção da capacidade RSSI que mede o nível dos sinais recebidos, mas também diminui os *idle listenings* e os atrasos em aplicações que necessitem de trocar dados urgentes o mais rápidamente possível.

Para aumentar a eficiência das comunicações, alguns protocolos adicionam a técnica FDMA ao TDMA. Entre as muitas razões desta adição, está a falta de fiabilidade da rede sincronizada TDMA quando o número de nós é muito elevado. A fiabilidade reduzida resulta de vários factores, sendo um deles o preenchimento total dos *slots* temporais disponíveis e/ou definidos numa *Superframe*, sem que o nó coordenador consiga atribuir *slots* a todas as ligações entre todos os nós. Como tal, atribuir diferentes frequências a ligações que estejam no mesmo *slot* temporal pode resolver o problema, apesar de ser um esquema mais complexo. Os protocolos que usam esta estratégia são conhecidos como protocolos MAC Multicanal (*multi-channel*).

O *Time Synchronized Mesh Protocol* (TSMP) é um protocolo MAC Multicanal. Numa rede com uma topologia em árvore, o nó coordenador agrega informação sobre os nós da rede e cria uma tabela de *slots* de tempo e frequência, atribuindo esses *slots* às ligações entre nós

vizinhos. Como a norma IEEE 802.15.4 implementada em 2.4 GHz pode utilizar 15 canais diferentes, 15 níveis diferentes podem ser construídos na tabela, com um número de colunas temporais que varia de aplicação para aplicação. A norma WirelessHART para RSSF especifica a utilização deste protocolo. As regras de ouro são:

- Duas ligações distintas nunca podem partilhar o mesmo *slot* de tempo/frequência;
- Num determinado período de tempo, um nó nunca pode receber ou emitir pacotes para mais do que um nó vizinho.

➢ **Método de Acesso *Beacon-Enable***

Este método, também designado como *slotted* CSMA/CA, utiliza uma *Superframe* definida pela norma IEEE 802.15.4. É extremamente utilizada em protocolos síncronos, onde os FFDs têm capacidade para emitir *beacons* e manter uma estrutura de *slots* temporais calendarizada. A estrutura da *Superframe* permite existir um período onde os nós estão activos e um período de inactivação dos mesmos para poupança de energia no estado SLEEP. O período activo tem disponíveis 16 *slots* temporais e é dividido em três sub-períodos dentro da *Superframe*: *Beacon Frame Period*, *Contention Access Period* (CAP), *Contention-Free Period* (CFP). A Figura 3.19 apresenta a estrutura da *Superframe*.

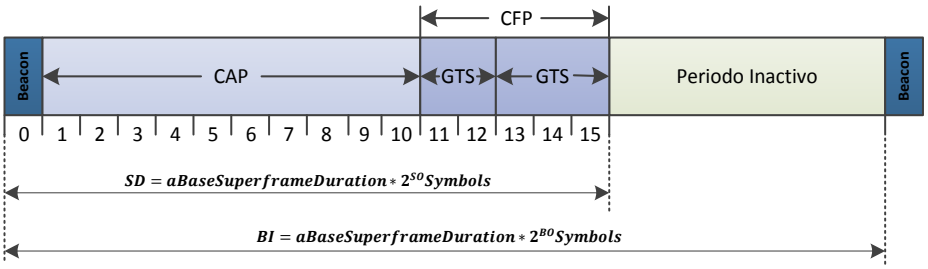

Figura 3.19 - Estrutura da *Superframe*.

Inicialmente, a *Superframe* inicia um novo cíclo com um *beacon* emitido pelo coordenador da rede com o objectivo dos nós sincronizarem-se com a *Superframe*. Após a recepção do *beacon* na *slot* 0, o mecanismo CSMA/CA é utilizado por todos os nós durante o CAP para emitirem os seus dados, competindo pelo acesso a um canal.

Após 10 *slots* de CAP, o período CFP organiza conjuntos de *slots* livres de competição (GTS), isto é, *slots* atribuídos especificamente a nós que necessitem de acesso determinístico ou largura de banda garantida. Estes dispositivos não necessitam, portanto, de utilizar o CSMA/CA enquanto as suas ligações estiverem atribuídas nas GTS. O pedido de atribuição ou eliminação de um nó numa GTS é realizado ao nível da camada de rede e decidido pelo coordenador. O coordenador da rede pode aceitar ou negar estes pedidos. Após terem passado no máximo 7 *slots* GTS, a *Superframe* finaliza com um período de inactivação

calendarizada (onde todos os nós podem transitar para o e estado SLEEP), até uma nova *Beacon Frame* ser emitida para dar inicio a uma nova *Superframe*.

- **Método de Acesso *NonBeacon-Enable***

Este método não utiliza a *Superframe* sendo, portanto, também designado como *unslotted* CSMA/CA. Este é um método competitivo, puramente assíncrono e não-calendarizado, onde sempre que um nó queira emitir os seus dados para os vizinhos e/ou coordenador da rede, tem de executar o CSMA/CA para ganhar acesso ao canal. Quando um nó quer comunicar, o coordenador da rede utilizada um *beacon* de maneira a sincronizar a emissão e recepção de dados. O *beacon* emitido pelo coordenador pode ser utilizado quando o coordenador deseja que um nó da rede saiba que existem dados à espera de serem transmitidos para esse mesmo nó. O campo *Pending Address Fields* contem os endereços destes nós com dados pendentes no coordenador. A utilização obrigatória do *beacon* nestes casos deve-se ao simples facto do coordenador não poder emitir tramas directamente para os nós se estes estiverem no estado SLEEP e, portanto, deve-se garantir que existe um determinado grau de sincronismo na rede.

- **Considerações sobre a versão IEEE 802.15.4-2012**

A última actualização aprovada da norma foi publicada em Fevereiro de 2012. São propostos alguns protocolos: O *Coordinated Sampled Listening* (CSL), o *Receiver-Initiated Transmission* (RIT), o *Deterministic and Synchronous Multi-Channel Externsion* (DSME) e o *Time-Synchronized Channel Hopping* (TSCH). Um dos objectivos da reformulação é incluir técnicas de multi-canal na norma, conceito de onde surgiu o TSCH. A actualização da norma incorporou novas tecnologias na camada MAC, bem como modificações ao nível da estrutura das tramas. Por exemplo, foram adicionadas às tramas ACK os campos de endereço, segurança e *payload*.

O CSL é um protocolo que utiliza o conceito *preamble sample* multi-canal, onde os nós acordam durante um breve período de tempo e voltam a transitar para o estado SLEEP caso não detectem nenhum *preamble*. A duração do *preamble* deve ser o suficiente para cobrir pelo menos duas transições para o estado activo dos nós receptores. O CSL pode ser utilizado tanto numa rede síncrona como assíncrona. No caso assíncrono, nó emissor utiliza um *preamble* longo o suficiente para que todos os potenciais receptores acordem e mantenham-se em modo de recepção até à transmissão da trama. No caso síncrono, o *preamble* é muito mais curto e tem o objectivo de compesar atrasos de relógio e perdas de sincronização entre os nós

O RIT utiliza o conceito de *polling* ao realizar uma solicitação de dados em difusão. O RIT introduz mais latência do que o CSL.

Baseado no TSMP, o TSCH é um protocolo de sincronização de ligações que visa construir uma tabela composta por várias *slotframes* em diferentes canais e sincronizada entre todos os nós da rede. Uma *slotframe* é basicamente uma *superframe* formada por um conjunto de *slots* temporais onde se realizam as comunicações num determinado canal. As *slotframes* são atribuídas a diferentes canais para que diferentes comunicações sejam realizadas em simultâneo. O protocolo utiliza técnicas de multi-canal e reutiliza o tempo e a frequência dum padrão cíclico, repetindo-se sempre o mesmo calendário caso não ocorram alterações na topologia de rede. Os nós sincronizam as suas ligações nesta tabela. Do ponto de vista do emissor, cada *slot* contém a ligação para um ou mais nós vizinhos caso o pacote de dados seja direcionado de um nó emissor para vários nós receptores. O comprimento da *slot* é suficiente para seja transmitida uma trama e recebido um ACK.

A Figura 3.20 apresenta a estrutura de uma *slotframe* de três *slots* temporais numerados. O nó A comunica com nó B, e nó B comunica com nó A. A terceira *slot* apesar de existir não é utilizada para nenhuma comunicação. O número total de *slots* temporais que já passaram desde o inicio da rede denomina-se *Absolute Slot Number* (ASN) e é incrementado globalmente em toda a rede.

Temporalmente sincronizados, o TSCH permite que os nós participem em simultâneo em diferentes *slotframes*. Diferentes *slotframes* podem ter diferentes comprimentos (diferentes ASN). Ao longo do funcionamento da rede, as *slotframes* podem ser adicionadas, removidas ou modificadas pelo coordenador da rede.

Podem ser atribuídas duas comunicações diferentes numa só *slot*. Este tipo de ligações partilhadas pode levar repetitivamente a colisões. Para reduzir as colisões sucessivas, quando a transmissão num *slot* partilhado falha, o nó emissor espera um tempo aleatório $(2^{BE} - 1)$ e retransmite a trama num novo *slot*. Se o novo *slot* for também partilhado, é adicionado um tempo de *back-off* especial para estas retransmissões. A CW resultante deste *back-off* de retransmissões aumenta a cada nova retransmissão até ter sucesso ou até atingir um número de tentativas máximo e notificar as camadas superiores da falha. Em *slots* não partilhados, o *back-off* pode ser desactivado e o CCA iniciado imediatamente.

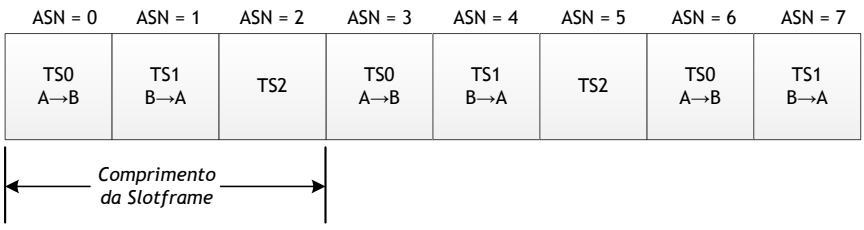

Figura 3.20 – Exemplo de uma *Slotframe* com três *slots*.

A construção do calendário pode ser realizada com uma estratégia *beaconing* ou *non-beaconing*. Em ambas as estratégias, os nós mais próximos do coordenador utilizam *beacons* (ou tramas de dados para efeitos de sincronização caso seja uma estratégia *non-beaconing*) para se sincronizarem com o coordenador. Todos os nós seguintes que sejam adicionados à rede utilizarão tramas de publicitação (*advertisement frames*) para informarem a sua presença e sincronizarem-se com os nós vizinhos mais próximos, previamente reconhecidos e adicionados pelo coordenador. Toda a sincronização dos *slots* temporais é mantida por um temporizador *keep-alive*. Após os *slots* serem atribuídos, os *beacons* deixam de ser necessários para dar início às comunicação.

Periodicamente, as ligações entre emissor e receptor mudam dinamicamente de canal consoante uma sequência conhecida por ambos os dispositivos. Esta técnica permite atribuir ligações em frequências diferentes, evitando a ocorrência de colisões. A sequência da escolha dos canais (*Channel-Hopping* - CHop) inclui um identificador, um tamanho pré-determinado e uma lista de canais disponíveis. A escolha do canal de comunicação, para um determinado período de tempo, numa ligação entre dois nós, é calculada através da Expressão 3.5.

$$CHop = macHoppingSequenceList \\ * [(macASN + channelOffset)\%macHoppingSequenceLength)] \tag{3.5}$$

$macHoppingSequenceList$ é o número do canal actual da banda de frequência em utilização, $macASN$ é o número de ASN, $macHoppingSequenceLength$ é o número de canais dentro da sequencia actual, $channelOffset$ permite que diferentes canais sejam utilizados para um dado $macASN$ e $macHoppingSequenceList$.

Por exemplo, para $macASN = 2$, $channelOffset = 16$ e $macHoppingSequenceLength = 16$ o resultado é $CHop = 2$, isto é, a ligação é transmitida na frequência 2410 MHz (canal 2).

O TSCH é um protocolo extremamente robusto, desenhado para aplicações com geração e partilha de dados periódica, e que necessitem de possuir fiabilidade elevada. Contudo, é um protocolo complexo e, na prática, existem várias implicações ao nível dos recursos da rede. Por exemplo, todos os nós armazenam calendarizações. Este aspecto, além de requerer alguma memória disponível, origina inicialmente bastante *overhead* para que sejam parcialmente conhecidas as ligações (por exemplo, para a atribuição ou negociação de um canal) e mantida a sincronização dos *slots* entre todos os nós. Na mesma medida, mesmo que um nó não tenha quaisquer dados para enviar, os nós receptores transitarão para o estado activo durante os *slots* correspondentes, enquanto todos os outros potenciais emissores permanecem à espera dos seus *slots*, causando atrasos e desperdício de energia.

O DSME utiliza o mesmo conceito Multicanal do TSCH mas com a estrutura da *Superframe*. O conceito da *Multi-Superframe* consiste em construir um ciclo de sucessivas e repetidas

Superframes em vários canais de comunicação, simultaneamente. Estas *Multi-Superframes* são tradicionalmente formadas pela *Beacon Frame*, CAP e atribuições GTS no CFP.

3.4 Sumário e Conclusões

Neste Capítulo propõe-se que a norma IEEE 802.15.4 seja utilizada nas redes capilares para a IoT e as comunicações M2M. A norma é estritamente direccionada para as redes sem fios de potência, alcance e ritmos de transmissão reduzidos (LR-WPAN), nomeadamente redes de sensores sem fios. As RSSF são compostas por dispositivos de tamanho e custo bastante reduzido, construídos com circuitos de instrumentação, sensores, actuadores e microprocessadores com desempenhos moderados de maneira a possibilitar consumos de energia reduzidos. Estes dispositivos, integrados em objectos e outros elementos físicos de um determinado ambiente, possibilitam a aquisição de dados e posterior partilha dos mesmos com outros dispositivos ou pontos de acesso.

A norma IEEE 802.15.4 está optimizada para aplicações de controlo e monitorização. É responsável por definir a camada física e a subcamada MAC. As suas especificações impõem limites ao nível dos recursos espectrais e energéticos para cumprir os requisitos deste tipo de redes. A optimização do tempo de vida das redes é um desses requisitos. Limitados fisicamente ao nível do consumo energético, os períodos activos de comunicação e processamento são extremamente reduzidos e adoptam estratégias de poupança de energia como, por exemplo, a transição de componentes *hardware* para estados "adormecidos" com consumos muito reduzidos. Mesmo em estados activos, o consumo de energia dos dispositivos continua a ser bastante reduzido para que os dispositivos permaneçam o máximo de tempo possível presentes e activos do ambiente e na rede, até ser reposto o sistema de alimentação (por exemplo, baterias).

Com potências de transmissão reduzidas, as comunicações têm raios de alcance não muito elevados. São redes cujos nós se encontram próximos, distanciados no máximo a algumas centenas de metros (~300 metros) e, portanto, a norma IEEE 802.15.4 define um conjunto de estratégias de forma a ultrapassar problemas de interferência.

Existe um número elevado de trabalhos relacionados com a norma IEEE 802.15.4 e as RSSF, cujos autores propõem algoritmos de optimização destas redes. Dependendo da aplicação em causa e das suas características, alguns protocolos são mais vantajosos do que outros, aumentando a eficiência de alguns requisitos mas diminuindo a qualidade noutros aspectos. Para cada aplicação, existem determinados equilíbrios que necessitam ser avaliados para o bom desempenho da aplicação e da rede consoante requisitos próprios.

Existem duas propostas muito semelhantes de pilhas protocolares para estas redes. Ambas as pilhas utilizam as duas primeiras camadas definidas pela norma IEEE 802.15.4. A Aliança

ZigBee foi a pioneira a definir a camada de rede e a camada de aplicação. Mas a ausência do endereço lógico IP motivou o IETF a definir protocolos de rede e aplicação que possibilitam a integração e comunicação directa com a Internet, e outras redes exteriores, por IPv6. No entanto, a ZigBee tem vindo a adoptar e a desenvolver protocolos que visam atribuir endereços lógicos IPv6.

Capítulo 4

Plataformas Hardware para RSSF

O mercado tem vindo a procurar maneiras de tornar os componentes *hardware* cada vez mais baratos e miniaturizados, nomeadamente para aplicações de baixo custo em redes de sensores sem fios. Devido às severas restrições na operação destas redes, os dispositivos devem ser concebidos de forma a respeitar estas limitações energéticas e computacionais. Actualmente, estas plataformas com capacidade de processamento e de comunicação, permitem rápidas implementações de baixo custo, altamente flexíveis, com moderados níveis de desempenho computacional, suporte robusto para uma vasta variedade de sensores e actuadores e, portanto, com capacidade de dar resposta às necessidades das aplicações inteligentes embutidas nos ambientes. Plataformas mais recentes permitem ainda consumos mais reduzidos em todos os estados possíveis, comunicações mais fiáveis, maiores níveis de processamento e compatibilidade com as mais recentes inovações tecnológicas ao nível de sensores.

4.1 Caracterização das Plataformas *Hardware*

Nesta Secção, são descritas as características mais importantes de algumas plataformas *hardware* presentes no mercado e bastante conhecidas na área das redes de sensores sem fios. A Tabela 4.1 enumera as plataformas consideradas e o fabricante das mesmas. As Tabelas 4.2 e 4.3 apresentam as características dos dois componentes *hardware* mais importantes das plataformas, o microcontrolador e os módulos de comunicação, respectivamente. Todas as plataformas consideradas operam na banda de frequência ISM 2.4 GHz, utilizam o esquema de modulação O-QPSK e possuem ritmos de transmissão até 250 kb/s (excepto a plataforma Tinynode que opera nas bandas de frequência 868/902 MHz e taxas de 1.2 até 152.3 kb/s).

A maioria das plataformas permite instalar um *Sistema Operativo Real-Time* (RTOS) que auxilia na execução das funções dos microprocessadores, tal como processamento de dados com restrições de tempo e atrasos. Este oferece suporte em *tempo-real*, com arquitecturas simples e com propósitos muito específicos devido às limitações de velocidade de processamento e memória. Plataformas que não utilizem RTOS são mais complexas ao nível da implementação computacional e portanto a (re)instalação de outros serviços pode-se tornar menos amigável. No entanto, estas plataformas podem poupar imensa memória e velocidade de processamento pois não existe desperdício de recursos em funções *middleware*.

Tabela 4.1 – Plataformas de hardware consideradas.

Memsic	
MicaZ [MZ]	Bateria: 2xAA Dimensões: 58x32x7 [mm] Pinout: 51-pin (Entradas Analógicas, I/O Digitais, I2C, SPI, UART)
IRIS [IS]	Bateria: 2xAA Dimensões: 58x32x7 [mm] Pinout: 51-pin (Entradas Analógicas, I/O Digitais, I2C, SPI, UART)
Lotus [LO]	Bateria: 2xAA Dimensões: 76x34x7 [mm] Pinout: 51-pin (Entradas Analógicas, I/O Digitais, I2C, SPI, 3xUART)
Imote2 [IM2]	Bateria: 3xAAA, baterias recarregáveis de Li-Ion ou Li-Ploy Dimensões: 36x48x9 [mm] Pinout: Entradas Analógicas, I/O Digitais, 2xSPI, I2C, 3xUART, SDIO, JTAG, Câmera
TelosB [TB]	Bateria: 2xAA Dimensões: 65x31x6 [mm] Pinout: Entradas Analógicas, I/O Digitais, I2C, SPI, UART
Libelium	
Waspmote [WM]	Baterias: 1150, 2300, 6600 mAh (recarregáveis); 13 Ah (não recarregável); Dois painéis solares (7V). Dimensões: 73.5x51x13 [mm] Pinout: 7 Entradas Analógicas, 8 I/O Digitais, PWM, 2xUART, I2C
Linear Technology	
WirelessHart [WH]	Baterias: 2xAA Dimensões: 37.465x24 [mm] Pinout: 4 Inputs Analógicos e 8 Digitais I/O
Sentilla (antiga Moteiv)	
T-Mote SKY [TMS]	Baterias: 2xAA Dimensões: 32x64.5x6.6 [mm] Pinout: 24-pin (16 de ligação IDC, 8 de ligação JTAG)
Tinynode	
TinyNode 584 [TN]	Baterias: 2xAA (ou 3xAA) Dimensões: 30x40 [mm] Pinout: 6 Entradas Analógicas, 2 Saídas Analógicas, 19 I/O Digitais, LVTTL, UART, SPI
Zolertia	
Z1 [Z1]	Baterias: 2xAA (ou AAA ou 1xCR2032) Dimensões: 56.8x34.5 [mm] Pinout: 52-Pin (I/O Digitais, I2C, SPI, UART)
Texas Instrument	
eZ430-RF2500 [EZ]	Baterias: 2xAA Dimensões: Não referenciado Pinout: 18-Pin (Entradas Analógicas, I/O Digitais, SPI, I2C, UART)
Oracle (SUN)	
SunSPOT [SS]	Baterias: 720 mAh (interna e recarregável) Dimensões: Não referenciado Pinout: 30-Pin (Entradas Analógicas, I/O Digitais, SPI, I2C, USART, I2S)

Tabela 4.2 – Sistema de Processamento das Plataformas.

Plataforma	Chip	RAM [kB]	Velocidade de Processamento [MHz]	RTOS	Estado Activo [mA]	Estado SLEEP [µA]
MicaZ	Atmega 128L 8-bit	4	4	TinyOS MantisOS	8	15
IRIS	Atmega 1281 8-bit	8	8	TinyOS	8	8
Lotus	Cortex M3 32-bit	64	100	TinyOS	50	10
TelosB	TI MSP430 16-bit	10	8	TinyOS MantisOS MansOS	1.8	4.1
T-Mote SKY	TI MSP430F1611 16-bit	10	8	TinyOS ContikiOS	1.8	4.1
Waspmote	Atmega 1281 8-bit	8	8 - 14.7	-	8	8
WirelessHart	Cortex M3 32-bit	72	72	-	2.4	1.2
Imote2	Intel PXA 271 16-bit	256	13-416	TinyOS	31	390
Tinynode	TI MSP430 16-bit	10	8	TinyOS	2.1	6.5
eZ430	TI MSP430F22x2 16-bit	1	8	ContikiOS	2.7	0.9
SunSPOT	AT91RM9200 32-bit	1000	400	-	25	33
Z1	MSP430f2617 16-bit	8	16	TinyOS ContikiOS MansOS	10	0.1

Tabela 4.3 – Sistema de comunicação das plataformas.

Plataforma	Radio	Normas/ Protocolos	Alcance [m]		SLEEP [µA]	TX [mA]	RX [mA]	Sensibilidade [dBm]
			Indoor	*Outdoor*				
MicaZ	CC2420	802.14.4 ZigBee	20-30	75-100	1	17.4 (0 dBm)	19.7	-94
IRIS	AT86RF230	802.14.4 ZigBee 6LoWPAN	>50	>300	0.02	17 (3 dBm)	16	-101
Lotus	AT86RF231	802.14.4 6LoWPAN ZigBee WirelessHART	-	100	0.02	17 (3 dBm)	16	-101
TelosB	CC2420	802.14.4 ZigBee	20-30	75-100	1	17.4 (0 dBm)	23	-94
T-Mote SKY	CC2420	802.14.4 ZigBee	50	>125	1	17.7 (0 dBm)	20	-94

Plataforma	Radio	Normas/ Protocolos	Alcance [m]		SLEEP [µA]	TX [mA]	RX [mA]	Sensibilidade [dBm]
			Indoor	*Outdoor*				
Waspmote	XBee	802.14.4 ZigBee	-	500	0.06	15	15	-92
WirelessHart	LTP5902-WHM	802.14.4 WirelessHart	100	300	1.2	4.4 (0 dBm)	4.5	-95
Imote2	CC2420	802.14.4 ZigBee	~30		1	17.4 (0 dBm)	19.7	-94
Tinynode	Semtech XE 1205	ETSI EN 300 220-1	40	200	1	24.9 (0 dBm)	14.9	-101
eZ430	CC2500	SimpliciTI	>50		0.4	21.2 (0 dBm)	16.6	-87
SunSPOT	CC2420	802.14.4 ZigBee	-	-	1	18 (0 dBm)	20	-90
Z1	CC2420	802.14.4 6LoWPAN	-	-	1	17.4 (0 dBm)	18.8	-

Outro factor importante a considerar é a velocidade de processamento. É de esperar que quanto maior for a capacidade de processamento de uma plataforma, menor eficiência energética ela terá. Algumas plataformas conseguem tecnicamente ter consumos bastante inferiores, e mesmo assim, atingir níveis de processamento bastante aceitáveis, mostrando soluções eficientes. A Figura 4.1 apresenta este factor.

Outro aspecto importante que se deve referir é a contribuição das Texas Instrument (TI) nesta área. Apesar da plataforma eZ430-RF2500 ser a única plataforma da TI a integrar todos os componentes necessários à implementação de redes de sensores sem fios (e não considere a norma do IEEE 802.14.4, utilizando a sua própria norma/protocolo, o SimpliciTI), a TI é fabricante da maioria dos módulos de comunicação que integram as outras plataformas (por exemplo, CC2420) e de grande parte dos microcontroladores para o sistema de processamento (por exemplo, MSP430).

Ao nível do consumo de energia, os módulos de redes de sensores sem fios que possuem os valores de corrente instantânea mais reduzidos são o WirelessHart e o Waspmote. Estas plataformas são bastante eficientes na medida em que os níveis de desempenho computacional e de comunicação que apresentam são aceitáveis mesmo com consumos instantâneos bastante reduzidos.

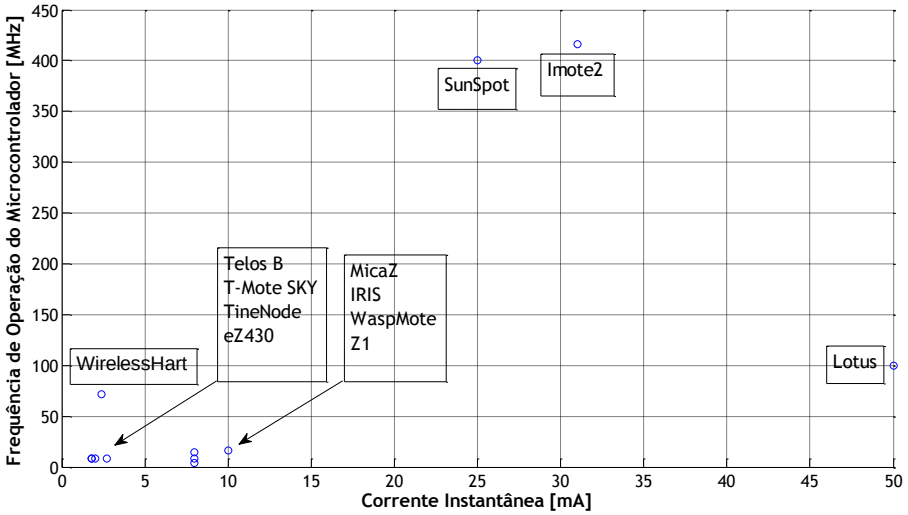

Figura 4.1 - Frequência do microcontrolador versus corrente instantânea de consumo.

4.2. Estimativa do Tempo de Vida

O modelo mais simples e utilizado para determinar o tempo de vida de uma bateria é a Lei de Peukert, que permite definir as relações não-lineares entre o tempo de vida da bateria e a taxa de descarga da mesma, sem ter em consideração o efeito de relaxação ou recuperação [JG08]. De acordo com a Lei de Peukert, *T* (tempo de vida) pode ser obtido dividindo a capacidade (*Q*) por uma corrente de descarga constante (*I*) (Expressão 4.1).

$$T = \frac{Q}{I^k} \tag{4.1}$$

A constante *k* representa a constante de Peukert, obtida experimentalmente para cada bateria. Idealmente, *Q* é igual à capacidade da bateria e *k* é igual a 1. Contudo, na pratica *Q* é uma variável com um valor aproximado da capacidade real e *k* é uma constante maior que 1 (representando as características não-lineares da bateria). Dependendo do tipo de bateria, esta constante *k* tem valores diferentes. Para a maioria das baterias, o valor de *k* está compreendido entre 1.2 e 1.7 [BatStuff]. Ao longo desta Secção, assume-se que *k* é igual a 1.04. A maioria das plataformas de *hardware* para redes de sensores sem fios utilizam duas pilhas AA alcalinas cuja a capacidade total está entre 2500 mAh e 3000 mAh. Assumindo que a capacidade da bateria é $Q_t = 2500\ mAh$ e que a corrente de consumo média é $I = 0.3\ mA$, o tempo de vida total é:

$$T_t = \frac{(2500 * 10^{-3})}{(0.3 * 10^{-3})^{1.05}} = 12501.5\ horas$$

Este resultado mostra que, considerando uma corrente de consumo constante de 0.3 mA e uma bateria com capacidade igual a 2500 mAh, um nó consegue estar ligado durante 12501.5 horas (aproximadamente 521 dias). *Q* pode representar a capacidade total da bateria, Q_t, ou

a capacidade consumida, Q_d após decorrido um tempo de operação, T_d. Considerando ambas as suposições do modelo apresentado em [PSS01] a capacidade remanescente (residual) é calculada através da Expressão 4.2. $Q_t{}'$ representa a capacidade previamente armazenada no intervalo de tempo $T_d < T_t$ e intervalo de capacidade $Q_d < Q_t{}' \leq Q_t$.

$$Q_r = Q_t{}' - Q_d \tag{4.2}$$

A capacidade consumida, Q_d, durante o tempo de intervalo entre T_0 e T_d é dado pela Expressão 4.3, onde T_0 representa o ponto inicial da descarga da bateria.

$$Q_d = \int_{t=T_0}^{T_0+T_d} I(t)dt \tag{4.3}$$

Combinando as Expressões 4.2 e 4.3, a capacidade residual, Q_r é dada pela Expressão 4.4.

$$Q_r = Q_t{}' - \int_{t=T_0}^{T_0+T_d} I(t)dt \tag{4.4}$$

$I(t) = I^k$ é a corrente consumida da bateria durante o tempo de operação T_d. Assim, a capacidade consumida durante esse período de operação é dada pela Expressão 4.5.

$$\begin{aligned} Q_d &= \int_{t=T_0}^{T_0+T_d} I(t)dt = \\ &= I^k t \Big|_{t=T_0}^{T_0+T_d} = \\ &= I^k * (T_0 + T_d) - I^k * (T_0) = \\ &= I^k * T_d \end{aligned} \tag{4.5}$$

A capacidade residual, Q_r, pode ser simplificada como se segue:

$$Q_r = Q_t^{'} - I^k * T_d \tag{4.6}$$

Para que seja realizado um estudo do tempo de vida das plataformas RSSF, foi estimado o consumo de energia médio variando o número de nós vizinhos. Durante cada ciclo de transmissão, existe apenas um nó activo que possui sempre uma trama que necessita de ser enviada, enquanto os outros nós vizinhos apenas aceitam as tramas e respondem com ACKs.

A Figura 4.2 apresenta o modo de acesso básico da norma IEEE 802.14.4 assumindo um ciclo de tempo activo (*duty cycle*) com comprimento fixo de 1 %, composto pelos seguintes períodos:

- Período de SLEEP;
- Tempo de transição entre o estado de SLEEP e estado de recepção;
- Fase de *back-off*;
- Mecanismo CCA;
- Tempo de transição entre estado de recepção e estado de transmissão;
- Tempo de transmissão da trama de dados;
- Tempo de transição entre o estado de transmissão e estado de recepção;
- Tempo de recepção do ACK;
- Tempo de transição entre o estado de recepção e o estado SLEEP;
- Novo período de SLEEP.

Figura 4.2 – Modo de acesso básico da norma IEEE 802.14.4.

Estes períodos estão descritos no Capítulo anterior. A Tabela 4.4 apresenta os parâmetros e especificações típicas dos módulos de comunicação segundo a norma IEEE 802.14.4.

Tabela 4.4 - Parâmetros e valores típicos da norma IEEE 802.14.4.

Descrição	Símbolo	Valor
Duração do período de back-off	T_{BO}	320 µs
Tempo de detecção CCA	T_{CCA}	128 µs
Tempo de transição TX/RX e RX/TX	T_{TA}	192 µs
Tempo de transmissão de uma trama ACK	T_{ACK}	544 µs
Tempo de transmissão de uma trama de dados com máximo comprimento	T_{DATA}	4256 µs
Tempo de inicialização dos rádios para os estados RX ou TX	$rxSetupTime$	1792 µs
Tempo de *Interframe Spacing* (IFS)	T_{IFS}	640 µs
***Overhead* da camada PHY**	L_{H_PHY}	6 bytes
***Overhead* da camada MAC**	L_{H_MAC}	9 bytes
Comprimento máximo de uma trama de dados	L_{DADOS}	118 bytes
Comprimento da trama ACK	L_{ACK}	11 bytes
Taxa de transmissão (*Data Rate*)	R	250 kb/s

Assume-se sempre o pior caso para o *back-off*, isto é, a janela de contenção máxima (CW_{max}) calculada através da Expressão 4.7 e com um expoente de *back-off* igual a 3 ($BE = 3$):

$$CW_{max} = (2^{BE} - 1) * T_{BO} = (2^3 - 1) * 320 = 2240\ \mu s \tag{4.7}$$

A janela de contenção média, dada pela Expressão 4.8, é igual a:

$$\overline{CW} = \frac{CW_{max}}{2} = \frac{2240}{2} = 1120\ \mu s = 1.12\ ms \tag{4.8}$$

O tempo de atraso devido ao procedimento CCA é dado pela Expressão 4.9:

$$ccaTime = rxSetupTime + T_{CCA} = 1920\ \mu s \tag{4.9}$$

Por fim, após a verificação do estado do canal numa duração compreendida de 8 símbolos (128 μs), os dados são enviados e uma trama ACK recebida, assumindo sempre que estamos na presença de um canal ideal sem erros. A Expressões 4.10 e 4.11 apresentam a duração em milissegundos da trama de dados (comprimento máximo, 127 bytes) e de uma trama ACK, respectivamente.

$$T_{DATA} = 8 * \frac{L_{H_{PHY}} + L_{H_{MAC}} + L_{DADOS}}{R} \tag{4.10}$$

$$T_{DATA} = 8 * \frac{6 + 9 + 118}{250000} = 4.256\ ms$$

$$T_{ACK} = 8 * \frac{L_{H_{PHY}} + L_{ACK}}{R} \tag{4.11}$$

$$T_{ACK} = 8 * \frac{6 + 11}{250000} = 0.544\ ms$$

A média do tempo de atraso mínimo, D_{min}, para transmitir um pacote e receber com sucesso um ACK, é dado pela Expressão 4.12.

$$D_{min} = \overline{CW} + ccaTime + T_{TA} + T_{DATA} + T_{TA} + T_{ACK} + T_{IFS} = 8.864\ ms \tag{4.12}$$

Para utilizar a Lei de Peukert para determinar o tempo de vida das plataformas, foi calculada a corrente de consumo média tendo em conta todos os estados possíveis dos módulos de comunicação e processamento. Assumindo que o nó emissor em questão envia um número de

tramas de dados igual ao número de nós directamente alcançáveis na sua vizinhança, a corrente média de consumo é dada pela Expressão 4.13.

$$\bar{I} = \left[i_{tx} * d_{tx} + i_{rx} * d_{rx} + i_{Rsleep} * d_{Rsleep} + i_{uPAct} * d_{uPAct} + i_{uPSleep} * d_{uPsleep}\right] * N \qquad (4.13)$$

O significado dos símbolos e variáveis da Expressão (4.13) é apresentado na Tabela 4.5.

Tabela 4.5 - Notação para determinar a corrente média consumida.

Descrição	Símbolo
Corrente do microcontrolador no estado activo	i_{uPAct} [mA]
Corrente do microcontrolador no estado SLEEP	$i_{uPSleep}$ [mA]
Corrente do módulo de comunicação no estado TX	i_{tx} [mA]
Corrente do módulo de comunicação no estado RX	i_{rx} [mA]
Corrente do módulo de comunicação no estado SLEEP	i_{Rsleep} [mA]
Tempo no estado TX	d_{tx} [%]
Tempo no estado RX	d_{rx} [%]
Tempo no estado SLEEP	d_{sleep} [%]
Tempo no estado activo do microcontrolador	d_{uPAct} [%]
Tempo no estado SLEEP do microcontrolador	$d_{uPsleep}$ [%]
Corrente média consumida	$\bar{I}$ [mA]
Número de nós vizinhos	N [Units]

A Tabela 4.6 apresenta as correntes médias consumidas, $\bar{I}$, para cada plataforma e para diferentes números de nós vizinhos, *N*.

Tabela 4.6 - Corrente média consumida de cada plataforma.

N	1	2	3	4	5	6	7	8
MicaZ [mA]	0.291	0.581	0.872	1.163	1.454	1.74	2.04	2.33
Iris [mA]	0.25	0.5	0.75	1	1.25	1.5	1.75	2
Lotus [mA]	0.672	1.34	2.02	2.69	3.36	4.03	4.7	4.37
TelosB [mA]	0.263	0.526	0.788	1.05	1.314	1.58	1.84	2.1
WirelessHart [mA]	0.072	0.143	0.215	0.287	0.359	0.431	0.503	0.574
Waspmote [mA]	0.231	0.462	0.693	0.924	1.155	1.386	1.62	1.85
T-Mote [mA]	0.219	0.438	0.657	0.876	1.095	1.314	1.533	1.752
TinyNode [mA]	0.173	0.346	0.519	0.693	0.866	1.039	1.212	1.39
eZ43 [mA]	0.218	0.44	0.655	0.874	1.092	1.31	1.53	1.75
Imote2 [mA]	1.05	2.09	3.14	4.18	4.23	6.28	7.32	8.37
SunSpot [mA]	1.073	2.145	3.22	4.29	4.36	6.44	7.5	8.58
Z1 [mA]	0.29	0.58	0.86	1.15	1.44	1.73	2.02	2.3

Assume-se inicialmente que um ciclo activo é igual a 1 % do tempo (módulos de comunicação e microcontrolador no estado activo durante 1 % do tempo e no estado de SLEEP durante 99 % do tempo). Para 1 % do tempo no ciclo activo, todos os elementos da plataforma estarão activo durante cerca de 48.9 ms. Após realizadas todas as operações necessárias (emissão de dados, recepção de ACK, processamento dos mesmos), todos os elementos transitarão e permanecerão no estado SLEEP durante aproximadamente 4.84 segundos, correspondente a 99% do tempo, até regressarem novamente às suas funções dos estados activos. A Figura 4.3 apresenta os resultados do tempo de vida de cada plataforma medido em dias.

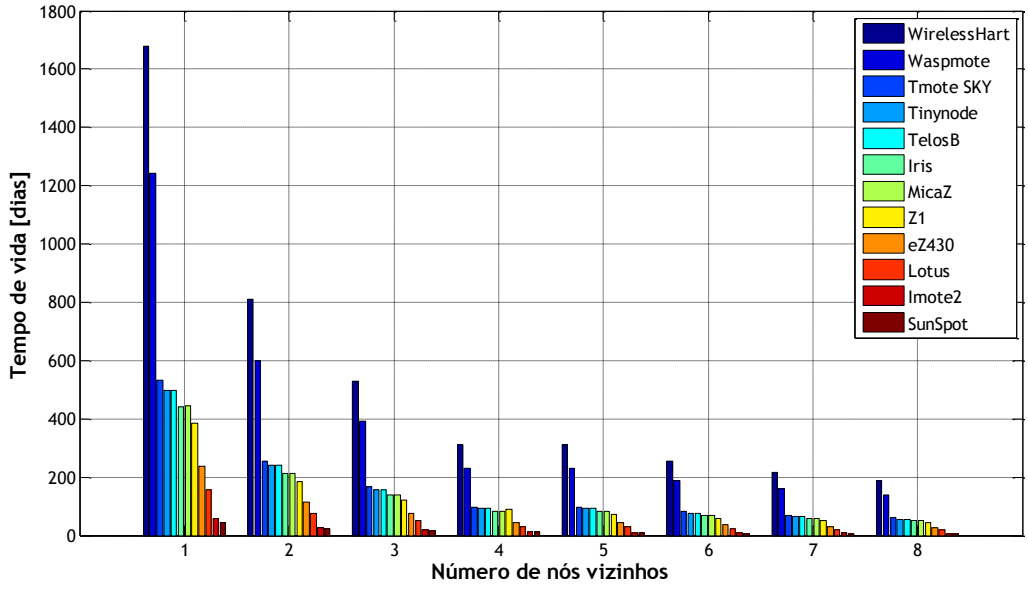

Figura 4.3 – Tempo de vida das plataformas.

A plataforma WirelessHart prova ter o consumo mais eficiente. Os resultados demonstram que esta plataforma possui o tempo de vida mais longo quando comparada com todas as outras. A plataforma Waspmote tem o segundo melhor resultado em termos energéticos devido, não só ao seu consumo de energia reduzido mas igualmente devido às suas próprias baterias com elevada capacidade (foi considerada a utilização da bateria com capacidade 6600 mAh). A plataforma Tmote SKY tem o terceiro melhor consumo. Apesar de possuir valores instantâneos mais ou menos elevados ao nível da comunicação, tem dos valores mais reduzidos no estado activo e SLEEP do microprocessador.

Com os piores desempenhos energéticos destacam-se o Lotus, o Imote2 e o SunSpot devido aos consumos de energia elevados. Note-se que a bateria considerada para o SunSpot possui uma capacidade de apenas 750 mAh, considerando que a plataforma não pode ser alimentada por nenhum outro tipo de fonte de alimentação a não ser a sua própria bateria. Da mesma maneira, o Imote2 é alimentado por 2 pilhas AAA e possui um elevado consumo o que se traduz numa eficiência energética mais reduzida.

O eZ430 tem um tempo de vida curto apesar da sua boa eficiência energética, devido à utilização de pilhas AAA cuja capacidade é menor (1200 mAh) que a das pilhas AA.

A Figura 4.4 apresenta o impacto do ciclo de tempo activo, quando variado no intervalo de [1 10] %, referente à eficiência do consumo energético. São consideradas as duas plataformas mais rentáveis energeticamente e a plataforma IRIS. O ciclo de tempo activo é aumentado essencialmente quando existem grandes quantidades de dados a serem transmitidos/recebidos e/ou processados num curto intervalo de tempo. Portanto, aumentar o ciclo de tempo activo traduz-se numa maior participação e interacção dentro da rede.

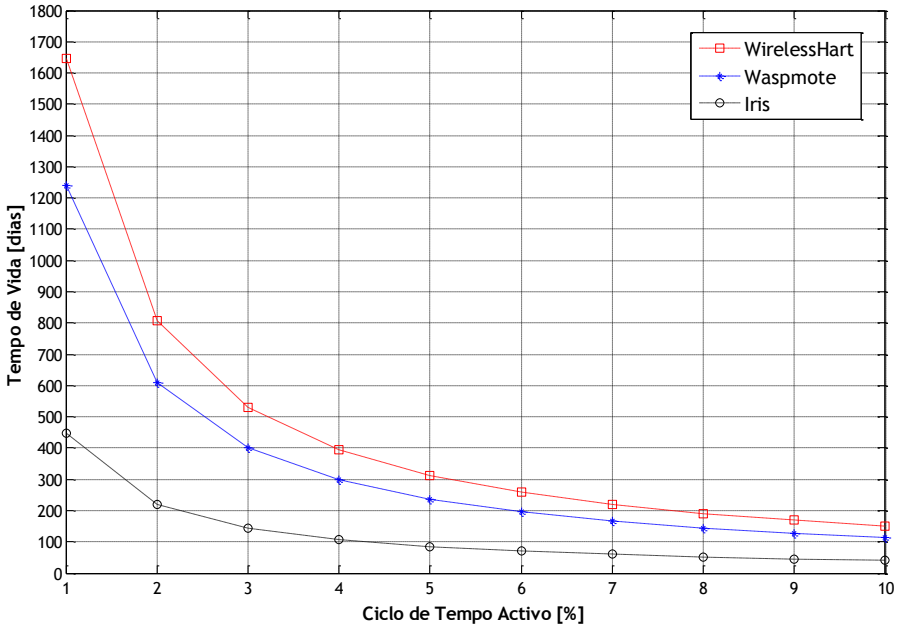

Figura 4.4 – Variação do ciclo de tempo activo.

Simultaneamente, um nó que permaneça mais tempo no estado activo pode receber e partilhar pacotes de controlo topológico, encaminhamento, sincronização e permanecer totalmente ciente da sua posição na rede. Aplicações que sejam bastante rigorosas nestes aspectos terão necessariamente de aumentar o ciclo de tempo activo, sacrificando o tempo de vida útil dos nós. No entanto, quanto maior for o ciclo de tempo activo, menor será o impacto energético, apesar de existirem diferenças notáveis para valores de *duty cycle* muito reduzidos. A análise e a escolha de uma plataforma para uma determinada aplicação deve-se basear nos equilíbrios (*trade-offs*) que melhor se adaptam a essa mesma aplicação.

4.3 Sumário e Conclusões

Neste Capítulo foram apresentadas diversas plataformas *hardware* actualmente disponíveis no mercado. Estas plataformas, compostas principalmente por módulos simples de microprocessamento, comunicação e *sensor boards*, são dispositivos de potência e capacidade reduzida, desenhados para tarefas e funções muito específicas e maioritariamente muito

simples. Do grupo de fabricantes principais, a Memsic é o fabricante com maior número destes dispositivos. Ao longo da sua história tem vindo a actualizar os seus produtos com componentes cada vez melhores. Estes dispositivos, tais como o MicaZ e TelosB, são bastante referenciados e utilizados, mas através do aparecimento de novas arquitecturas de microprocessadores mais poderosos, de 32-bit, plataformas como o Lotus foram concebidas para dar lugar a melhores níveis de desempenho computacional. A contrapartida está precisamente no desperdício de energia que estas plataformas têm com componentes mais poderosos, tornando-os energeticamente pouco eficientes. Por contraste, plataformas mais recentes como o WirelessHart utilizam também estes chips de 32-bits e conseguem atingir níveis de consumo muito mais reduzidos, inclusive mais reduzidos que plataformas com microprocessadores mais fracos, tornando-se fortes escolhas para a implementação destas redes. O WirelessHart é referenciado como tendo um funcionamento eficaz em parceria com o próprio protocolo WirelessHart e a norma IEEE 802.14.4. Contudo a pilha protocolar WirelessHart é bem mais complexa que outras (por exemplo, ZigBee), e portanto, com a ausência de qualquer RTOS, é necessário conhecimento sólido para que seja eficientemente implementado.

Plataformas como o SunSpot e o Imote2 por outro lado apresentam os maiores consumos de energia entre todas as plataformas. As suas elevadas capacidades de processamento destinam-se a aplicações cujo desempenho computacional é exigente, tal como aplicações de vídeo e voz. Por exemplo, o Imote2, além de ter instalado *codecs* de áudio (AC97), permite facilmente a integração de câmaras com mecanismos de processamento de imagem. O SunSpot por outro lado é uma plataforma com uma eficiência energética muito "comprometedora", mas ao contrário da maioria suporta protocolos bastante eficientes, ligação com a internet, elevado grau de segurança e desenhada para todo o tipo de aplicações, desde as mais elementares até às mais completas. Devido aos elevados consumos energéticos, ambas as plataformas possuem componentes eficientes para o recarregamento das suas baterias por recolha de energia do ambiente.

A plataforma Waspmote é uma plataforma com um módulo de processamento simples, essencialmente para aplicações de monitorização e automação. Goza igualmente de uma eficiência energética adequada. Com um dos consumos mais reduzidos, possui baterias com elevadas capacidades e facilmente recarregáveis por recolha de energia. A principal vantagem da plataforma Waspmote é a possibilidade de implementar facilmente qualquer aplicação. Com um código fonte aberto e inúmeras bibliotecas específicas para aplicações muito concretas, torna-se extremamente fácil de programar. A acompanhar, a Libelium prova possuir um vastíssimo leque de sensores e actuadores, concebidos e desenhados especificamente para os Waspmotes, tornando-se acessível e fácil a interligação desses componentes electrónicos com a plataforma.

Plataformas como a Tinynode, T-Mote SKY, Z1 e a generalidade da Memsic são escolhas adequadas para aplicações simples e cujo custo é um factor importante a considerar na implementação da rede. A aderência ao longo dos anos destas plataformas fomentou o aparecimento de outros RTOSs para além do TinyOS como, por exemplo, o ContikiOS e a sua posterior compatibilidade com o *hardware* em questão. A vantagem em utilizar estas plataformas é essencialmente a diversidade de estudos e testes realizados com as mesmas. Propostas de inovações de normas e novos protocolos foram largamente implementadas nestas plataformas como são descritos em vários trabalhos na literatura [AMP11], [DGAZ10], [DFMM06], [CLPKEAH11], [CCG12], [AGB10] e [LXCDW11].

Presentemente, devido principalmente às limitações do próprio *hardware* e apesar da larga evolução da última década, o conceito de *smart dust* (pó inteligente) continua a estar praticamente por ser realizado. As dimensões mais reduzidas são inferiores à palma de uma mão. As fontes de energia participam igualmente e activamente na questão da miniaturização dos dispositivos. Mesmo que nos próximos anos seja possível atingir elevados níveis de processamento e eficiência nas comunicações com componentes *hardware* miniaturizados abaixo da escala do milímetro, não se prevê o mesmo para as fontes de alimentação. Em contraste com o conceito *smart dust*, as plataformas apresentam tamanhos consideravelmente reduzidos. Contudo, com a ligação da fonte de alimentação, a plataforma final apresenta, na maioria dos casos, o dobro do tamanho.

Capítulo 5

Camada de Rede em RSSF

Inicialmente, as camadas superiores às duas primeiras camadas definidas pela norma IEEE 802.15.4 foram sugeridas pela aliança ZigBee [F08]. A aliança foi formada com o objectivo de desenhar uma camada de rede e aplicação, eficientes e direccionadas para as redes de sensores sem fios baseadas na norma IEEE 802.15.4 [IEE03].

A ZigBee define uma estrutura e um conjunto de regras, directamente compatíveis com a norma IEEE 802.15.4, de extrema importância para o funcionamento correcto e eficiente das RSSF [F08]. A camada de rede NWK (*Network*) é responsável pela selecção da topologia de rede, formação da rede e criação de encaminhamentos. Define três tipos de dispositivos diferentes ao nível da rede: *Dispositivo-Final ZigBee*, *Router ZigBee* e *Coordenador ZigBee*. O primeiro tipo engloba a maioria dos nós existentes na rede e tem correspondência com o grupo RFD da norma IEEE 802.15.4, cujos nós apenas conseguem emitir os seus dados, sem capacidade para participar como nós intermédios no encaminhamento de pacotes. O segundo tipo pertence ao grupo FFD. O terceiro tipo, também FFD, é o coordenador da rede. Ao nível da rede, este coordenador tem a função de criar e formar a topologia, além de atribuir endereços de rede compatíveis com os endereços MAC 16-bit. As funções da camada NWK são semelhantes às funções das camadas de rede de qualquer pilha protocolar: descoberta, manutenção e reparação de encaminhamentos.

A camada de aplicação APL (*Application Layer*) é dividida em três secções: APS (*Application Support Sublayer*), ZDO (*ZigBee Device Object*) e a *Application Tramawork* [F08]. Apesar das normas iniciais ZigBee não incluírem ligações IPv6, a versão 2 (*ZigBee IP Smart Energy 2.0*) foi disponibilizada em 2013 com a inclusão dos protocolos 6LoWPAN, RPL e UDP, que serão descritos nas próximas Secções.

5.1 Caracterização de Protocolos de Encaminhamento

Os protocolos de encaminhamento para aplicações de RSSF são importantes em redes muito densas que cobrem grandes áreas, especialmente em aplicações com um número muito restrito de *gateways* disponíveis. Aplicações com estas características levam à existência de nós fora do raio de alcance dos coordenadores da rede e *gateways*, e vice-versa. Como tal, conduzem à necessidade de introduzir protocolos de encaminhamento de dados, metadados e informação. Alguns trabalhos sugerem soluções sem a necessidade de introduzir protocolos de encaminhamento. Os autores em [TM09] sugerem a utilização de *gateways* móveis para aumentar a eficiência energética da rede. Outros trabalhos instalam as *gateways* em posições óptimas e ideais dentro da rede, calculados através de algoritmos complexos [PHCSS03].

Os protocolos da camada de rede são divididos em 2 grupos: *Distance Vector* e *Link-State*. O primeiro atribui valores de custo a cada nó da rede baseado em métricas usuais como a contagem do número de saltos (*hop count*) até ao nó destino. Este grupo utiliza o algoritmo *Bellman-Ford* para calcular os caminhos. O segundo grupo, por sua vez, atribui valores de custo de encaminhamento às ligações existentes entre os nós. As métricas para o segundo grupo baseiam-se normalmente na largura de banda e no ritmo de transmissão de dados. O algoritmo utilizado é o *Dijkstra*, que é mais rápido do que o *Bellman-Ford*.

Os protocolos *Distance-Vector* calculam o melhor caminho entre os nós. Protocolos mais antigos enviam periodicamente as suas tabelas de encaminhamento para os nós vizinhos. Alguns protocolos mais recentes apenas emitem a parte da tabela que é alterada e só o fazem quando existe de facto uma alteração. O objectivo final é conseguir que todos os nós tirem proveito da informação enviada pelos nós vizinhos sem ser necessário conhecerem na íntegra toda a topologia da rede.

Os protocolos *Link-State*, ao contrário dos *Distance-Vector*, constroem uma tabela de topologia idêntica para todos os nós através da qual é possível mapear toda a rede. Caracterizam-se por formarem hierarquias na rede e apenas enviarem informação sobre a rede aos nós adjacentes quando detectadas inconsistências nos encaminhamentos. Normalmente mantêm tabelas de encaminhamento, de vizinhança e de topologia.

Devido ao funcionamento complexo do segundo grupo de protocolos, os protocolos *Distance-Vector* são extremamente utilizados em redes sem fios, nomeadamente em redes Wi-Fi e *Mobile Ad Hoc Network* (MANET). Alguns protocolos, frequentemente citados na literatura e descritos em [RT99], são o *Destination-Sequenced Distance-Vector* (DSDV), o *Clusterhead Gateway Switch Routing* (CGSR), o *Ad Hoc On-Demand Distance Vector Routing* (AODV), o *Dynamic Source Routing* (DSR), entre outros. Com o DSDV os nós propagam periodicamente uma parte ou toda a informação da sua tabela de encaminhamento, com um mecanismo de sequências que evita problemas de caminhos fechados (*loops*). O CGSR organiza hierarquias na rede, com os *Cluster Heads* (CH), com a implementação do DSDV em simultâneo. Devido às restrições espectrais das redes sem fios, pode-se concluir que, em redes densas, ambos os protocolos podem causar problemas quando os "anúncios", que contêm informação sobre a rede, ocupam uma considerável parte da largura de banda. Os protocolos *On-Demand*, como o AODV e o DSR, evitam este problema ao minimizarem a difusão (*broadcast*) de controlo e sinalização. Basicamente, um nó que pretende comunicar para outro nó ou com uma rede externa, fora do seu alcance, emite *pacotes de descoberta* que serão sequencialmente transmitidos ao longo do(s) caminho(s) até ao nó destino. Todos os nós intermédios armazenam em memória o endereço do nó que emitiu a primeira cópia desse *pacote de descoberta*. A resposta a este pacote, proveniente do nó destino, seguirá o caminho inverso mais curto até ao nó de origem inicial. Por de trás do processo todos os nós constroem as suas

tabelas de encaminhamento através da partilha de *pacotes de descoberta* e da criação destes caminhos.

A implementação de protocolos de encaminhamento implica que haja uma reavaliação cuidadosa das restrições energéticas e espectrais das RSSF. Os protocolos mais dinâmicos são eficazes ao nível da topologia de rede e fiabilidade das comunicações. No entanto, o tempo de vida da rede é inevitavelmente afectado, assim como a largura de banda devido ao *overhead* (extra) com a constante troca e processamento de pacotes de controlo. Apesar das fortes semelhanças com as redes *ad hoc*, a gestão destes recursos deve ser necessariamente mais eficiente em RSSF.

Os protocolos de encaminhamento necessitam portanto de dar resposta a estes desafios, como descrito em [AK04] e [AY03]:

- **Dinâmicas e distribuição dos nós** – O desenvolvimento da rede e a posição dos nós são dois factores com duas vertentes diferentes: determinístico e não determinístico. A distribuição determinística dos nós tem por norma encaminhamentos pré-determinados. A distribuição não-determinística caracteriza-se pela aleatoriedade do desenvolvimento e da posição dos nós. Este tipo de implementação apresenta desafios mais complexos como a capacidade de auto-organização da rede, a necessidade de existirem técnicas eficientes de *clustering* e o posicionamento de *gateways*, sem comprometer o desempenho alcançado pelos protocolos das camadas inferiores. O nível de dinamismo das aplicações também é crucial. Quanto mais aleatórios ou críticos forem os eventos, ou quanto maior for a geração de dados (e consequente tráfego), maior terá de ser a robustez e a fiabilidade do encaminhamento *multihop*. O dinamismo da rede está também intrinsecamente ligado a aplicações com nós sensores móveis. A mobilidade aumenta a complexidade dos protocolos quando comparada com redes estacionárias. Aplicações desta natureza como, por exemplo, eventos de *tracking*, necessitam sempre de encontrar novos pontos óptimos de estabilidade e consistência ao longo do tempo numa determinada área, mesmo que inevitavelmente haja degradação no tempo de vida da rede. O objectivo passa sempre pela optimização da gestão dos recursos seguindo os equilíbrios específicos de cada aplicação.

- **Consumo de energia** – O consumo de energia dos nós é o desafio mais debatido a todos os níveis nas RSSF. A implementação básica de protocolos de rede sugere sempre um acréscimo no consumo de energia devido à troca de pacotes de controlo entre os nós, portanto existe mais *overhead* e mais processamento de tramas. Em redes multi-salto (*multihop*), além das suas funções como fontes geradoras de dados, grande parte dos nós sensores poderão vir a participar como intermédios, o que implica permanecerem mais tempo no estado activo (maiores *duty cyles*) para que

todos os pacotes de dados sejam efectivamente encaminhados. Esta consideração acrescenta a forte probabilidade dos nós intermédios ficarem sem energia mais rapidamente do que o resto da rede, criando buracos de encaminhamento (zonas mortas). Os protocolos devem ter em conta as características energéticas actuais destes nós, como a energia residual ou a taxa de consumo energético, de maneira a equilibrar o consumo da rede no geral. Alguns trabalhos propõem soluções para estes problemas, tal como a detecção e eliminação de cópias de pacotes com dados idênticos, a agregação de dados, e a utilização de métricas de energia no cálculo dos melhores encaminhamentos. Apesar dos desafios energéticos, os protocolos de rede podem, por outro lado, optimizar o consumo de energia em determinadas aplicações e situações. Um exemplo clássico é a existência de obstáculos e/ou de grandes distâncias entre nós e *gateways*. Sem a existência de protocolos de encaminhamento, para que a comunicação directa entre o nó emissor e o nó receptor (comunicação com apenas um salto) seja possível e corretamente realizada, a potência de emissão deve aumentar, o que implica um aumento do consumo de energia.

- **Escalabilidade e tolerância a faltas** – Os protocolos de encaminhamento podem ajudar a aumentar a escalabilidade da rede. Quer isto dizer que novos nós podem ser adicionados a uma rede já antes desenvolvida. Além disso, a adição de novos nós não implica a adição de mais *gateways*, desde que os encaminhamentos se mantenham consistentes, actualizados, e os mecanismos de descoberta sejam eficientes. A ampliação da rede pode levar a um aumento da probabilidade de ocorrerem falhas. Implementar uma rede multi-salto deve ser uma escolha segura e bem analisada seguindo as características e equilíbrios (*tradeoffs*) requeridos para cada aplicação. Os protocolos de encaminhamento utilizados devem além disso possuir mecanismos de redundância de encaminhamentos, ou a própria rede possuir nós redundantes quando outros ficam incontactáveis (devido a questões energéticas ou limites de memória e *buffer*).

- **Qualidade de Serviço** – As aplicações diferenciam-se pelos seus níveis de qualidade de serviço (QoS). Algumas aplicações toleram melhor a latência na entrega dos dados do que outras. Por exemplo, a monitorização da luminosidade de uma sala não é uma aplicação crítica com dados urgentes, mas a monitorização de incêndios ou fracturas de infraestruturas têm obrigatoriamente que alertar os seus eventos logo que sejam detectados. Nestes exemplos, o atraso destes alertas deve ser mínimo. A primeira observação a fazer quanto a este desafio é relativamente ao número de nós que participa no encaminhamento. Quanto maior é o número de nós por onde o pacote é encaminhado, maior será o atraso da sua entrega. Por outro lado, se o protocolo contar apenas o número de saltos do encaminhamento, selecionando o que tiver menor número, é provável que as ligações escolhidas tenham piores desempenhos, tal

como débito binário reduzido e fiabilidade reduzida, devido à existência de distâncias elevadas entre emissores e receptores. Em termos de latência, a retransmissão de pacotes perdidos pode, em várias situações, ser pior do que a participação de vários nós, mais próximos uns dos outros, num único encaminhamento, reduzindo a média de pacotes não recebidos e retransmitidos, oferecendo portanto desempenho superior.

Os protocolos não podem, apesar de tudo, ser muito complexos ou dinâmicos. O nível de complexidade dos seus funcionamentos está sempre seriamente restringido às características das LR-WPAN.

Variadíssimos trabalhos sobre RSSF dividem os protocolos em diversas categorias, dependendo das características que os autores consideram. No geral, os protocolos são divididos nas seguintes categorias: *Data-centric, Location/Geographic-based,* Hierárquicos e QoS-*based*. Segue-se a descrição detalhada dessas mesmas categorias:

- ***Data-Centric*** – A categoria *data-centric* parte do pressuposto que os nós não possuem, na sua globalidade, endereços únicos que os identifiquem. Consequentemente, a transmissão dos mesmos pacotes de dados entre nós intermédios pode-se tornar extremamente redundante. Como este cenário é impraticável, existem esquemas de agregação de dados que utilizam a solicitação de dados e eliminam a redundância de pacotes. Os dois protocolos mais importantes desta categoria são o *Sensor Protocols for Information Negotiation* (SPIN) e o *Direct Diffusion*. O SPIN é um protocolo *handshaking* de 3 estágios, isto é, as comunicações são realizadas através de três tipos de mensagens: o ADV que alerta a existência de novos dados, o REQ que faz o pedido dos dados e o pacote de dados (DATA). O *Direct Diffusion* utiliza um esquema de encaminhamento mais robusto e, mesmo assim, eficiente. A *gateway* propaga pacotes de controlo denominados "*interesse*". Estes pacotes de "interesse" definem o tipo de dados requisitados através de uma lista de valores, tais como nomes de objectos, duração e área geográfica [AY03]. Com esses pacotes de "interesse" é possível solicitar dados e eliminar redundância quando os pacotes são comparados com os valores dos interesses. À medida que os "interesses" vão sendo propagados, os nós intermédios atribuem *gradientes* que representam o custo da ligação para o nó emissor do "interesse". A criação de gradientes concretiza a construção de encaminhamentos. Quando os nós fontes disseminam os dados pela rede, os pacotes são encaminhados através das ligações com os gradientes mais baixos.

- ***Geographic-based*** – Como o próprio nome indica, estes protocolos baseiam-se na localização geográfica dos nós e da rede através da utilização de um GPS de baixo consumo. A presença deste tipo de informação pode auxiliar em decisões de

propagação dos dados, quando apenas se pretende difundir para uma determinada região. De forma similar, se os nós souberem as suas localizações, podem calcular a distância para o receptor e fazer uma estimativa do consumo energético [AY03]. Protocolos como o *Minimum Energy Communication* Network (MECN) calculam determinadas distâncias e formam regiões nas quais os nós dentro duma região são potenciais intermédios. Todas as comunicações destinadas a nós que estejam para lá dos limites da região são consideradas pouco eficientes, pouco fiáveis e portanto não realizáveis. O custo dos encaminhamentos é então obtido dentro das regiões de cada nó. O *Geographic Adaptive Fidelity* (GAF) cria uma grelha virtual na rede [XHE01]. Os nós, com informação GPS, calculam as suas posições nessa grelha. Dois nós no mesmo ponto da grelha são considerados equivalentes em termos de custo de encaminhamento. A partir desse pressuposto, esses dois nós determinam a localização dos nós vizinhos e criam encaminhamentos redundantes que irão ser utilizados alternadamente. Por exemplo, enquanto um nó intermédio recebe e reencaminha dados, o nó vizinho com o mesmo custo permanece em SLEEP. Na difusão seguinte, será o nó anteriormente adormecido que permanecerá activo para reencaminhar os dados. Este balanceamento de tráfego consegue aumentar o tempo de vida da rede [XHE01]. Quanto maior for a redundância de nós, maior será a eficiência energética. No entanto, quanto maior for o número de nós, maior será o custo de desenvolvimento da rede. Por fim, estes protocolos facilitam ainda outra característica muito particular: a mobilidade dos nós.

- ***Hierárquicos*** – Os protocolos hierárquicos baseiam-se na criação de aglomerados (*clusters*). Como já visto, os *clusters* são grupos de nós, hierarquicamente distribuídos, bastante eficazes para grandes áreas e implementações densas. As comunicações são essencialmente realizadas por *multihop* e esquemas de agregação de dados. Os *clusters* são criados a partir de nós denominados *Cluster Heads* (CH), selecionados para coordenarem esses grupos de nós. Os protocolos mais importantes desta categoria são o *Low-Energy Adaptive Clustering Hierarchy* (LEACH), *Hybrid Energy Efficient Distributed* (HEED), *Power-Efficient Gathering in Sensor Information Systems* (PEGASIS), *Threshold-sensitive Energy Efficient* (TEEN), entre outros. Os protocolos hierárquicos têm particularidades únicas de funcionamento que os destinguem, mas comumente todos têm em consideração a maximização da eficiência energética. O funcionamento básico de um protocolo hierárquico passa primeiro pela selecção do CH que coordenará o *cluster*, dando permissões, respondendo a pedidos e encaminhando os dados para *gateways*. O LEACH utiliza uma função que decide quais os nós pretendentes a CH e quais dos pretendes irá ser realmente um CH. Esta decisão é retomada periodicamente e outro nó pode assumir o papel de CH. Este é um exemplo básico de balanceamento do consumo de energia na rede, dado que o CH é o principal agregador de dados que reencaminha os mesmos para fora da rede. Os

esquemas hierárquicos auxiliam na implementação de métodos secundários mais robustos, tais como a identificação de grupos de nós ou estabelecimento de regiões a partir do tipo de dados, do tipo de tráfego ou devido a restrições adicionais e muito particulares.

- ***QoS-Based*** – A criação de encaminhamentos na rede considera métricas para calcular os custos das ligações. Protocolos baseados na qualidade de serviço e no fluxo consideram como métricas o débito binário e o atraso extremo-a-extremo dos pacotes de dados. Nas RSSF, a função de custo *QoS-Based* junta a estas duas métricas e o consumo de energia para construir caminhos na rede, como foi proposto em alguns trabalhos [FASL01]. Estes são protocolos do tipo *Distance-Vector*. O *Sequential Assignment Routing* (SAR) é um exemplo de um protocolo que cria árvores de encaminhamento múltiplo através de métricas de QoS, energia e nível de prioridade de cada pacote de dados. Os caminhos múltiplos são mantidos em tabelas em cada nó participante, armazenando em memória os endereços dos nós vizinhos e os custos das ligações para estes. Como em qualquer esquema de encaminhamento, as comunicações extremo-a-extremo entre nós fontes de dados e *gateways* são feitas ao longo dos caminhos com os custos mais reduzidos. Estes protocolos são extremamente robustos, com capacidade moderada de evitarem/resolverem falhas (por exemplo, caminhos/ciclos fechados), apesar de introduzirem algum *overhead*.

A escolha do protocolo dependerá sempre do tipo de aplicação desenvolvida. Protocolos *data-centric* fazem solicitações e realizam a agregação de dados. Eliminam a redundância, têm resultados adequados em aplicações móveis e eficiência energética adequada. No entanto, não possuem caminhos múltiplos redundantes, são pouco escaláveis, não consideram a qualidade do tráfego e do fluxo dos dados, e podem demorar a responder às solicitações, o que os torna excelentes candidatos para aplicações muito simples, onde a troca de dados não é necessariamente urgente.

Alguns protocolos hierárquicos conseguem ser muito escaláveis e oferecer níveis elevados de eficiência energética, mas continuam a depender de funções e esquemas com alguma nível de complexidade, tomadas de decisão frequentes e alterações globais em *clusters* (apesar de poderem fazer apenas reparações locais). Estes desafios podem ser facilmente ultrapassados se os CH forem nós com capacidades superiores aos restantes, tal como tempos de vida mais longos e desempenhos mais elevados ao nível do processamento.

Apesar das vantagens descritas, os protocolos baseados na localização dos nós não são na sua maioria aplicáveis em RSSF, visto que utilizam funções matemáticas complexas e implicam consumo de energia adicional em tecnologias GPS. Existem outras categorias e esquemas bem mais específicos do que os enumerados aqui, alguns com o objectivo de resolverem determinados desafios ou baseados em equilíbrios mais concretos.

Estes protocolos foram desenhados para responder às restrições impostas pelas RSSF, e realizam simplesmente a construção dos encaminhamentos dos dados. Até aqui, a interligação directa com redes externas, principalmente com a Internet, era algo impensável e impraticável, devido às mesmas restrições. Assim, os endereçamentos numa rede não tinham que ser globalmente únicos. Eram apenas locais. Isto significa que dois nós na mesma rede tinham que ter endereços diferentes, mas esse endereçamento não tinha impacto nas redes exteriores. A atribuição de endereços únicos e globais era impraticável devido ao excesso de *overhead* dos pacotes IP e devido à complexidade inerente a essa atribuição. Por outras palavras, era consensualmente considerado impossível utilizar os mesmos métodos e protocolos associados à atribuição de endereços IP. Portanto, seria impossível a interligação directa das RSSF com a Internet.

A criação do 6LoWPAN e, posteriormente do protocolo de encaminhamento RPL, alterou este paradigma. A partir de ambos, as RSSF e as LR-WPAN no geral têm a possibilidade de fazer parte da IoT e construir uma rede capilar embutida nos ambientes inteligentes. É por isso importante mostrar como o 6LoWPAN e o RPL são dois protocolos que tornam possível o desenvolvimento da IoT.

5.2 Camada de Adaptação 6LoWPAN

A possível atribuição de endereços IP aos nós de uma rede de sensores sem fios é um passo fulcral na integração de redes capilares com a Internet. O 6LoWPAN é um protocolo que permite esta atribuição de endereços IPv6 em redes "*low-power and lossy channel*" [RFC4944]. A inclusão do IPv6 ao invés do IPv4 deve-se ao crescente número de dispositivos ligados à Internet. Com a adição de milhares ou milhões de nós sensores, a escolha torna-se óbvia.

A camada de adaptação foi incluída na pilha protocolar para que o protocolo 6LoWPAN pudesse fazer a ligação entre a subcamada MAC e a camada de rede. O 6LoWPAN utiliza pacotes IPv6 em redes IEEE 802.15.4. O comprimento máximo, *Maximum Transmission Unit* (MTU), dos pacotes IPv6 é de 1280 bytes, um comprimento demasiado elevado para uma só trama da norma IEEE 802.15.4 que especifica um comprimento máximo de 127 bytes. Retirando os 9 bytes de *overhead* da subcamada MAC, um pacote em IEEE 802.15.4 terá espaço para 118 bytes na camada de rede. O cabeçalho do IPv6 ocupa 40 bytes. Mais 8 bytes para o cabeçalho de um pacote UDP da camada de transporte, sobram somente 70 bytes para dados da camada de aplicação [IPSO09]. Esta limitação pode-se revelar bastante inoperante.

Mesmo que muitas aplicações não requeiram 70 bytes de espaço para dados num só pacote, não comprometendo o tamanho máximo da trama, a utilização do cabeçalho IPv6 revela-se demasiado ineficiente pois a aumento do consumo de energia de um nó está directamente associado, entre outras características, com o aumento do tamanho dos pacotes enviados,

bem com o número dos mesmos recebidos. Como tal, o 6LoWPAN mantém um mecanismo de compressão do cabeçalho que diminui o seu *overhead*. Em conformidade, para pacotes de dados com comprimentos elevados, o 6LoWPAN oferece um mecanismo simples de fragmentação.

5.2.1 Endereçamento 6LoWPAN

Como descrito no Capítulo 3, a norma IEEE 802.15.4 define dois tipos de endereçamentos distintos: o de 64-bit e o de 16-bit. O endereçamento IPv6 deriva do endereço físico e local EUI64 do dispositivo. Assim, o endereço 6LoWPAN deriva (tal como o IPv6) do endereço físico e local EUI64 do nó sensor. No entanto, a utilização do endereço 16-bit é igualmente possível em 6LoWPAN, diminuindo o *overhead* e, consequentemente, o consumo de energia.

O endereçamento é realizado da seguinte maneira para o *short address* de 16 bits: para manter a unicidade do nó, é adicionado um pseudo-endereço de 48 bits. Os primeiros 32 bits (32 bits mais significativos do endereço) são compostos pelos 16 bits do endereço do coordenador da rede e por 16 bits colocados a zero. Os 32 bits são concatenados com os 16 bits do endereço do próprio nó. Como resultado, o U/L bit do identificador da interface é colocado a zero, especificando que o endereço não é globalmente único pois o *short address* de 16 bits só pode ser utilizado localmente [HBE12].

5.2.2 Compressão 6LoWPAN

A compressão do cabeçalho IPv6 é realizada ao nível das camadas de rede e de transporte. Assumindo o protocolo de transporte para pacotes UDP, a compressão tem como objectivo reduzir o tamanho do cabeçalho IPv6 e do cabeçalho UDP. As Figuras 5.1 e 5.2 apresentam os cabeçalhos completos e originais de ambos os protocolos.

As compressões podem ser *Stateless* ou *Stateful* [HC08], [HC10]. No mecanismo *stateless*, ao longo do encaminhamento dos pacotes, a compressão não partilha qualquer estado ou contexto. Por sua vez, a compressão *stateful* é utilizada em vários tipos de rede para aumentar a eficiência da largura de banda, principalmente em redes móveis e em pacotes IPv6 cujo cabeçalho introduz aumentos consideráveis de *overhead*. Ambos operam obrigatoriamente sobre o mesmo protocolo e partilham o mesmo estado.

Existem várias técnicas de compressão de dados direccionadas para a compressão de cabeçalhos em sequências de pacotes. A compressão implica a existência de um compressor, do lado do emissor, e um descompressor do lado do receptor. A compressão de cabeçalhos pode ser realizada extremo-a-extremo mas é extremamente ineficiente quando os nós intermédios necessitam de visualizar os endereços IP completos para poderem tratar o pacote. Significa que o cabeçalho IPv6 tem de ser transmitido com os 128 bits relativos ao endereço.

A compressão salto-a-salto (*hop-by-hop*) é realizada apenas entre dois nós. Isto permite que vários esquemas e diferentes parâmetros sejam utilizados a cada salto que o pacote é transmitido. As decisões são locais entre os dois nós, possibilitando compressões bastante eficientes [SB09].

Outra técnica de compressão é a compressão baseada em fluxos de pacotes. Um conjunto de pacotes é tratado da mesma maneira em ambos os lados de uma comunicação. As características ou regras de compressão/descompressão podem ser estabelecidas durante o processamento do primeiro pacote de um conjunto de pacotes. O conjunto de pacotes por sua vez mantém as mesmas características (por exemplo, os endereços de origem e destino). Assim, os pacotes de dados são classificados como pertencentes ao mesmo fluxo de pacotes estabelecido e, portanto, pertencem ao mesmo estado de compressão. Por outras palavras, um único fluxo de pacotes de dados pertencentes ao mesmo conjunto tem associado um determinado contexto, e que é partilhado entre o nó origem e o nó destino [SB09].

Este tipo de compressão/descompressão pode tornar-se pouco robusto e escalável. Primeiro porque a perda de pacotes causa eventualmente a perda de sincronização referente ao contexto do fluxo e da compressão/descompressão. E segundo, em redes cujo tráfego é originado a partir de milhares de dispositivos que utilizam a mesma *gateway* para acederem à rede exterior, a gestão do próprio contexto e do fluxo torna-se extremamente complexa [WK05]. Em redes IEEE 802.15.4 o encaminhamento dos fluxos de dados varia significativamente, o que leva à necessidade de mudar o contexto da compressão para todos os encaminhamentos que se vão construindo e actualizando ao longo do tempo.

A primeira compressão 6LoWPAN normalizada só implementa compressão *stateless*. Sem necessidade dos nós partilharem qualquer estado, o algoritmo de compressão é bastante simples. O único contexto que os nós partilham é o de estarem de facto ligados à mesma rede 6LoWPAN e, portanto, a mesma informação é válida para todos os nós.

Os esquemas de compressão 6LoWPAN definem a compressão não só do cabeçalho IPv6 como também do cabeçalho do protocolo de transporte UDP. Assim, inicialmente o 6LoWPAN definia dois esquemas de compressão: HC1 para comprimir o cabeçalho IPv6 e o HC2, para comprimir o cabeçalho UDP [IETF6LP07], [HBE12].

Relativamente ao cabeçalho IPv6, o HC1 remove informação redundante tal como o campo *Versão IP* (a versão é sempre a 6) e o *Comprimento do Payload* (já incluído no encapsulamento da camada MAC). Os campos *Classe de Tráfego* e *Flow Label* são colocados a zero adicionando a *flag* "C" e colocando-a a 1. O *Limite de Saltos* é mantido. O campo *Next Header* é comprimido para apenas 2 bits.

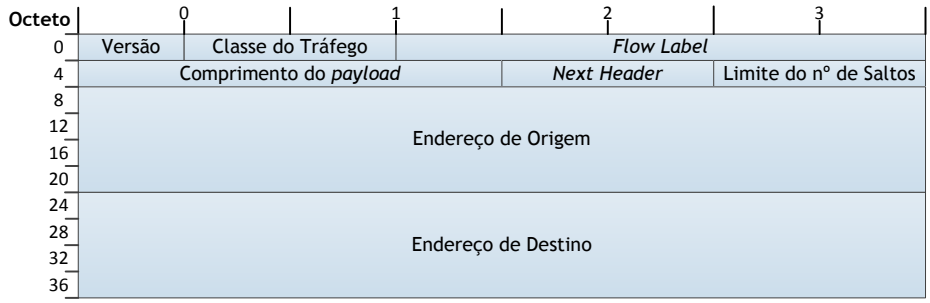

Figura 5.1 - Cabeçalho IPv6.

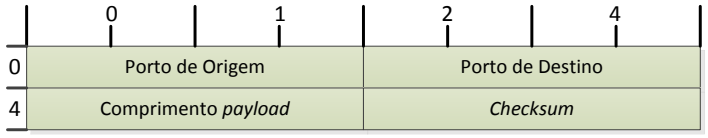

Figura 5.2 - Cabeçalho UDP.

Para comunicações locais, os endereços de origem e destino são redundantes, na medida em que são os endereços locais incluídos no encapsulamento MAC, e, portanto, são eliminados do cabeçalho. Na mesma medida, o prefixo global é omitido. No entanto, existem opções de compressão e não compressão para os endereços e os seus prefixos, como se pode observar na Tabela 5.1.

Tabela 5.1 - Opções de endereçamento.

	Prefixo	Endereço
00	Não comprimido	Não comprimido
01	Não comprimido	Derivado da camada MAC
10	Omitido	Não comprimido
11	Omitido	Derivado da camada MAC

Relativamente ao cabeçalho UDP, o HC2 comprime os campos referentes aos portos de origem e destino para 4 bits (*short port value*) se e só se os portos estiverem entre 61616 e 61631. O *Comprimento do Payload* é omitido assumindo que esta informação está presente no encapsulamento de camadas inferiores.

A compressão HC1+HC2 não é uma compressão robusta do ponto de vista global dos endereços IPv6. Na prática não é possível comprimir os cabeçalhos IPv6 sem qualquer estado. Por isso, a compressão HC1+HC2 foi substituída pela compressão IPHC, um esquema mais robusto, baseado em contexto [SB09], [HBE12], [IPSO09]. A compressão *stateful* IPHC tem como objectivo incluir a compressão de endereços IPv6 globais e *multicast* baseados na partilha de

estados dentro de um contexto. Isto é necessário para se realizarem comunicações IPv6 para fora (ou para dentro) da rede local. Um nó pode utilizar até 16 contextos diferentes.

O IPHC utiliza alguns campos opcionais do cabeçalho IPv6. Este campos são incluídos num cabeçalho 6LoWPAN especial denominado LoWPAN_NHC [SB09], [HBE12]. Estes cabeçalhos possuem informação opcional mas extremamente relevante, tanto dentro da rede local 6LoWPAN como de redes exteriores. Os cabeçalhos opcionais são sempre seguidos do cabeçalho IPHC e são apenas possíveis em encaminhamentos do tipo *route* (encaminhamentos ao nível da rede, isto é, ao longo de vários saltos IPv6). O campo *Next Header* (NH), que indica se existem ou não campos opcionais em utilização, é comprimido para apenas 1 bit e tem o valor 1 se existirem os cabeçalhos opcionais.

O IPHC utiliza apenas 13 bits, 5 dos quais compostos pelos bits menos significativos do campo *dispatch*. O IPHC partilha todas as características do HC1: os campos *Versão IPv6*, *Comprimento do Payload* e os endereços dos nós são omitidos. O prefixo dos endereços é igual ao prefixo utilizado localmente, previamente atribuído e conhecido por toda a rede local. Os campos *Classe de Tráfego* (CT) e *Flow Label* (FL) podem ou não ser incluídos. O campo *Limite de Saltos* (LS) mantém com um valor pré-conFigurado pelo coordenador.

O IPHC adiciona alguns campos que ocupam um total de 16 bits adicionais para dar suporte à compressão baseada em contexto. Os nós partilham estados representados nos prefixos dos endereços.

- ***Context* ID (CID)** – se a *flag* estiver a 1, existem extensões ao contexto;
- ***Source/Destination Address Compression* (SAC e DAC, respectivamente)** – se a *flag* estiver a 1, então a compressão é baseada num contexto. Se a *flag* estiver a 0, a compressão é sem estado e só é possível realizar comunicações locais;
- **Compressão *Multicast* (CM)** – se a *flag* estiver a 1, então o endereço de destino é um endereço *multicast*;
- **Modo de Endereçamento do nó de Origem e Destino** (MEO e MED, respectivamente) – os modos de endereçamento do nó origem e do nó destino são compostos por campos de 2 bits cada um. A Tabela 5.2 apresenta as combinações possíveis para o tipo de endereçamento utilizado no cabeçalho IPHC.

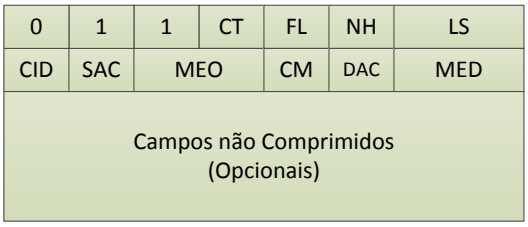

Figura 5.3 - Cabeçalho IPHC.

Tabela 5.2 - Opções de endereçamento IPHC.

SAC/DAC	MEO/MED	Descrição
0	00	Sem Compressão
	01	64 bits não comprimidos, sem prefixo
	10	16 bits não comprimidos (*short address*), sem prefixo
	11	Endereço totalmente omitido
1	00	Reservado
	01	Prefixo baseado num contexto e endereço 64 bits não omitidos
	10	Prefixo baseado num contexto e endereço 16 bits não omitidos
	11	Prefixo baseado num contexto e endereço omitido

Os endereços *multicast* são importantes na medida em que uma das características do IPv6 é o de não comunicar em difusão (*broadcast*), isto é, não possui um endereço *broadcast*. O IPv6 reune um vasto conjunto de endereços *multicast* para diferentes tipos de rede, equipamentos, ou funções. O prefixo utilizado para o endereço *multicast* em redes 6LoWPAN locais tem o formato FF02::A1. Pacotes com este endereço de destino em particular são recebidos por todos os nós.

Se a *flag* NH estiver a 1, o cabeçalho IPHC é seguido pelo cabeçalho LoWPAN_NHC (3 bits) e de uma nova *flag* NH. A indicação do seguimento de um cabeçalho UDP é uma excepção à regra adicionando duas *flags*, C e P, que informam se o *checksum* UDP é eliminado e/ou se os portos de origem e destino são comprimidos, respectivamente. É possível ter até oito campos opcionais para funções bastante específicas. Dois dos campos são reservados para propostas futuras. A Tabela 5.3 apresenta as combinações para a ocupação dos campos opcionais em IPHC.

Tabela 5.3 - Ocupação dos campos opcionais IPHC.

1110 000 NH	Cabeçalho *Hop-by-Hop* IPv6
1110 001 NH	Cabeçalho *Routing* IPv6
1110 010 NH	Cabeçalho *Fragment* IPv6
1110 011 NH	Cabeçalho *Destination Options* IPv6
1110 100 NH	Cabeçalho *Mobility* IPv6
1110 101 NH	Reservado
1110 110 NH	Reservado
1110 111 NH	Cabeçalho IPv6
11110 CP	Cabeçalho UDP

5.2.3 Encaminhamento *Mesh* versus *Route*

A camada 6LoWPAN permite facilmente a utilização dos endereços locais em encaminhamentos do tipo *mesh* através de saltos múltiplos dentro da mesma rede local 6LoWPAN. Portanto, o encaminhamento é realizado localmente. Se o tráfego tiver como destino um nó que se encontre fora da rede local, o encaminhamento terá de ser obrigatoriamente do tipo *route*, isto é, o encaminhamento é realizado puramente através da camada de rede utilizando tabelas de endereçamento IPv6 [HC08], [HC10], [OSR11].

Nos encaminhamentos do tipo *mesh,* os endereços de origem e de destino são alterados na trama de dados a cada salto, isto é, é alterado o cabeçalho MAC, mais especificamente os seus endereços locais. Este é o comportamento tradicional dentro de uma *Local Area Network* (LAN). O endereço do último nó destino é imutável e sempre conhecido ao longo do encaminhamento devido ao encapsulamento de um cabeçalho específico denominado *Mesh Addressing Header*, cuja descrição será avaliada a seguir. O endereço final é conhecido neste cabeçalho como *Endereço Mesh*. Este endereço permite que os fragmentos de uma única trama sejam encaminhados através de múltiplos caminhos possíveis, sem que seja necessário realizar o reagrupamento (*reassembly*) nos nós intermédios. Esta técnica é bastante semelhante à *Distribuição da Carga* (*Load Balance*), técnica que permite que várias tramas sejam enviadas para o mesmo nó destino através de vários caminhos, iguais em termos de custo.

No encaminhamento do tipo *route*, é realizado cada novo salto tendo em conta as tabelas de encaminhamento construídas pelos nós. Esta técnica de encaminhamento possibilita a entrega de pacotes fora da sua rede local, seja para redes 6LoWPAN, seja para outro tipo de redes externas como a Internet. O prefixo IPv6 é global e, por isso, necessariamente o mesmo para todos os nós na mesma rede. A função responsável por manter as tabelas de encaminhamento é separada da própria função que permite o encaminhamento por IP, ao verificar na tabela qual o melhor caminho para fazer chegar o pacote até ao destino final. Este tipo de encaminhamento é mais robusto e eficiente do que o encaminhamento *mesh*. No entanto, tem a desvantagem de fragmentar e desfragmentar os fragmentos de pacotes por cada salto que é feito. A diferença entre ambos os dois tipos de encaminhamento é ilustrada na Figura 5.4. O traço a vermelho representa o encapsulamento/desencapsulamento do tipo *route*, enquanto que o traço verde representa o do tipo *mesh*.

Existem vários protocolos de rede propostos para este tipo de encaminhamento e que garantem que as restrições 6LoWPAN não sejam comprometidas. A construção da tabela de encaminhamento é efectuada através da reutilização de mensagens de descoberta de vizinhança já definidas pelo IPv6 (nomeadamente o ICMPv6 *Neighbour Discovery* (ND)).

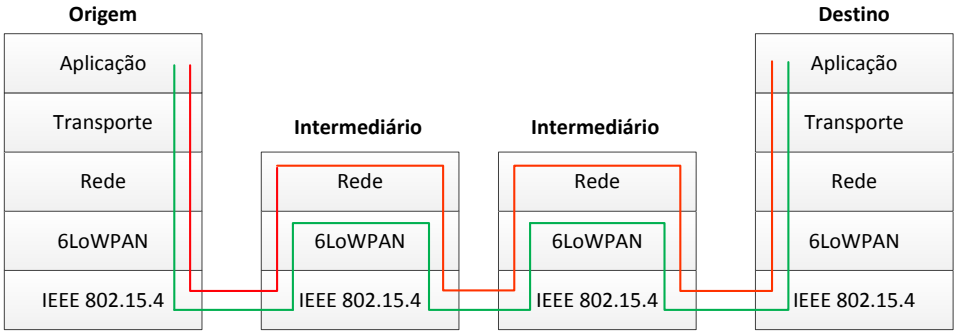

Figura 5.4 - *Mesh* versus *Route*.

Através das mensagens ICMPv6, a topologia ao nível da camada de rede é construída para uma determinada sub-rede, e partilhada pelos nós que a compõem. A topologia de encaminhamento será concluída mais eficientemente se os protocolos de rede possibilitarem a partilha de parâmetros e métricas entre nós que indiquem quais os melhores caminhos para realizar o encaminhamento dos dados como, por exemplo, a qualidade das ligações ou o nível de recursos disponíveis nos nós [OSR11]. Para tal, o IETF propõe um protocolo de rede directamente direcionado para redes do tipo 6LoWPAN, descrito na Secção seguinte.

5.2.4 Encapsulamentos 6LoWPAN

O 6LoWPAN permite o encapsulamento de cabeçalhos específicos consoante o tipo de ligação pretendida [HBE12], [HC08]:

- *Mesh addressing header;*
- Cabeçalho *Hop-by-Hop* (opcional);
- Cabeçalho de fragmentação (opcional);
- Cabeçalho *Compressed/Uncompressed*.

Todos estes cabeçalhos incluídos na camada de adaptação são identificados com subcabeçalhos, denominados *dispatch*. Cada *dispatch* é composto por 1 byte que identifica a natureza do cabeçalho. A identificação concreta do cabeçalho associado a um determinado *dispatch* é realizada por um padrão de 6 bits (8 bits no total) e, portanto, são possíveis 64 identificações, sendo a maioria reservada para propostas futuras. A Tabela 5.4 apresenta as sequências possíveis do *dispatch*.

Uma opção do *dispatch* muito particular é o campo 6LoWPAN Not-A-LoWPAN (NALP) que permite que haja coexistência com protocolos e tipos de ligações/redes diferentes. Todos os nós dentro de uma rede 6LoWPAN deverão descartar todas as tramas que contenham este cabeçalho. Todas as restantes 57 identificações são reservadas.

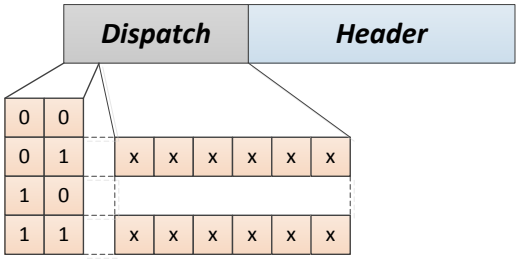

Figura 5.5 - *Dispatch Byte*.

Tabela 5.4 - Descrição da sequência *Dispatch Byte*.

Bits do *Dispatch*	Descrição
00	Informa que não é um pacote 6LoWPAN (NALP)
01 000001	Ipv6 não comprimido
01 000010	Informa a presença de compressão HC1/IPHC
01 001000	Pacote *broadcast*
10	*Mesh Cabeçalho*
11 000xxx	Primeiro *Cabeçalho* de Fragmentação
11 100xxx	Sequência de Fragmentações

Existem quatro tipos de tramas principais em redes 6LoWPAN. Todas estas tramas têm encapsulados os cabeçalhos IPHC e os cabeçalhos MAC e PHY da norma IEEE 802.15.4. O cabeçalho UDP da camada de transporte é opcional em todos os tipos de tramas. A Figura 5.6 apresenta a trama mais simples cujo destino se situa a uma distância de apenas um salto e que não necessita, portanto, de ser fragmentada.

Figura 5.6 - Trama base.

Figura 5.7 - Trama com implementação de encaminhamento *mesh*.

A trama da Figura 5.7 é formada pelo *Mesh Addressing Header*, que possibilita encaminhamento multi-salto local (*mesh*) reservando um espaço na trama para identificar o endereço 802.15.4 do nó inicial e do nó final. Desta maneira, a cada novo salto durante o encaminhamento do pacote, mesmo que os endereços locais indicados na trama sejam

substituídos no novo encapsulamento realizado por um nó intermédio, os endereços de origem e destino iniciais são mantidos neste campo. A trama possui um contador *time-to-live* que é decrementado cada vez que a trama percorre um salto, evitando que entre em caminho fechado (*loop*). Este campo tem 4 bits, limitando o número de saltos para um valor máximo de 14.

O terceiro e quarto tipo de encapsulamento (Figura 5.8 e Figura 5.9, respectivamente), independentemente da adição do cabeçalho *mesh*, adiciona o cabeçalho de fragmentação.

Figura 5.8 - Trama com fragmentação.

Figura 5.9 - Trama com fragmentação e encaminhamento *mesh*.

A fragmentação possibilita codificar os pacotes em múltiplas tramas se o *payload* somado de todas as camadas for maior do que 127 bytes. O mecanismo é semelhante ao utilizado na camada de transporte do modelo OSI (referido como segmentação), que parte o PDU em segmentos mais pequenos e realiza o reagrupamento no lado do receptor, para manter a ordem correcta do PDU original. No 6LoWPAN, quando o pacote é partido, o primeiro segmento especifica o tamanho total do pacote num campo denominado *datagram size*. Após essa especificação, todos os segmentos têm um campo *datagram tag* em comum que identifica um pacote IP particular. Este campo é utilizado pelo receptor, em conjunto com os endereços MAC do emissor e do próprio receptor, de maneira a conseguir identificar os segmentos que compõe um dado pacote.

5.3 IETF ROLL – Camada de Rede RPL

O *Routing Protocol for Low-power and lossy networks* (RPL) foi criado em 2011 pelo IETF ROLL para providenciar a capacidade de se construírem topologias de rede eficientes e executar os encaminhamentos 6LoWPAN em redes de potência reduzida [RFC6550]. Foi elaborado um conjunto de documentos *Request for Comments* (RFC) que reúne e destila uma década de investigação sobre protocolos e restrições em RSSF, descrevendo formas de implementação em espaços urbanos, edifícios, casas e na indústria. A avaliação realizada pelo ROLL dos inúmeros protocolos bem conhecidos (*well-known*) na literatura levou à criação do RPL, um protocolo semelhante ao protocolo *Collection Tree Protocol* (CTP) [GFJML09].

O RPL visa dar suporte aos encaminhamentos do tipo *route* (por IP) construindo uma topologia de rede entre os nós de uma rede. As restrições de memória, energia e qualidade das ligações

são tidas em linha de conta. Portanto, a informação relacionada com a construção da topologia e dos encaminhamentos não pode ser abundantemente partilhada. A tabela de encaminhamento não deverá ser muito extensa, bem como o número de contextos (prefixos) de rede partilhados entre os nós [HC10].

Todas as ligações deverão ser bidirecionais, embora o RPL considere várias características, como a possível assimetria das ligações. Utiliza uma topologia em árvore construída a partir de nós "raízes" (*roots*) com características semelhantes ao dos *cluster heads* de protocolos de rede profundamente estudados e conhecidos na literatura, como por exemplo o LEACH [HCB00]. O protocolo suporta três tipos de tráfego:

- Tráfego da *gateway* para múltiplos nós em comunicações *point-to-multipoint* (P2MP);
- Tráfego de muitos nós para a *gateway* em comunicações *multipoint-to-point* (MP2P);
- Comunicação *point-to-point* (P2P).

A topologia em árvore é baseada no conceito de *Direct Acyclic Graph* (DAG), conceito proveniente da teoria dos grafos que estuda as relações que podem existir entre dois objectos do mesmo conjunto. Em RPL, os objectos são os nós sensores de uma rede cujas relações, entre eles, são as suas próprias ligações. Num DAG, os objectos não têm ligações/relações com eles próprios (nenhum caminho se inicia e termina no mesmo objecto) mas mantêm sempre ligações com os objectos vizinhos. Apenas o nó raiz do grafo pode ter ligações com ele próprio. Obrigatoriamente, estes grafos têm pelo menos um nó fonte e um nó raiz.

Uma rede baseada em DAG agrega um conjunto de sub-redes por cada nó raiz (nó coordenador) designadas *Destination Oriented DAG* (DODAG). Vários nós raíz numa só DAG podem estar ligados por uma ligação *backbone* comum, possibilitando a interligação de DODAGs. A topologia da DODAG é mantida tendo como base quatro variáveis distintas:

- **DODAGID** - identifica uma DODAG em particular;
- **DODAGVersionNumber** - identifica qual a versão da DODAG após a sua formação e possíveis modificações da topologia;
- **RPLInstanteID** - identifica um conjunto de DODAGs, podendo ser a partir do identificador DODAGID e/ou pela versão DODAGVersionNumber (um conjunto de DODAGs numa rede é uma Instância RPL);
- **RANK** – variável cujo valor define as posições dos nós dentro das DODAGs, calculada através de uma função objectivo.

As DODAGs são construídas segundo uma função objectivo que define o *rank* para cada nó. A partir dos resultados obtidos com a função objectivo, os nós escolhem e optimizam os caminhos utilizados no encaminhamento de pacotes de dados ao associarem-se a nós vizinhos. A função objectivo é calculada a partir de métricas de custo (qualidade das ligações, energia

residual dos nós, etc). Consoante a aplicação que se pretenda desenvolver, diferentes equilíbrios (*tradeoffs*) podem ser sugeridos.

Semelhante ao CTP, o RPL assenta sobre os princípios da agregação e coleção de dados, sendo as ligações M2P o tipo de tráfego mais comum [KTHHCHL11]. Um nó poderá ser o intermédio de vários outros nós, dos quais recebe os seus pacotes de dados e reenvia para nós mais próximos do nó raiz (ou até para o próprio nó raiz, caso tenha ligação com ele).

O RPL funciona sobre um plano de controlo e um plano de dados, ambos presentes para melhorar o desempenho da topologia de rede, desde a manutenção e reparação da mesma. O plano de controlo é composto por um número de pacotes de controlo que, eficientemente trocados entre nós num esquema de *Beaconing Adaptativo*, prova responder aos desafios de manutenção da topologia e dos encaminhamentos dos dados. O plano de dados concretiza essencialmente uma estratégia de *Validação dos Caminhos de Dados* (*Datapath Validation*). Inconsistências e falhas nos encaminhamentos podem ser detectados através dos próprios pacotes de dados se o *rank* do emissor for agregado ao pacote de dados [GFJML09].

5.3.1 Construção e manutenção da topologia

A função objectivo e o *rank* incluem-se nas mensagens ICMPv6 de *broadcast*, *DODAG Information Object* (DIO). Estas mensagens são exclusivamente de controlo, trocadas entre os nós para formar, manter e actualizar a topologia da DODAG.

Como apresentado na Figura 5.10, a construção da rede é iniciada pelo coordenador/*nó raiz*/*gateway*, quando emite o primeiro pacote *DODAG Information Object*. Todos os seus nós vizinhos que receberem o DIO adicionarão o coordenador a uma tabela armazenada em memória, relacionada com a tabela de encaminhamento, que armazena essencialmente o endereço dos nós vizinhos com os quais existem futuras e possíveis ligações de canal ascendente (*uplink*) e canal descendente (*downlink*).

A Figura 5.11 ilustra o processo normal da construção inicial da rede. Os nós participantes na rede emitem o seu DIO, avisando os vizinhos do seu *rank*, permitindo melhorar os encaminhamentos e a participação de nós mais afastados do nó raiz.

Nesta tabela inclue-se os *ranks* dos vizinhos e os seus respectivos endereços. O valor do *rank* do emissor é obtido a partir do DIO e, após o seu processamento, os nós recetores calculam o seu *rank* com base no *rank* do emissor. Quando o *rank* calculado é maior que o *rank* do emissor do DIO, o receptor aceita o emissor como sendo seu nó pai (ou nó progenitor) para encaminhamentos ascendentes, criando através dele um caminho até ao nó raíz. Caso contrário, rejeita o emissor como sendo seu nó pai.

O *rank* do coordenador é uma constante maior do que zero e tem de ter o valor mais reduzido entre todos os *ranks* calculados na rede pelos nós.

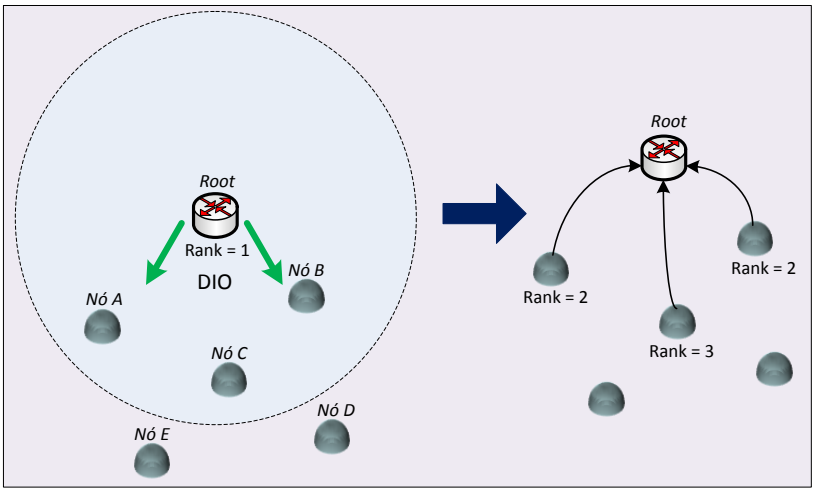

Figura 5.10 - Construção inicial da rede.

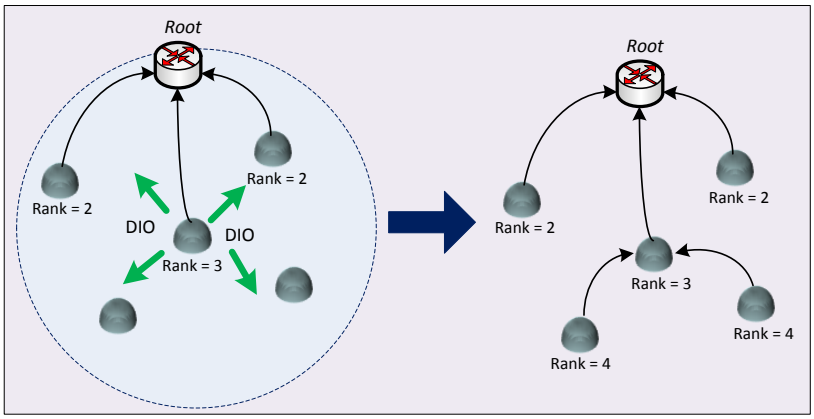

Figura 5.11 - Descoberta de vizinhos próximos.

A mensagem DIO é propagada por um nó após este se juntar a uma DODAG com a respectiva Instância RPL. Nós mais distantes que recebam os DIOs destes nós emissores podem juntar-se à mesma DODAG, ou a outra caso recebam DIOs com identificações diferentes e com uma Instância RPL direccionada para as suas funções.

O pacote DIO possui os quatro campos essenciais para identificar uma DODAG além de outros campos que auxiliam na formação de encaminhamentos entre os nós e na decisão de participar ou não numa DODAG. A Figura 5.12 ilustra o cabeçalho DIO.

Ao longo e após a formação da DODAG, os pacotes DIO continuam a ser emitidos com alguma frequência pelos nós. O objectivo é actualizar e reparar os encaminhamentos quando detectada alguma falha ou inconsistência, mas não só. Mesmo em cenários onde após formada a topologia não ocorram quaisquer erros, os valores das métricas de custo dos *ranks* sofrem alterações ao longo do tempo. Eventualmente podem ser sempre descobertos e optimizados melhores encaminhamentos, aumentando o rendimento binário ou balanceando o tempo de vida da rede.

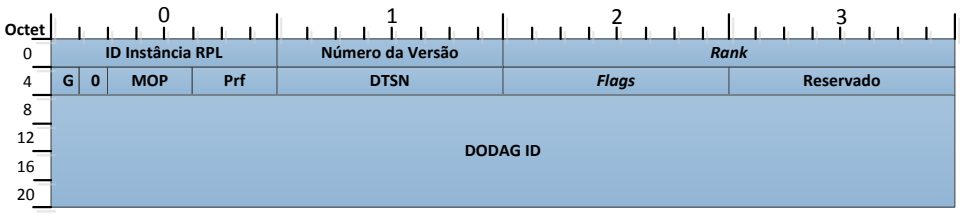

Figura 5.12 - Objecto DIO.

Uma rede de sensores sem fios RPL é modelada como um grafo $G = (N, L)$, onde *N* é o conjunto dos *n* nós (excluindo o nó raiz) presentes na rede e *L* é o conjunto de ligações que interliga esses nós. Para cada nó $n_i \in N_S\ (i = 1, 2, \dots, n)$ existe um conjunto *A* de nós candidatos cuja eleição para se tornarem o nó pai de outro nó é realizada consoante o *rank* de cada um dos nós. O conjunto dos nós duma rede e o conjunto de nós progenitores de um qualquer outro nó são representados pelos conjuntos das Expressões 5.1 e 5.2, respectivamente:

$$N_S = \{n_1, n_2, \dots, n_n\} \tag{5.1}$$

$$A(n_i) = \{a_1, a_2, \dots, a_m\} \tag{5.2}$$

Sendo *A* um subconjunto do número total de nós, tem-se $A(n_i) \subset R_S$. O nó raíz representado como n_S, é o único nó da rede que não possui lista de candidatos e portanto, o seu subconjunto é vazio:

$$A(n_S) = \{\emptyset\} \tag{5.3}$$

Para cada nó n_i existe um conjunto de *m* ligações *l*, igual ao número de nós candidatos (Expressão 5.4). No conjunto *L* existe uma ligação $l \in L(n_i)$. Assim, tem-se que $\exists a_j \in A(n_i)$, onde a_j é o nó candidato selecionado para ser o nó pai de n_i.

$$L(n_i) = \{l_1, l_2, \dots, l_m\} \tag{5.4}$$

Cada ligação *l* tem indirectamente associado um *rank* igual ao *rank* calculado no nó n_i, em relação a um nó pai seleccionado $a_j\ (j = 1, 2, \dots, m)$:

$$l(n_i, a_j) = RANK \tag{5.5}$$

Para um nó candidato a_p ser selecionado como nó pai, a Expressão 5.6 nunca deve ser violada:

$$l(n_i, a_p) < \forall\, l(n_i, a_j) \tag{5.6}$$

Concluindo, a construção da rede para cada nó n_i pode ser definida da seguinte forma:

$$C = \{\, \forall n_i \in R_S, \quad \exists a_p \in A(n_i) : l(n_i, a_p) < \forall\, l(n_i, a_j) \} \tag{5.7}$$

Quando um nó n_x recebe um pacote DIO de um nó n_y verifica a instância, a versão e a identificação da DODAG. A partir destes valores podem ser classificados vários casos diferentes [WTZA10]. Se o nó participar na mesma DODAG, verifica se o nó emissor está presente na sua lista de endereços:

- Se não estiver corresponde ao primeiro caso;
- Se estiver presente mas não for o seu pai corresponde ao segundo caso;
- Se estiver e for o seu nó pai actual corresponde ao terceiro caso.

Em todos os três casos, o nó receptor processa um *rank* temporário R_T com base no *rank* R_y do nó emissor, e compara-o com o seu *rank* actual, R_x, de maneira a tomar decisões quanto à sua própria posição dentro da rede.

- **1º Caso - *Emissor não presente na lista***

  - O nó n_x adiciona o nó n_y à sua lista de vizinhos;

    - Se $R_T > R_x$ então o nó n_y encontra-se em pior posição em relação ao pai do nó n_x. Com base no quociente (R_T/R_x), o nó n_x verifica se consegue optimizar o *rank* de n_y;
    - Se $\left(\frac{R_T}{R_x} > R_{Threshold}\right)$ [WTZA10], então o nó n_x responde ao nó n_y com um DIO *unicast*, onde inclui o valor de R_x com o objectivo de optimar a posição de n_y. Caso contrário, ignora o DIO;

    - Se $R_T < R_x$ então o nó n_y encontra-se em melhor posição comparativamente ao pai do nó n_x. O nó n_x selecciona novo nó pai nomeando n_x e emitindo um DIO com o novo *rank* calculado (R_T);

    - Se $R_T = R_x$ o nó n_x ignora o DIO e descarta-o. Garante-se assim a não formação de possíveis caminhos fechados.

- **2º Caso – *Emissor presente na lista mas não como nó pai***

  - Se $R_T > R_x$ ou $R_T < R_x$ então realizam-se os mesmos procedimentos do primeiro caso para ambas as situações;

- **3º Caso – *Emissor como nó pai***

  - Se $R_T > R_x$, então o nó n_x procura na lista um nó vizinho com melhor *rank*, selecionando um novo ou permanecendo com o mesmo consoante os resultados. Em ambas as situações emite-se um novo pacote DIO com o novo *rank*;

  - Se $R_T < R_x$, então o nó permanece com o mesmo nó pai e emite um pacote DIO com o novo *rank*.

As Figuras 5.13 e 5.14 apresentam exemplos de partilha de DIOs que resultam na actualização das ligações e melhoria dos encaminhamentos desde os nós mais profundos até ao nó raiz. A Figura 5.13 corresponde ao caso de um nó mais profundo na rede emitir um DIO, cujo valor do *rank* calculado por um nó receptor menos profundo emite uma resposta com o objectivo de melhorar a sua posição na rede.

A Figura 5.14 ilustra outro exemplo de actualização da posição dos nós e dos encaminhamentos criados. Quando um nó mais profundo recebe um DIO cujo cálculo do *rank* temporário é menor que o *rank* actualmente considerado, é realizada uma nova selecção e é emitido um novo DIO, alertando a nova posição (novo *rank*) com o novo encaminhamento consequentemente criado.

A Figura 5.15 apresenta o diagrama de estados para os diferentes casos possíveis, e os passos necessários a realizar para manter a rede consistente e actualizada quando ocorre a recepção de um DIO em qualquer nó (excluindo o nó raiz).

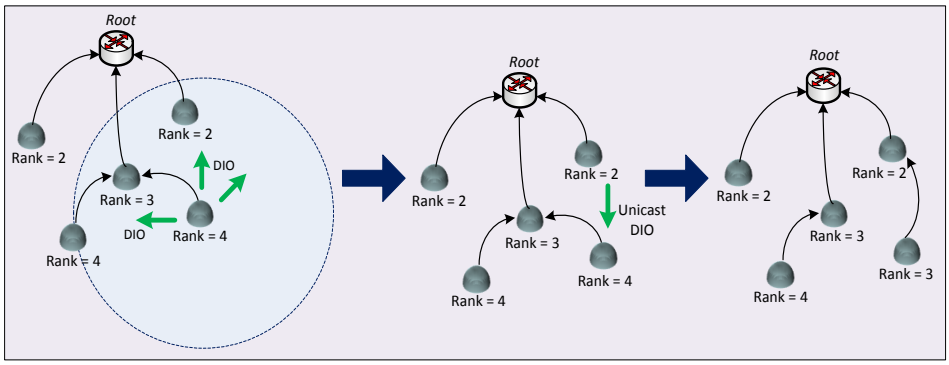

Figura 5.13 - Partilha de DIOs com exemplo de actualização para melhorar encaminhamento (1).

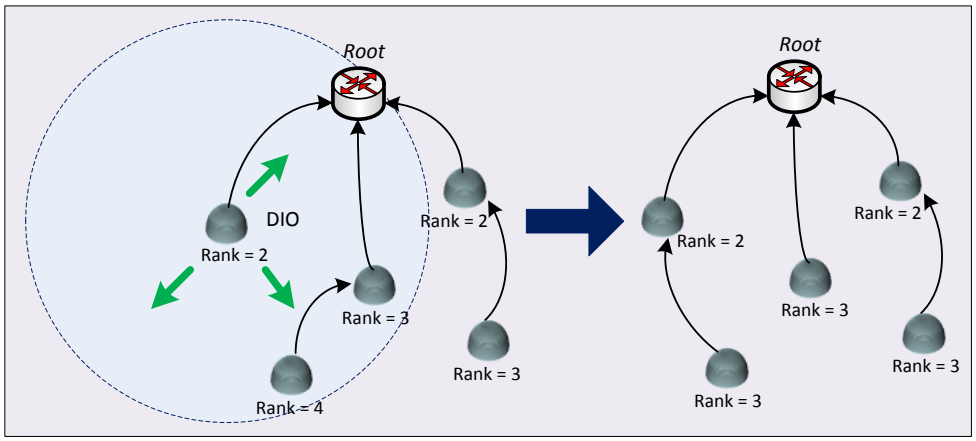

Figura 5.14 - Partilha de DIOs com exemplo de actualização para melhorar encaminhamento (2).

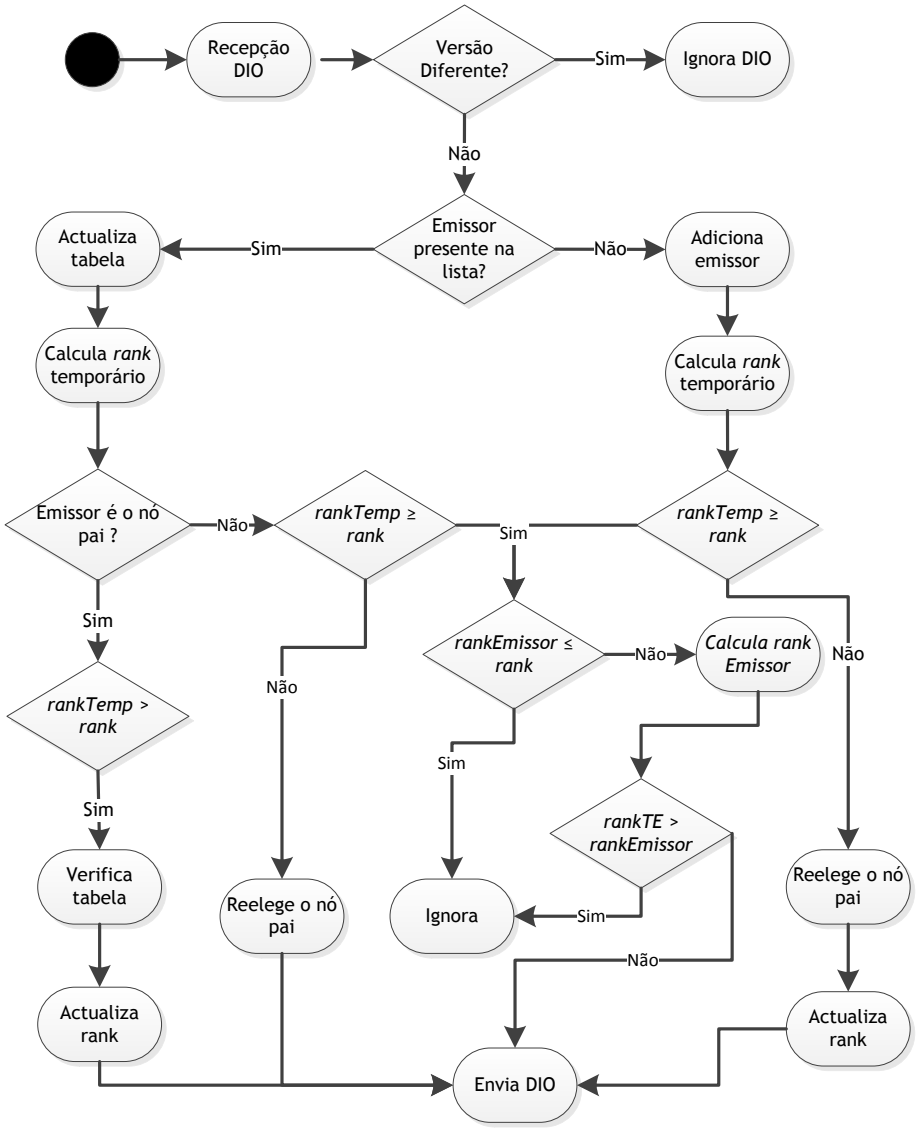

Figura 5.15 - Diagrama de estados para os diferentes casos na recepção de um DIO.

5.3.2 Algoritmo *Trickle*

O tempo de espera até um novo pacote DIO ser emitido é algo difícil de determinar. Além de depender fortemente do tipo da aplicação, também é fundamental considerar os diferentes e possíveis equilíbrios que se pretendam manter. Para aplicações onde ocorrem constantemente profundas alterações em toda (ou parte) da rede, o tempo de espera até enviar um novo DIO deverá ser curto o suficiente para manter as versões actualizadas por todos os nós participantes. Por exemplo, cenários com forte mobilidade. Aplicações para missões críticas não deverão ter períodos elevados de espera, sendo importante assegurar que os encaminhamentos se mantém actualizados e sem quebras de comunicação. No entanto, cenários onde os nós estão continuamente fixos e o tempo de vida da rede é uma das características mais importantes a optimizar, este intervalo de espera pode ser alongado e diminuído o *overhead*.

O RPL utiliza o algoritmo *Trickle* [IETFTri11], [LPCS05], para gerir este intervalo de tempo entre as emissões de DIOs. Este algoritmo foi criado para propagar e manter actualizados código e metadados em RSSF. A propagação dos DIOs durante a descoberta da rede deve ser rápida e alcançar todos os nós numa vizinhança que desejem participar na DODAG. Após a estabilização da rede, a troca de DIOs é aliviada para um intervalo de tempo suficientemente longo para manter mínima a manutenção da rede. No entanto, sempre que os nós detectem inconsistências e falhas, o intervalo de tempo de emissão do DIO deve ser reduzido para permitir que se actualize imediatamente a topologia.

Resumidamente, o *Trickle* estabelece intervalos de tempo aleatórios em cada nó no intervalo $(0 < t \leq \tau)$. A variável τ tem um valor mínimo τ_l e um valor máximo τ_h. Inicialmente utiliza o valor mínimo, o que possibilita rápida propagação de DIOs.

Enquanto o nó receber DIOs com informação diferente da sua, actualizada ou desactualizada, irá permanece no tempo mínimo. Quando receber informação igual à sua e não ocorrerem alterações do seu estado, o nó passa para o tempo máximo, alongado o intervalo τ. O algoritmo participa ainda com um contador *C* e uma constante *k*. O contador é incrementado sempre que um DIO recebido possuir informação idêntica ao do nó receptor. Quando o nó alcançar o instante *t*, onde *C* < *k*, então o nó emite o seu DIO. Sempre que o intervalo de tempo termina, o contador é reinicializado e um novo instante de tempo aleatório *t* é escolhido. Caso o contador ultrapasse o valor constante *k*, o temporizador *t* é forçado a reinicializar e um novo intervalo de tempo é escolhido.

Para protocolos MAC com princípios de sincronização entre os nós, o algoritmo *Trickle* é bastante escalável e eficiente. No entanto, para redes *non-beacon* os nós poderão perder, no pior dos cenários, vários pacotes DIO se transitarem do estado SLEEP para o estado activo em tempos diferentes dos tempos de difusão de DIOs por parte de nós emissores [LPCS05]. Se os nós tiverem um *duty cycle* curto de permanência no estado activo, o suficiente apenas para

escutar pacotes, terão obrigatoriamente de aumentar os tempos de verificação do canal. No entanto, se o *duty cycle* for verdadeiramente curto, e dada a natureza do *Trickle*, os nós poderão continuar com uma taxa de recepção de pacotes DIO extremamente reduzida. Existirão sempre emissões redundantes de DIOs nestes cenários.

O algoritmo *Trickle* apresenta uma solução oposta a estes cenários, pois estabelece um período de escuta no alcance de $(\tau/2 < t \leq \tau)$. Forçando esta restrição, os nós garantem um intervalo de tempo suficiente para escutar os seus vizinhos. Quando um DIO é recebido, todos os nós receptores transitam para o estado SLEEP com uma duração mínima igual ao seu tempo de escuta.

5.3.3 Solicitação de Informação

As mensagens ICMP *DODAG Information Solicitation* (DIS) são utilizadas por novos nós adicionados à rede para solicitar informação *broadcast* DIO aos vizinhos [RFC6550]. Portanto, todos os nós duma vizinhança que recebam um DIS responderão com um DIO para que o novo nó possa tomar uma decisão quanto à sua posição na DODAG. É o substituto das mensagens de controlo *Router Solicitation* (RS) do IPv6 ND para a versão RPL, cujo objectivo é actualizar rapidamente a rede com o aparecimento de novos nós. O DIS é um substituto essencial do *Neighbor Unreachability Detection* (NUD) se for expandido para detectar precocemente falhas de encaminhamento. O DIS pode ser programado para ser emitido periodicamente com o objectivo de detectar falhas de ligação com um nó pai.

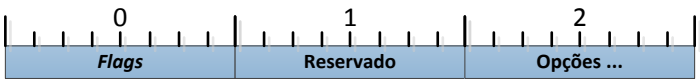

Figura 5.16 - Objecto DIS.

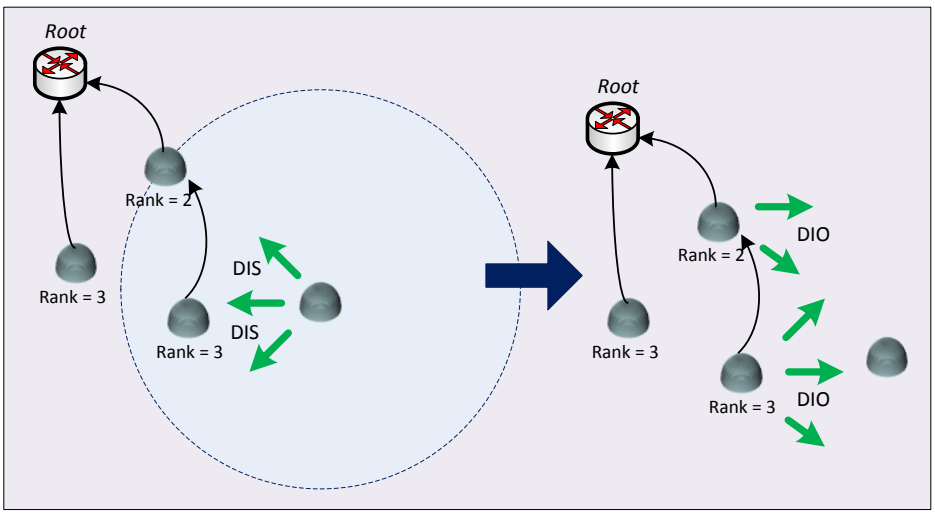

Figura 5.17 - Descoberta/solicitação com pacotes DIS.

O contador DIS, apesar de baseado no *Trickle*, não é igual ao contador DIO. Os seus intervalos de tempo deverão ser mais reduzidos inicialmente para que todos os nós da sua vizinhança recebam e respondam com um DIO ($\tau_l = 1s$). Após a recepção de pelo menos um DIO e um limite mínimo de DISs emitidos, garantindo a sua recepção por toda a vizinhança, o intervalo de tempo do DIS pode ser fortemente aumentado, tendo apenas o nó pai como destino.

Nós que ainda não estejam associados à rede e recebam DIS de outros nós com o mesmo objectivo, ignoram até encontrarem um nó pai e calcularem o próprio *rank*. Se ao fim de um limite máximo de pacotes DIS emitidos os nós não obtiverem resposta, passam para o estado SLEEP até à próxima vez que se encontrarem no estado activo. A Figura 5.17 apresenta a utilização básica e primária do DIS.

É apresentado na Figura 5.16 o objecto base DIS num pacote ICMPv6. Os dois primeiros bytes (campos *flags* e *reservado*) não são normalmente utilizados, inicializados com 0 pelo emissor e ignorado pelo receptor. Nas opções pode-se adicionar um ou vários *paddings* e/ou solicitar outro tipo de informação (com o hexadecimal 0x07).

5.3.4 Detecção e Recuperação de Falhas

Uma falha numa ligação pode ocorrer em qualquer instante. Se ocorrer no intervalo de tempo entre a emissão de um DIO e a emissão de dados para o mesmo nó pai, ambos não serão entregues. O emissor reenviará o pacote de dados na falta da resposta ACK. Se a ligação permanecer inactiva, o nó falhará até o mecanismo MAC actuar. Em vez de reenviar várias vezes o mesmo pacote de dados, o emissor pode reinicializar o seu estado, mantendo a tabela de encaminhamento com os endereços dos seus vizinhos e emitir ICMPv6 DIS *broadcast*. Após uma nova ligação ser estabelecida, os dados podem ser encaminhados novamente. A Figura 5.18 apresenta esta situação.

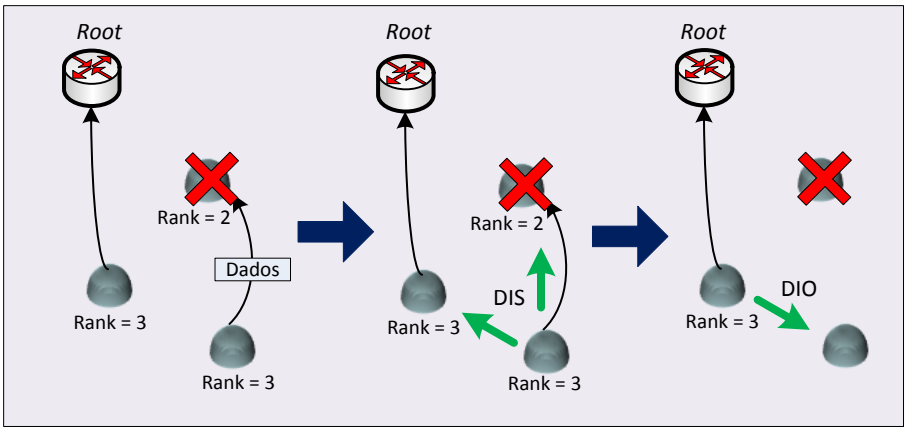

Figura 5.18 - Procura e descoberta dum novo encaminhamento após detectada uma falha.

Uma ligação perdida pode ser detectada antecipadamente se o nó pai permanecer durante um tempo máximo predefinido sem emitir pacotes DIO. Após esse tempo, o nó associado emitirá um DIS *unicast* para o nó pai e esperará pela resposta. Se a ligação estiver de facto indisponível, o nó não receberá a resposta do seu nó pai actual, reinicializando o seu estado e emitindo um DIS *broadcast* para todos os seus vizinhos. Eventualmente o nó estabelecerá nova ligação com outro nó pai. A situação é apresentada na Figura 5.19.

Nestes casos, é essencial definir uma estratégia coerente na difusão de pacotes DIS e posterior reeleição de um nó pai, eventualmente precoce se existirem nós candidatos com melhores condições para criar o encaminhamento. Assumindo que após a reinicialização do estado o nó mantém a sua tabela de encaminhamento mais recente com os endereços e *ranks* associados dos nós vizinhos, a estratégia sugerida é a seguinte, onde se assumem as seguintes variáveis:

$\boldsymbol{x}$ — Número de pacotes DIO recebidos com *rank* inferior ao *rank* do primeiro DIO recebido;

$\boldsymbol{y}$ — Número de pacotes DIS emitidos após a recepção do primeiro DIO;

$\boldsymbol{n}$ — Número de vizinhos na tabela com um *rank* inferior ao *rank* recebido no primeiro DIO.

A ligação a um nó pai só será realizada se a Expressão 5.8 for satisfeita:

$$\frac{x * y}{n} \geq 1 \tag{5.8}$$

Esta solução garante, com elevada probabilidade, que todos os nós (especialmente os que têm *ranks* mais reduzidos) recebam o DIS e emitam o mais rapidamente possível um DIO.

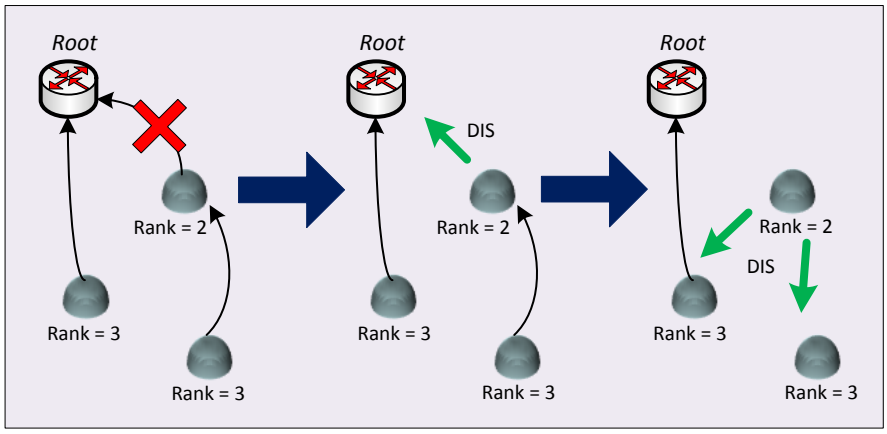

Figura 5.19 – Detecção antecipada de uma falha de ligação.

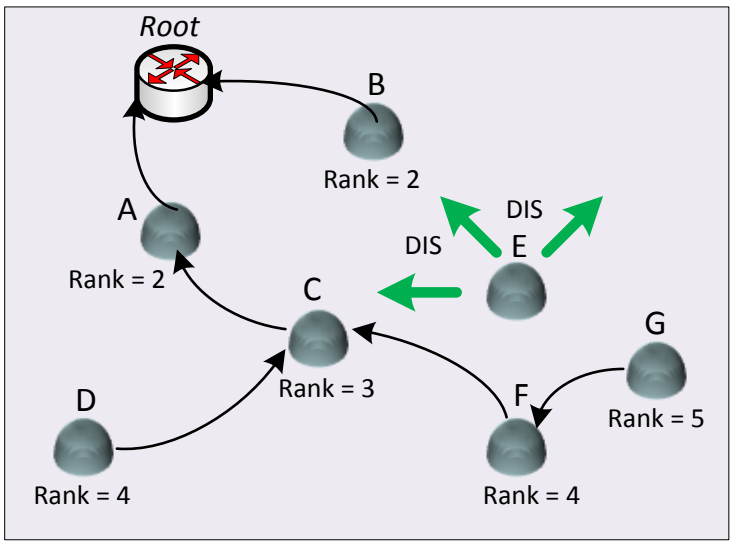

Figura 5.20 – Procura de novo nó pai após perda de ligação.

Considere-se um breve exemplo que se apresenta na Figura 5.20. Se o nó E perder a ligação com o seu nó pai, difundirá o pacote DIS por toda a rede, ao qual, os seus nós vizinhos responderão com DIOs. Como participante activo da rede antes de perder a sua ligação, o nó E possui uma tabela de encaminhamento (tabela que, na pior das hipóteses, estará completamente desactualizada, e na melhor das hipóteses estará totalmente actualizada). O mecanismo simplesmente considera ambas as situações, levando à emissão do DIS e à não reeleição cega do primeiro nó a responder com um DIO.

Tabela 5.5 - Tabela de encaminhamento do nó E da Figura 5.20.

Endereço do Nó	*Rank*
A	2
B	2
C	3
F	4
G	5

Se o primeiro DIO recebido for do nó F, independentemente do *rank*, o nó E verificará na sua tabela o número de nós que possuem *ranks* inferiores ao nó F, cujo valor é 3 nós vizinhos e cuja condição leva ao resultado $\frac{1*1}{3} = 0.333$. Como o menor intervalo de emissão dos DIOs é mais reduzido que o menor intervalo de emissão dos DISs, há garantia de que todos os nós que recebam o DIS emitirão o seu DIO para E. Se o nó G responder com um *rank* igual a 5, o nó E rejeitará o seu DIO. Eventualmente, os outros nós com melhores *ranks* emitirão os seus DIOs, levando à eleição de um nó pai favorito quando $\frac{3*1}{3} = 1$. Se por qualquer razão, nenhum outro

DIO for recebido a tempo ao fim de, por exemplo, $y = 3$ pacotes DIS emitidos $\left(\frac{1*3}{3} = 1\right)$, o nó E elegerá o nó F, mesmo existindo nós com melhores condições para o encaminhamento. Quando a condição for concluída e a nova ligação estiver estabelecida, o nó E encontrará eventualmente melhores encaminhamentos com a partilha normal e sucessiva de DIOs.

As falhas podem resultar em inconsistências na topologia de rede. Propagações das actualizações mais demoradas por toda a DODAG, após detectadas falhas nos encaminhamentos, resultarão nestas incoerências sem precedentes. No pior dos cenários, poderão ser criados problemas de caminhos fechados indesejados num ou vários ramos da DODAG.

Considere-se a Figura 5.21. As inconsistências podem ser rapidamente detectadas e resolvidas após a emissão de um DIO (plano de controlo) ou de um pacote de dados cujo *rank* e informação sobre a posição do nó emissor estão desactualizados (plano de dados). Se o *rank* do emissor for igual ou menor que o *rank* do seu nó pai, este último deverá emitir o seu DIO para o emissor, com o objectivo de actualizar o seu estado. A actualização disparará a reeleição de um novo nó pai caso existam melhores candidatos ou simplesmente alterará o *rank* permanecendo com o mesmo nó pai. Qualquer que seja a actualização realizada, o nó emissor inicial deve propagar posteriormente o seu DIO para que outros nó mais profundos possam também actualizar-se.

A Figura 5.22 apresenta um cenário mais crítico. A detecção multi-salto de falhas de encaminhamento através do plano de dados é, além de adequada, fundamental para a correcta resolução de falhas. Considere-se que um pacote de dados é emitido por um intermédio cuja ligação descobre estar comprometida. Existem duas soluções, dependentemente da aplicação. A demonstrada na Figura 5.22 prevê que o último intermédio renomeie um vizinho como nó pai, emitindo para este o pacote de dados. Se nenhum vizinho responder com um DIO, significa que os nós intermédios mais profundos, ligados ao nó sabem que perderam a sua ligação de encaminhamento e tentam igualmente arranjar ligações com DIS. Nesta situação, o nó emitirá o pacote de dados para o intermédio anterior, que tentará resolver a sua situação, emitindo o pacote para o nó pai que eleger. Se o mesmo ocorrer e não forem encontradas ligações, voltará a emitir o pacote para o intermédio anterior até o pacote voltar ao nó fonte. Partindo do pressuposto que os nós são capazes de identificar pacotes de dados duplicados, o nó fonte saberá que o seu último pacote de dados não foi entregue com sucesso. A partir do plano de controlo, a topologia é reorganizada e o nó fonte emitirá o pacote de dados através dos novos encaminhamentos estabelecidos.

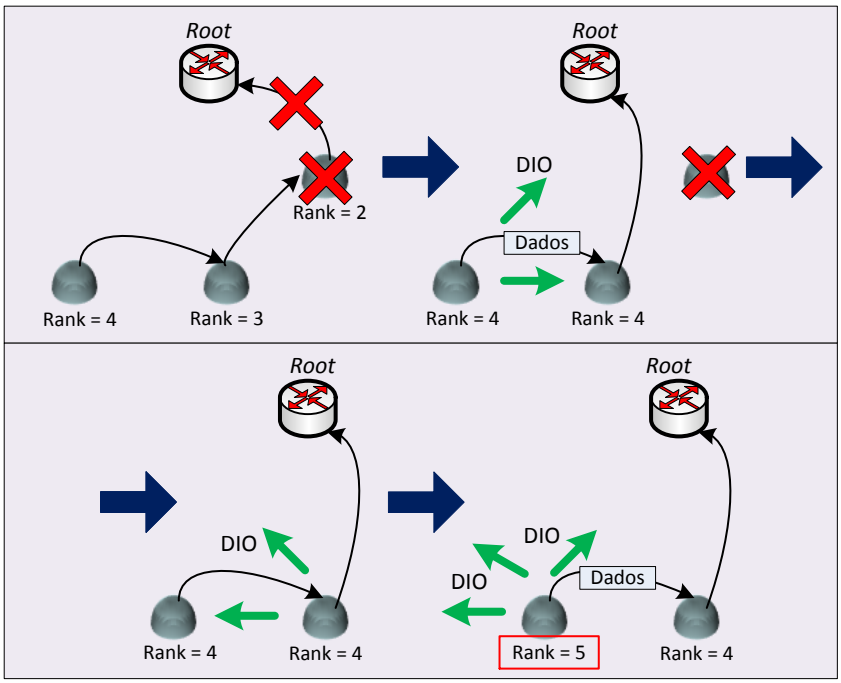

Figura 5.21 - Manutenção/actualização dos encaminhamentos.

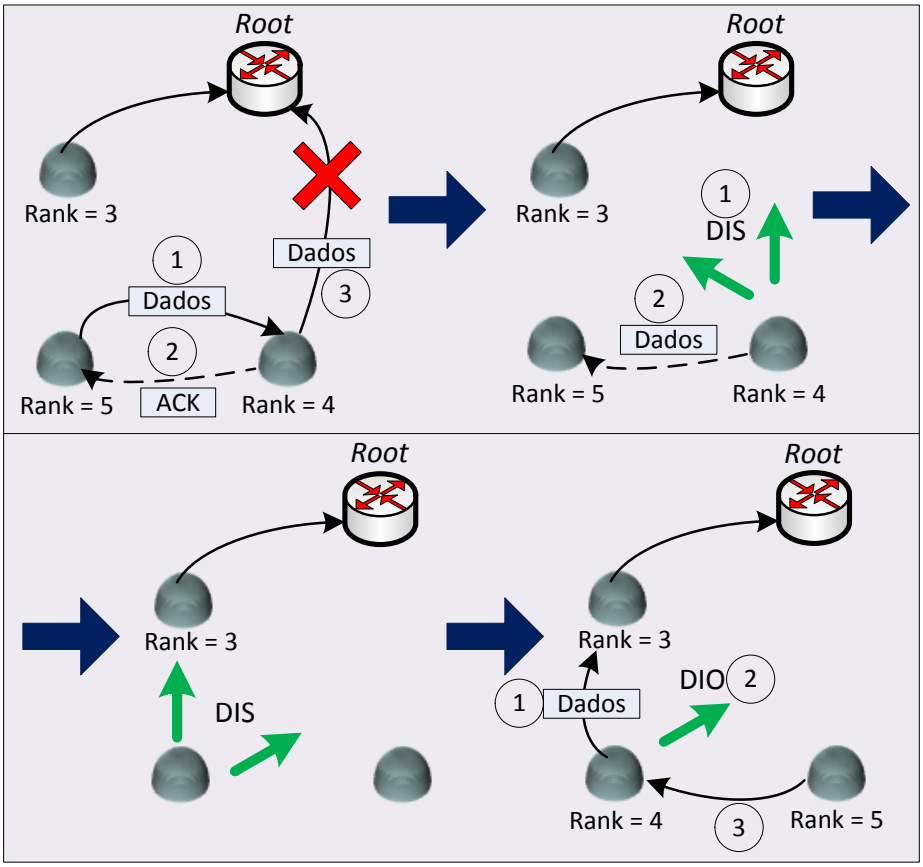

Figura 5.22 - Cenário crítico de detecção de uma falha durante a transmissão de dados.

5.3.5 Encaminhamentos de ligação descendente

As *Destination Advertisement Object* (DAO) são mensagens cuja finalidade é construir encaminhamento em ligação descendente, isto é, caminhos de cima para baixo da DODAG, normalmente pacotes provenientes do nó raiz para os nós da DODAG. Fundamentalmente, o mecanismo DAO é utilizado em pelo menos duas situações distintas. A primeira é a entrega P2P de pacotes de um nó de uma DODAG para outro nó da mesma DODAG. A segunda é a entrega de pacotes entre nós de uma DODAG e nós de outras redes, sejam de outras DODAGs ou de redes externas e distintas. Note-se que não é obrigatória a construção destes encaminhamentos em aplicações que não requerem tráfego descendente.

A mensagem de controlo DAO possibilita a construção de um estado de encaminhamento de ligação descendente em cada um dos nós que participará no dito encaminhamento. O objecto DAO é incluído em pacotes de controlo ICMPv6 e está representado na Figura 5.23.

Quando é colocada a 1 a *flag K*, é obrigatório a resposta de um DAO-ACK. A *flag D* é colocada a 1 quando o campo *DODAG ID* está presente (só estará presente se a instância da DODAG for local, e omitida se a instância for de natureza global).

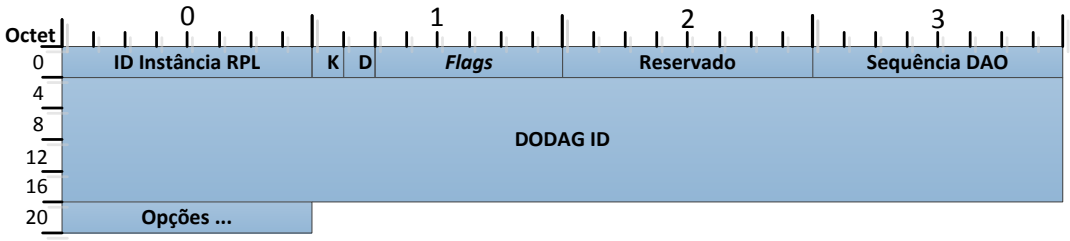

Figura 5.23 - Objecto DAO.

A construção dos encaminhamentos descendentes tem dois tipos de mecanismos:

- **Sem-memorização** (encaminhamento por *nó raiz*).
- **Com-memorização** (*stateful*);

O mecanismo *sem-memorização* alivia a ocupação extra de memória dos nós, mais concretamente em nós intermédios. Um nó fonte de dados emite o DAO para o seu nó pai, e este por sua vez envia o seu DAO para o seu nó pai, e assim sucessivamente, até ser atingido o *nó raiz*. O endereço de destino *unicast* de cada um dos DAOs é sempre o do nó raiz. No nó raiz é gravada a sequência de saltos do DAO entre cada um dos endereços participantes no encaminhamento construído. Qualquer um destes nós pode receber dados provenientes do nó raiz. A desvantagem deste mecanismo face a cenários densamente povoados e com inúmeras DAGs é a exigência dos endereços serem únicos e globais pois toda a decisão sobre o

encaminhamento de pacotes de ligação descendente é realizada pelo nó raiz ao nível da espinha dorsal das redes.

No mecanismo com-memorização os nós intermédios armazenam em tabela os endereços dos nós mais profundos, ocupando memória, mas com a vantagem de existir um número inferior de pacotes a ser emitido por toda a DODAG. Os encaminhamentos na ligação descendente são realizados apenas dentro da DODAG entre nós da mesma Instância. Um exemplo clássico é a comunicação de um nó sensor de chama para um nó actuador, que activa o sistema de anti-incêndio na divisão de uma casa. Ambos os nós encontram-se sob o mesmo contexto e na mesma DODAG, portanto, em algum ponto da sub-rede partilharão um nó intermédio em comum, que possibilitará o encaminhamento dos dados na ligação descendente. Os nós que operem com este mecanismo partilham os DAOs apenas para os seus nós progenitores, em unicast. Os respectivos tipos de endereços de origem e destino encapsulados devem obrigatoriamente os endereços globais e únicos.

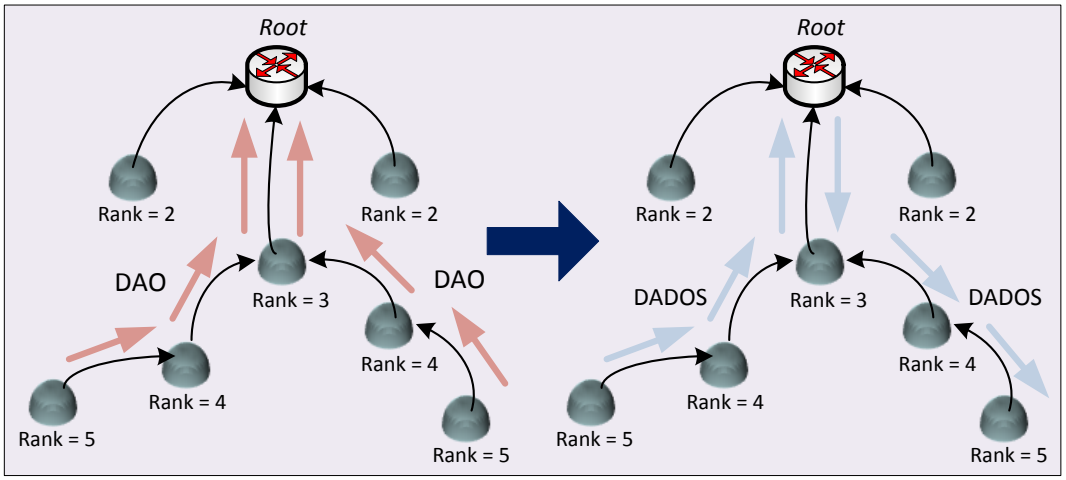

Figura 5.24 - Encaminhamento descendente Sem-Memorização.

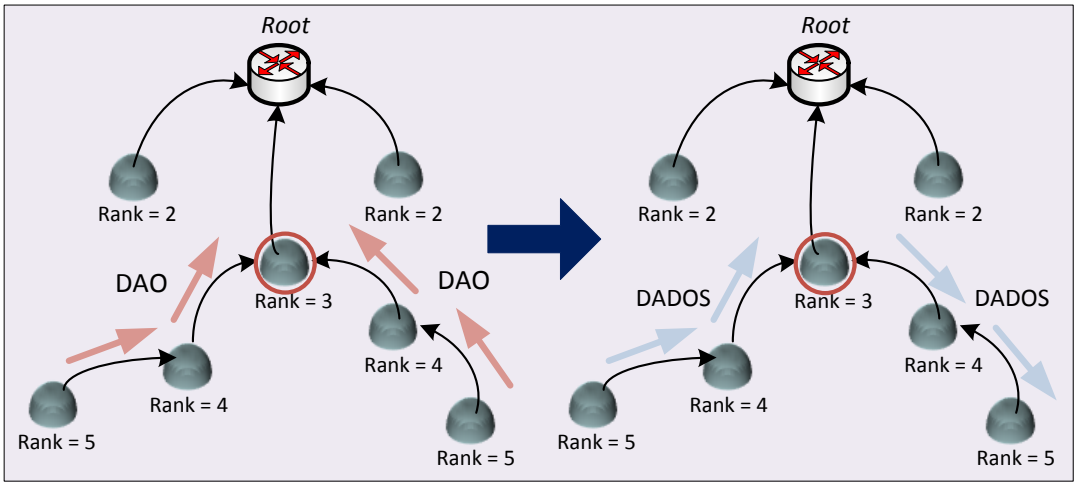

Figura 5.25 - Encaminhamento descendente Com-Memorização.

Um mecanismo hibrido pode oferecer um equilibrio confortável entre memória requerida, energia e largura de banda consumida por sinalização. Para encaminhamentos locais na ligação descendente, o mecanismo *com-memorização* é o mais indicado se uma DODAG não for superpovoada com centenas ou milhares de nós. No entanto, nestes cenários, uma DODAG pode ser dividida em sub-DODAGs (identificadas com prefixos de contextos específicos) por nós intermédios mais robustos em termos de recursos e que eficientemente interliguem as sub-DODAGs. Para existir partilha de tráfego entre redes diferentes, terá sempre de existir um mecanismo *sem-memorização* em que o nó raiz possui informação sobre todos os encaminhamentos possíveis para os nós da sua DODAG. A adição de nós intermédios mais robustos que agrupam as DODAGs internamente, pode aliviar este conflito. Os encaminhamentos pelas ligações descedentes principais são formados apenas entre o nó raiz e esses nós intermédios. A partir dos nós intermédios, o resto do encaminhamento é dividido para os nós mais profundos associados. Apesar dos pacotes de controlo DAO necessitarem de alcançar sempre o nó raiz, a troca de pacotes de dados entre nós da mesma DODAG não ocupará nem os canais, nem os nós acima dos nós intermédios necessários ao encaminhamento P2P. Este mecanismo é vantajoso em termos de consumo de energia e de ocupação da largura de banda da rede, além de suportar tráfego para outras redes. Contudo mantem a desvantagem da ocupação de memória extra nos intermédios. Além de ser aconselhável que estes nós intermédios sejam mais robustos em termos de recursos.

Devido à natureza do RPL e às características inerentes das topologias de rede, as ligações aos nós progenitores mudam, actualizam e falham. As alterações nos encaminhamentos de ligação ascendente influenciam a necessidade de também actualizar, em conformidade os encaminhamentos de ligação descendente. Assim, um pacote DAO deve ser emitido por um nó (seja o nó origem, ou seja um nó intermédio) sempre que um caminho é actualizado, ou sempre que se altera o nó pai.

Incluir o plano de dados pode igualmente ajudar na manutenção. Se um nó receber um pacote de dados de um nó mais profundo (ligação ascendente) do qual não recebeu anteriormente nenhum DAO e cujo *rank* é superior, pode ser construído um encaminhamento de ligação descendente. Isto ocorre em duas situações possíveis: ou o nó pai falhou por qualquer razão a recepção do DAO, ou ocorreu uma actualização recente seguida da emissão de pacotes de dados, na direcção de ligação ascendente, em lista de espera no *buffer*.

Sempre que um nó elimina a sua ligação com um nó pai com o qual também tem criado encaminhamento descendente, deverá emitir para este um DAO cujas *opções* incluem o campo *Informação de Tráfego* colocado a zero (0x00000000) que indica a perda de ligação. Esta mensagem DAO é denominada *No-Path*.

A presença de uma resposta DAO-ACK é essencial para prevenir a geração de caminhos fechados descendentes. Quando um nó A elimina o encaminhamento descendente com o seu

nó pai B, mas este último falha a recepção da notificação da alteração do encaminhamento, irá erroneamente passar o tráfego de ligação descendente para o nó A. A inclusão do DAO-ACK pode até um certo nível prevenir falhas desta natureza e garantir que o encaminhamento é eliminado com sucesso.

5.3.6 Métricas de Encaminhamento e Funções Objectivo

São propostas e descritas três métricas para calcular o custo dos encaminhamentos em RPL. O custo é calculado através de uma função objectivo, composta por uma ou várias métricas. São propostas algumas funções objectivo em função das métricas apresentadas a seguir.

1. **Número Mínimo de Saltos (*Minimum Hop Count*)**

Inicialmente, quando os protocolos de encaminhamento em redes sem fios de potência reduzida e restrições elevadas começaram a ser alvo de estudo, a métrica mais utilizada para criar caminhos era a contagem do número de saltos (*hop-count*) entre emissor e receptor, incluída na classe protocolar *distance-vector*. Especificamente, os protocolos DSDV [PB94] e AODV [PR99] assumem que todas as ligações entre nós ou são 100 % fiáveis ou não funcionam de todo, o que não é a abordagem mais correcta em redes sem fios. Devido à natureza dinâmica dos canais, especialmente em redes de potência reduzida e desempenho em termos de débito binário, as comunicações sofrem variações relacionadas com interferência e características concretas de cada aplicação tais como factores de *linha-de-vista*.

A métrica *hop-count* sofre ainda quebras de desempenho ao nível da rede, principalmente em redes bastante densas. Minimizar o número de saltos, aumenta a distância de cada salto, minimizando o RSSI e a taxa de perdas. E mesmo se o melhor encaminhamento for o que mantém o menor número de saltos, poderão existir outros encaminhamentos possíveis, com melhores desempenhos e melhor qualidade de serviço. Portanto, a escolha do encaminhamento com o menor número de saltos não será eventualmente o melhor encaminhamento possível [CABM03].

2. ***Expected Transmission* (ETX)**

A solução proposta em [CABM03] para a criação de caminhos é a métrica *Expected Transmission* (ETX). Através do ETX, os encaminhamentos são criados nas ligações onde se espera existirem o mínimo de transmissões e retransmissões de um pacote até chegar ao destino. Como o ETX resultante atribui um custo às ligações bidireccionais pela taxa de perdas que elas apresentam em ligações descendentes e ascendentes, força-se a escolha de encaminhamentos com débito binário elevado. O cálculo básico deste quociente pode ser simplesmente inferido através do número de transmissões, *t* de *p* pacotes de dados entregues com sucesso entre o nó ***x*** e o nó ***y*** [WTZA10] (5.9).

$$g(x,y) = t/p \tag{5.9}$$

O ETX faz uma contagem indirecta do número de saltos. Por exemplo, se a ligação entre dois nós (apenas de um salto) tiver uma taxa de sucesso de 50 %, o ETX será igual a 2. Mas se a mesma ligação contar com um nó intermédio, apoiando-se na entrega de pacotes em dois saltos, e se ambas as ligações tiverem uma taxa de sucesso de 10 0%, o ETX será também igual a 2.

Como referido, incluem-se as características bidirecionais. O ETX classifica as ligações utilizando as taxas de sucesso das ligações ascendentes e descendentes. A probabilidade de existir sucesso numa transmissão inclui por isso a probabilidade de sucesso da entrega (f_z) e a probabilidade de sucesso de uma resposta ACK (r_z), concluindo uma probabilidade total de [$f_z * r_z$]. O ETX total de uma ligação P2P pode ser calculada, tendo em conta os *n* saltos entre o emissor e o receptor final, através da seguinte Expressão:

$$ETX = \sum_{z=1}^{n} \frac{1}{f_z * r_z} \tag{5.10}$$

A proposta inicial do ETX focava-se essencialmente na norma IEEE 802.11b. As probabilidade de sucesso em ambas as direcções são calculadas com base na emissão *broadcast* de pacotes de controlo (*sondas*) com intervalos de tempo τ fixos entre cada um. Cada nó armazena uma variável em memória do número de *sondas* recebidas *w* de um só vizinho durante um período máximo σ. Assumindo esta estratégia, a taxa de entrega com sucesso num sentido é calculada, em sgundos através da seguinte Expressão [CABM03]:

$$h(w) = \frac{w}{\sigma/\tau} \tag{5.11}$$

Numa ligação entre o nó ***x*** e ***y***, ambos os nós podem utilizar esta estratégia e função para calcular tanto a taxa f_z como a taxa r_z. Apesar de em [CABM03] os autores sugerirem que o nó ***y*** sabe que de τ em τ deve receber *sondas* de ***x***, conseguindo calcular a taxa de perdas, ***y*** poderá desconhecer a existência de ***x*** enquanto não receber nenhum pacote deste. Portanto, a contagem em ***y*** só começa a ser realizada quando recebe a primeira *sonda* de ***x***, mesmo que tenha perdido *sondas* mais antigas.

As avaliações realizadas provam que a métrica ETX melhora o desempenho dos encaminhamentos quando comparada com a métrica *hop-count*.

Os trabalhos realizados em RPL consideram o ETX a sua métrica principal para o cálculo do *rank*. A sua utilização é baseada nas mesmas considerações feitas pelo protocolo CTP: um nó nunca deve eleger um nó pai com um ETX superior ao dele [GFJML09].

Devido às limitações energéticas das redes de sensores sem fios, a taxa de troca de *sondas* deve ser mais leve que a sugerida em [CABM03]. No Capítulo 6 é proposta uma estratégia para a obtenção dos valores de ETX para as redes de sensores sem fios.

Em [KETHVDTDC11] os autores implementam a camada de adaptação 6LoWPAN e camada de rede RPL nos sistemas operativos TinyOS e ContikiOS. Ambas as implementações actualizam o *rank* ETX_p após o envio do pacote *p* com a Expressão 5.13.

$$ETX_p = \alpha * T_p + (1 - \alpha) * ETX_{p-1} \tag{5.13}$$

Onde T_p é o número de tentativas de emissão do pacote *p*. α é uma constante definida durante a implementação (o ContikiOS considera $\alpha = 0.2$ e o TinyOS considera $\alpha = 0.5$). Esta Expressão é utilizada na solução proposta no Capítulo 6.

3. *Received Signal Strength Indicator* (RSSI)

O RSSI pode também ser utilizado como métrica para calcular o *rank* dos nós. Alguns trabalhos baseiam-se neste indicador porque as plataformas *hardware* têm capacidade para medir o valor do sinal recebido, incluindo o ruído e interferências de nós vizinhos (da mesma rede ou de redes diferentes) [FV12].

O protocolo *Mixed Hop and signal Received routing for mobile Wireless Sensor Networks* (MHRWSN) [FV12] utiliza os valores de RSSI medidos para calcular o custo, acumulando esse custo ao longo de vários saltos de encaminhamento como é praticado em RPL. A partir do cálculo do BER é possível verificar a taxa de erros em função dos valores de SNIR. O cálculo do BER definido pela norma [IEE03] é realizado através da seguinte Expressão:

$$BER = \frac{8}{15} * \frac{1}{16} * \sum_{k=2}^{16} (-1)^k * \binom{16}{k} * e^{\left(20 * SNIR * \left(\frac{1}{k} - 1\right)\right)} \tag{5.14}$$

Quanto mais se aproximar de zero, melhores serão os resultados e menor será o custo da ligação.

A partir do RSSI medido pelos nós, o valor de SNIR pode ser calculado a partir da seguinte Expressão:

$$SNIR = \frac{RSSI}{N + \sum_{i=1}^{n} I_i} \tag{5.15}$$

O ruído *N* é uma constante definida pela constante de Boltzmann ($k_B = 1.3803 * 10^{-23}$), pela temperatura ambiente, em kelvin ($T = 296K$, correspondente a 22.85ºC) e pela largura de banda ($B = 2.4 * 10^6 Hz$). No total, o ruído *N*, em watts e dBm, é:

$$N_W = k_B \,.T.B$$
$$= 9.806\, x10^{-12} mW \tag{5.16}$$
$$N_{dBm} = -110.085\ dBm$$

A grande maioria dos rádios emissores/receptores efectua leituras mínimas de RSSI que rondam os -100dBm (por exemplo, os rádios CC2520, CC2420, CC2430 e CC2500). Outros, efectuam leituras mínimas de – 91 dBm (por exemplo, os rádios ATRF86230 e ATRF86231). A maioria considera um máximo teórico registado de -10 dBm. A variação do SNIR com o RSSI para o intervalo RSSI = [-100 -10] dBm é apresentado na Figura 5.25.

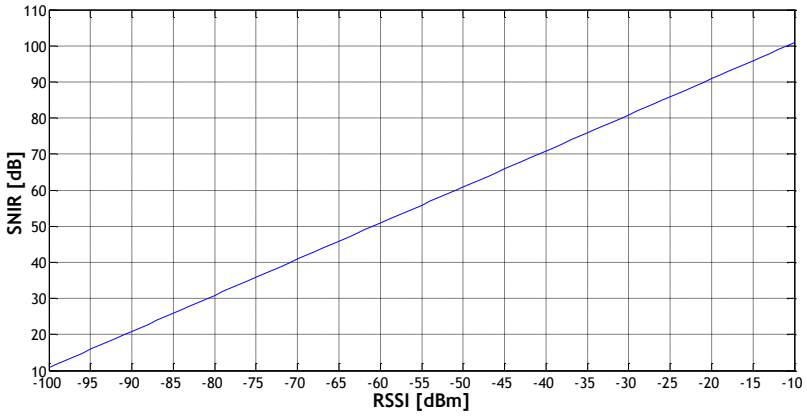

Figura 5.26 - Valores de SNIR em função do RSSI.

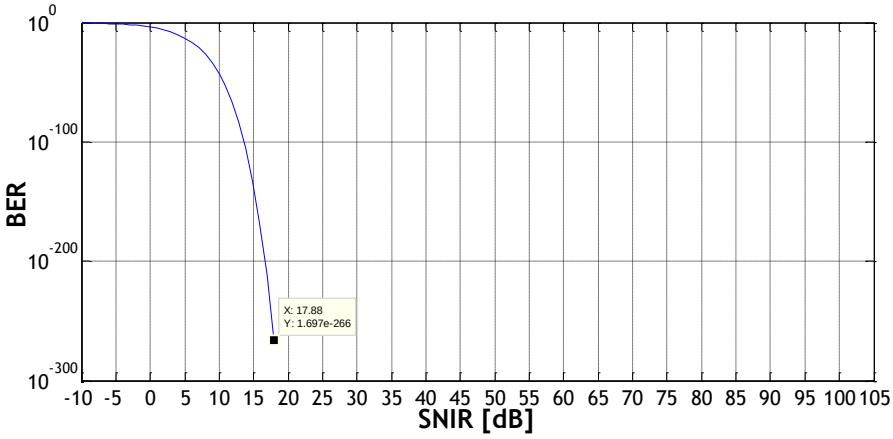

Figura 5.27 - BER em função do SNIR.

A Figura 5.27 representa os valores de BER em função de SNIR, no intervalo [-10 100] dB. Pode ser verificado que, para valores de SNIR acima de 18 dB (RSSI = 93 dBm), o BER é praticamente zero.

Para potências recebidas e calculadas, os valores RSSI, em Watts, para o intervalo [-100 -10] dBm encontram-se no intervalo $[10^{-10}\ 0.1]$ mW, como pode ser verificado nas Figuras 5.28 e 5.29.

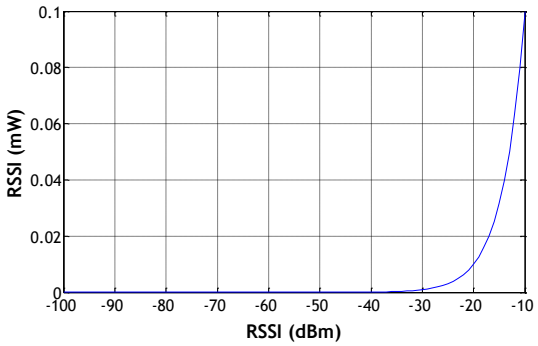

Figura 5.28 - RSSI (mW) vs RSSI (dBm).

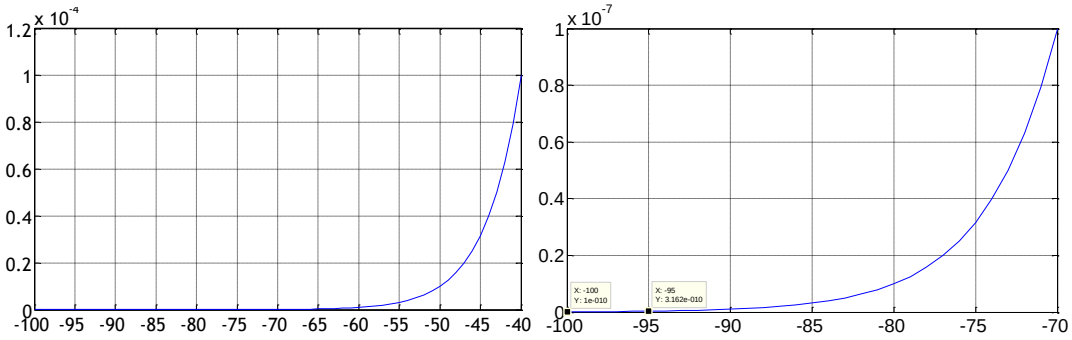

Figura 5.29 - RSSI (mW) vs RSSI (dBm).

Os valores RSSI calculados são armazenados em registos de memória específicos dos rádios emissores/receptores (por exemplo, no AT86RF230 ocupa os 5 bits menos significativos do registo *PHY_RSSI* de 8 bits, cujas conversões decimais estão compreendidos no intervalo [0 28], e onde o valor 28 corresponde a medições $\mathrm{RSSI} \geq -10\ dBm$, e o valor 0 corresponde a medições $RSSI < -91\ dBm$).

Apesar das fórmulas computacionais serem diferentes entre diferentes rádios emissor/receptor (por vezes até mesmo dentro da mesma família de rádios emissor/receptor), a grande maioria pode disponibilizar o valor do RSSI médio, preferencialmente em dBm. Este valor de RSSI é a média de várias medições de um só sinal. Armazenando para cada nó vizinho, na tabela de endereços vizinhos, a média RSSI de cada nova transmissão, esta pode ser utilizada como métrica. Por sua vez, poderá auxiliar na escolha do melhor nó pai quando o *rank* (por exemplo, calculado com a métrica ETX) é precisamente o mesmo, entre dois ou vários nós progenitores.

A Figura 5.30 e a Expressão 5.17 apresentam a variação *Free-Space Path Loss* (FSPL) com a distância para a banda de frequências de 2.4 GHz. A partir da fórmula de Friis, é possível

calcular a distância em função da potência recebida, P_r. A Expressão 5.18 calcula a potência recebida. Assume-se que os nós emitem o sinal uma potência $P_t = 0\ dBm$. A potência recebida (RSSI) num nó receptor é representada na Figura 5.31 em função da distância em metros.

$$FSPL(d) = 92.44 + 20 * \log(2.4) + 20 * \log\left(\frac{d}{1000}\right)\ [dB] \tag{5.17}$$

$$RSSI_p(d) = P_t - FSPL(d)\ [dBm] \tag{5.18}$$

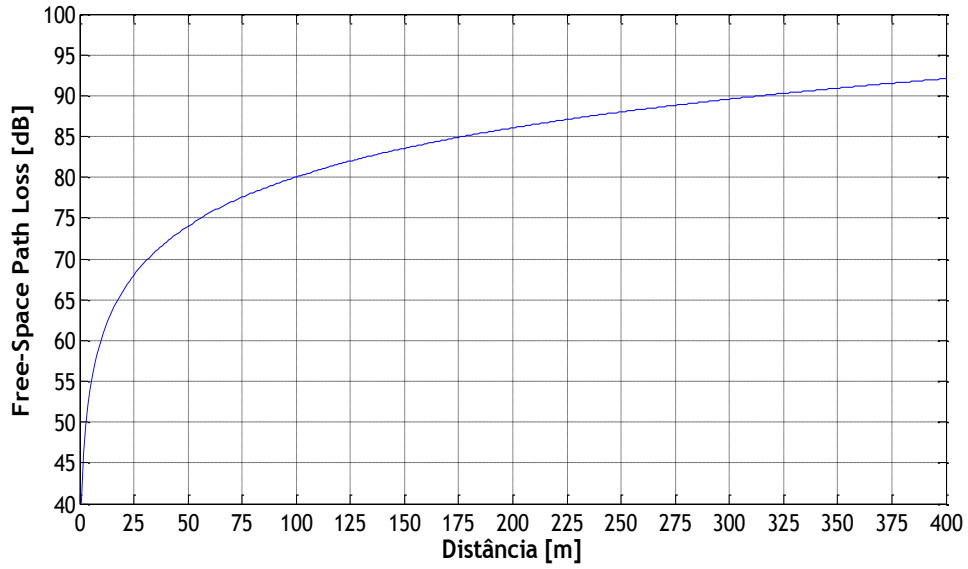

Figura 5.30 - *Free-Space Path Loss* em função da distância.

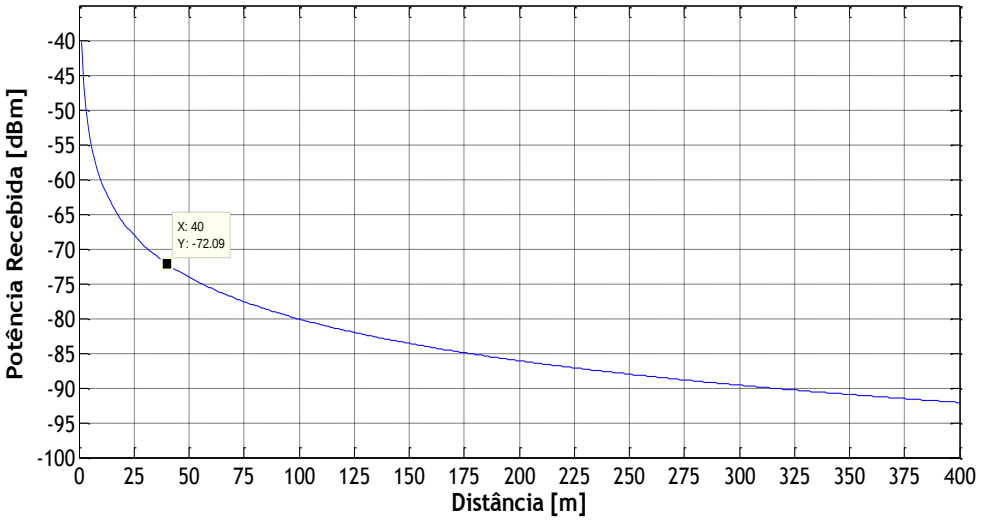

Figura 5.31 - Potência recebida em função da distância.

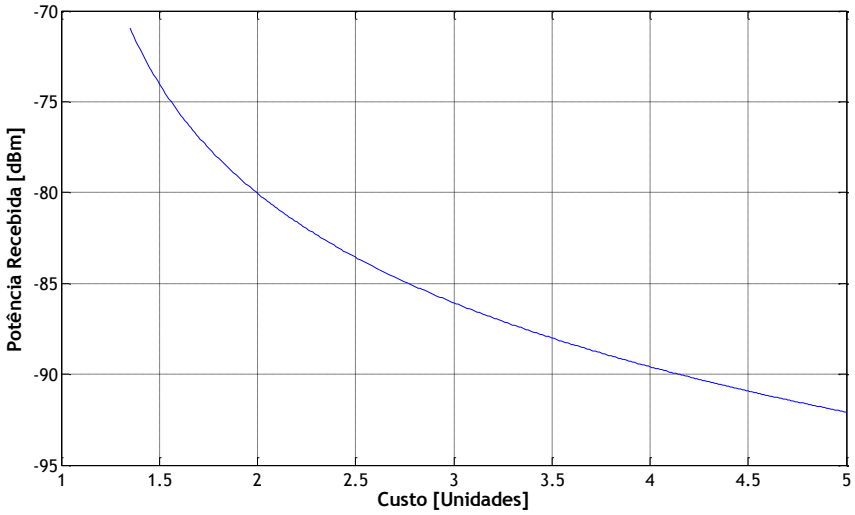

Figura 5.32 - Potência recebida em função do custo (potência menor que -72 dBm).

A partir dos valores de RSSI é definida uma função de custo para o encaminhamento que varia segundo a Expressão 5.19 no intervalo $R_p \rightarrow [1,5]$. A Figura 5.32 mostra o comportamento da potência recebida em função da função custo, R_p. Acima de -72 dBm, aproximadamente equivalente a uma distância de 40 metros entre os nós, o custo é igual a 1. Nesta situação considera-se que a probabilidade de ocorrerem erros é desprezável e que a qualidade da ligação é óptima.

$$R_p = \begin{cases} 1 & -72\ dBm \leq RSSI_p \\ 10 * 10^{\left(\frac{-RSSI_p - 100.044}{20}\right)} + 1 & -72\ dBm > RSSI_p \end{cases} \quad (5.19)$$

A Expressão 5.20, baseada na Expressão 5.13, actualiza o *rank* ao longo do tempo para um dado vizinho

$$RSSI_p = \alpha * R_p + (1 - \alpha) * RSSI_{p-1} \quad (5.20)$$

4. Funções Objectivo

A Expressão 5.21 e a Expressão 5.22 fazem o cálculo final do *rank* associado a um dado nó vizinho, para o ETX e RSSI, respectivamente. A função objectivo $RANK(i)$ utiliza o ETX/RSSI correspondente ao nó vizinho, $ETX\big(i, p(i)\big)$, e o *rank* desse mesmo nó, $RANK\big(p(i)\big)$.

$$RANK(i) = RANK\big(p(i)\big) * ETX\big(i, p(i)\big) + 1.0 \quad (5.21)$$

$$RANK(i) = RANK\big(p(i)\big) * RSSI\big(i, p(i)\big) + 1.0 \quad (5.22)$$

Adicionalmente, é proposta uma função objectivo que junta ambas as métricas. Esta junção pode tornar mais rigoroso o cálculo do *rank*. Definindo os custos de RSSI. É proposta uma função objectivo, Expressão 5.25, resultante das Expressões (5.23) e (5.24). A função objectivo final proposta é baseada nas funções definidas em [KETHVDTDC11], [WTZA10]:

$$T_p = \frac{R_p * ETX_p}{R_{p-1} * ETX_{p-1}} \tag{5.23}$$

$$C_i = \alpha * T_p + (1 - \alpha) * T_{p-1} \tag{5.24}$$

$$FO(i) = FO(p) * C_i + 1.0 \tag{5.25}$$

5.4 Sumário e Conclusões

Neste Capítulo foi descrita a camada de rede proposta pelo IETF que inclui uma camada de adaptação (situada entre a camada de ligação de dados e a camada de rede) especificada para atribuir endereços IPv6 a redes de potência reduzida e dispositivos cujos recursos são escassos. A atribuição dos endereços IPv6 é realizada utilizando o protocolo 6LoWPAN que comprime os cabeçalhos IPv6, tanto em modo *stateless* como em modo *stateful*, isto é, para encaminhamentos locais (*mesh*) ou globais (*route*), respectivamente. A capacidade dos nós em redes de sensores sem fios possuírem endereços lógicos IPv6, oferece a oportunidade destas redes integrarem-se com redes exteriores e a Internet, identificando-os globalmente e unicamente.

A camada de adaptação 6LoWPAN dá igualmente suporte à camada de rede. Para tal, o IETF definiu o protocolo de encaminhamento RPL, baseado no protocolo CTP, que permite construir encaminhamentos de forma eficiente através de *ranks* partilhados e calculados para cada nó, de uma forma implícita por toda a rede. O RPL é um protocolo hierárquico, que constrói uma topologia em árvore, com um nó central (nó raiz, ou *root*) como coordenador da rede e ponto de acesso entre as redes de potência reduzida e a espinha dorsal de redes exteriores. O protocolo prevê comunicações ascendentes e descendentes através de múltiplos encaminhamentos. Adicionalmente, são propostas neste Capítulo, técnicas de manutenção e recuperação de falhas dos encaminhamentos construídos com base nas características e conceitos especificados pelo IETF para o protocolo RPL.

Por fim, são descritas algumas métricas de custo que possibilitam o cálculo do *rank* e, portanto, as posições e funções dos nós na criação e manutenção dos encaminhamentos. As métricas de custo são discutidas seguindo algumas propostas de implementação e formulação de funções objectivo, baseadas em propostas presentes na literatura.

Capítulo 6

Comparação do Desempenho entre Métricas de Encaminhamento Distintas

Neste capítulo considera-se uma abordagem de simulação e são analisadas três métricas de encaminhamento discutidas no Capítulo 5: o ETX, o HOP-COUNT e o RSSI. As simulações foram realizadas através do simulador OMNeT++ [V01] e a framework MiXiM [KSW08]. Ambas as ferramentas são descritas no Anexo A e Anexo B, respectivamente. No Anexo C, é descrita a implementação do protocolo RPL no simulador. O foco principal das simulações e análise realizadas concentra-se na criação e manutenção dos encaminhamentos descendentes desde os nós geradores de dados até ao nó central agregador de dados, o nó raiz. Assim, só é descrita e analisada a estratégia para a partilha dos pacotes *DODAG Information Object* (DIO) e do cálculo do *rank* descendente.

As simulações e posteriores estudos realizados baseiam-se na aquisição de valores de métricas de referência, para perceber o nível de qualidade e eficiência das redes em causa como, por exemplo, o débito binário útil (*goodput*). O *goodput* é a taxa de transferência do *payload* de dados, recebido sem erros, no nó receptor [QC02]. O objectivo das simulações é demonstrar a eficiência das métricas e do protocolo RPL no âmbito da entrega/recepção de dados ao longo dos encaminhamentos construídos até ao nó raiz, isto é, a medição do QoS considerando o débito binário útil ao nível da aplicação [KKSBM12]. Assim, é utilizado o termo *goodput* ao invés do termo *throughput*, sendo o primeiro o número total de bits recebidos, e o segundo, o número de bits de dados recebidos.

Em todas as simulações, o *goodput* é calculado segundo a Expressão 6.1, que segue a mesma formulação utilizada em [IJM11] para obter o débito binário agregado máximo:

$$goodputByte = \left[\frac{nbRX}{nbTX}\right] * \left[\frac{N * B}{tPeriodic}\right] \quad (6.1)$$

onde $nbRX$ é o número total de pacotes recebidos no *nó raiz*, $nbTX$ é o número total de pacotes emitidos por todos os nós da rede, *N* é o número de nós, *B* é o valor, em bytes, do *payload* de dados (dados da camada de aplicação) e $tPeriodic$ a periodicidade da geração e emissão dos pacotes de dados.

As simulações em OMNeT++ disponibilizaram outros valores tal como o tempo de convergência por nó, que representa o tempo que um nó demora a negociar com os outros nós activos da rede e a estabelecer a sua ligação com esses nós vizinhos de maneira a participar

activamente na rede e nos encaminhamentos. Outro valor envolvido na análise é a latência extremo-a-extremo que representa o tempo que um pacote demora a ser criado, emitido, re-encaminhado e recebido no nó raiz.

Inicialmente é proposta uma solução para a implementação da métrica ETX. Os trabalhos relacionados com a métrica ETX em redes de sensores sem fios propõem funções objectivo e algoritmos energeticamente eficientes de actualização dos *ranks* consoante os valores de ETX. No entanto, nenhum especifica como esses valores de ETX são adquiridos pelos nós da rede de sensores sem fios. Alguns trabalhos propõem estratégias para esse efeito mas todos baseiam-se na norma IEEE 802.11, para a qual o ETX foi inicialmente sugerido e especificado. O ETX original é, como explicado no Capítulo 6, calculado tendo em conta as ligações bidirecionais, os tempos de espera do *Trickle* e devido a algum *overhead* causado pela troca de pacotes de sonda que possibilitam o cálculo da probabilidade ETX. A implementação inicial definida em [CABM03] envia os seus pacotes de sonda com comprimentos fixos e bastante reduzidos, em difusão. As sondas enumeradas são emitidas de um em um segundo (adiciona-se um tempo *jitter* para evitar colisões), não sendo retransmitidas nem respondidos por ACK. Os pacotes de sonda são emitidos durante 10 segundos (resultando, no máximo, 10 pacotes de sonda emitidas por cada nó). Os nós gravam quantos pacotes de sonda receberam de um determinado vizinho durante σ segundos, permitindo calcular a probabilidade de ETX segundo a Expressão 5.11.

Como o ETX em IEEE 802.11 é calculado segundo a probabilidade de entregar com sucesso pacotes de dados em ambos os sentidos, ascedente e descendente, todos os emissores devem manter em memória o número de pacotes de sonda recebidos por cada nó vizinho, e incluir esses valores nos seus próprios pacotes de sonda, possibilitando o cálculo das taxas de envio e taxas de resposta, f_z e r_z respectivamente. O ETX é então calculado a partir da Expressão 5.10.

Apesar dos excelentes resultados demonstrados [NLM08], [CABM03], e da robustez do algoritmo, a estratégia tem vários desafios quando implementadas em redes de sensores sem fios. A primeira é o excessivo *overhead*, não só devido à elevada quantidade de pacotes de sonda emitidos num curto intervalo de tempo, mas igualmente devido a situações onde existe um elevado número de nós vizinhos a propagarem os seus pacotes de sonda. Na mesma medida, o cálculo através dos intervalos de tempo pode-se tornar complexo em situações particulares, tal como a falha de recepção dos primeiros pacotes de sonda e o armazenamento em memória dos tempos e pacotes de sonda de todos os vizinhos.

Durante a fase inicial de construção e troca dos primeiros pacotes de sonda, um segundo de intervalo entre cada pacote de sonda pode introduzir elevados atrasos em RSSF que cumprem tempos de transição e comunicação bem mais reduzidos que as redes Wi-Fi e MANET. Além

disso, em redes bastante populosas, os pequenos desvios de tempo (*jitter*) do *Trickle* podem mesmo causar muitas colisões entre pacotes de sonda.

Devido a estes aspectos, é proposto um novo algoritmo para cálculo do ETX em redes de sensores sem fios.

6.1 Métrica ETX-WSN

O nó raiz inicia a construção da rede com a propagação do seu DIO. O DIO é utilizado como pacote de sonda pois o mesmo permite que todos conheçam o *rank* dos vizinhos e por ter um PPDU reduzido de 40 bytes (consideram-se todos os 25 bytes de cabeçalho do DIO). Todos os nós que recebam o DIO do nó raiz calendarizam a propagação dos seus próprios DIOs num intervalo entre 500 ms e 1 segundo escolhido aleatoriamente. Este intervalo de tempo corresponde à variável temporal τ_h do *Trickle*. Inicialmente, todos os nós possuem *rank* nulo. Ao contrário do algoritmo original, este esquema não necessita de um temporizador para calcular quantos pacotes de sonda são recebidos do lado dos receptores dos DIOs. Propõe-se a utilização de dois campos inutilizáveis dos pacotes DIO, *flags* e *reservado*. Estes dois campos são utilizados para dois identificadores: DIO-ID e DIO-SEQ. Estes dois identificadores possibilitarão aos receptores reconhecer quantos DIOs foram recebidos por um respectivo nó emissor. O DIO-ID é o número do DIO emitido. O campo DIO-SEQ representa um grupo de DIO-IDs.

Um nó emite 4 pacotes DIO por cada grupo DIO-SEQ. O DIO-SEQ é composto por 8 bits e, portanto, tem uma combinação de 1 até 256. Consequentemente, cada nó emitirá 255*DIO-ID pacotes DIO, antes de reinicializar de novo o contador DIO-SEQ. Sempre que uma sequência termina, o receptor emite uma resposta DIO-ACK com o número de DIOs recebidos de uma determinada sequência. Para cada um, o nó pode calcular o ETX. Assim, podem ocorrer três situações, como será descrito a seguir.

O algoritmo *Trickle* proposto inclui o intervalo de tempo mínimo, $\Delta\,\tau_l$, e máximo, $\Delta\,\tau_h$. O intervalo $\Delta\,\tau_l$ é sempre utilizado na propagação de DIO-IDs duma única sequência. Este intervalo é igualmente utilizado entre cada sequência SEQ-DIO enquanto um nó (ou a rede) não convergir. Quando o nó converge e não ocorre nenhuma alteração, o intervalo de tempo entre cada sequência é o intervalo máximo, $\Delta\,\tau_h$.

- **1º ETX = 100 %**

A Figura 6.1 mostra os pacotes DIO que um nó recebeu de um nó vizinho. A recepção de todos os pacotes DIO indica que existe uma probabilidade igual a 100 % do nó receber todos os pacotes enviados pelo nó emissor vizinho. Nesta situação, o ETX é igual a 1.

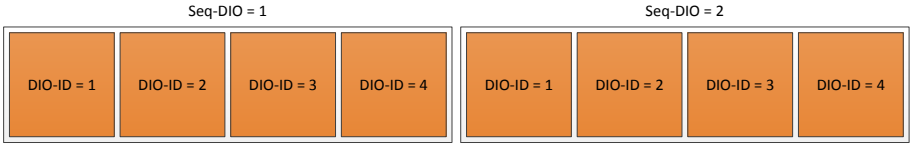

Figura 6.1 - ETX = 100 %.

- **2º ETX < 100 % com recepção do último DIO-ID**

A Figura 6.2 representa a sequeência de recepção com perdas de pacotes DIO. Neste caso o ETX é maior que 1. Referente ao exemplo da Figura 6.2, durante a sequência DIO-SEQ = 1, o receptor apenas recebe 2 pacotes dos 4 emitidos. O ETX é calculado segundo Expressão 5.9.

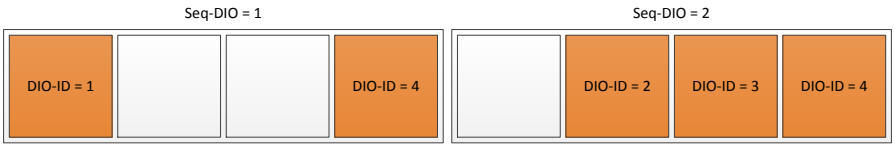

Figura 6.2 – Sequência de recepção com perda de pacotes DIO para ETX<100 %, com recepção do último DIO.

$$ETX_p = \frac{4}{2} = 2$$

Durante a sequência DIO-SEQ = 2 apenas um pacote não é recebido correctamente. O custo do ETX é calculado segundo a Expressão 5.13, tendo $\alpha = 0.5$:

$$ETX_{p+1} = \frac{4}{3} = 1.333$$

$$ETX_t = ETX_p * \alpha + (1 + \alpha) * ETX_{p+1} =$$

$$= 2 * 0.5 + (0.5) * 1.333 = 1.6665$$

- **3º ETX < 100 % sem recepção do último DIO-ID**

Este é um caso especial onde o último DIO-ID não é recebido. Aqui, como se apresenta na Figura 6.3, o receptor não emitirá imediatamente o DIO-ACK. Por sua vez o emissor não poderá calcular o ETX, depreendendo a ocorrência de uma de duas situações possíveis, ou falhou o envio do último DIO ou falhou a recepção do DIO-ACK. Neste caso, o emissor ignorará e calendarizará a próxima sequência normalmente. Como se apresenta na Figura 6.3, o receptor receberá o segundo DIO da sequência 2 e detetará que falhou a recepção do último DIO da sequência anterior. A partir daqui, a resposta DIO-ACK acompanhará o número da sequência anterior e número de DIOs dessa mesma sequência, permitindo, posteriormente, o cálculo do ETX segundo a Expressão 6.2. O resto do algoritmo decorre normalmente.

Figura 6.3 - Sequência de recepção com perda de pacotes DIO para ETX<100 %, sem recepção do último DIO-ID.

$$ETX = \frac{t * res}{p} \tag{6.2}$$

onde *res* é o número de sequências atrasadas calculada pela Expressão 6.3. O número da sequencia correcta e actual é *actSeqDIO* e o número da sequência atrasada é *atrSeqDIO*.

$$res = actSeqDIO - atrSeqDIO \tag{6.3}$$

6.1.1 Variação do número de pacotes DIO por sequência

A métrica ETX-WSN foi originalmente idealizada para emitir quatro pacotes DIO (DIO-ID=4) em cada sequência SEQ-DIO. No entanto, o número de pacotes DIO pode variar consoante os objectivos pretendidos de uma aplicação. Foram realizadas simulações variando o número de pacotes DIO emitidos para calcular o ETX e o número de nós activos numa rede. As simulações foram realizadas para um período de operação igual a 10 minutos (600 segundos). Os primeiros 30 segundos são dedicados à construção dos encaminhamentos e, portanto, nenhum pacote de dados é gerado e emitido durante esse período. Após um intervalo de tempo, os nós geram e emitem pacotes de dados de 10 em 10 segundos, periodicamente. Nesta Secção, as simulações foram realizadas variando o número de nós geradores de dados, presentes e activos na rede, excluindo o nó central nó raiz. Para estas simulações, o intervalo de tempo mínimo, $\Delta\,\tau_l$, escolhido encontra-se no intervalo [0.5, 1] segundos, enquanto que o intervalo de tempo máximo, $\Delta\,\tau_h$, é igual a 15 segundos.

As Figuras 6.4 e 6.5 apresentam os resultados de *goodput* agregado no nó raiz, em percentagem e em bytes, respectivamente. Observa-se que uma sequência com quatro DIOs tem um nível de *goodput* mais elevado que as sequências de três e dois DIOs, devido à probabilidade mais elevada de criar encaminhamentos mais precisos e melhorados. Esta elevada precisão duma sequência de quatro DIOs origina-se devido à elevada troca de DIOs, como sondas, quando comparada com as sequências de dois e três DIOs, e portanto, mais preciso serão os *ranks* e, consequentemente, mais precisos, robustos e fiáveis serão os encaminhamentos.

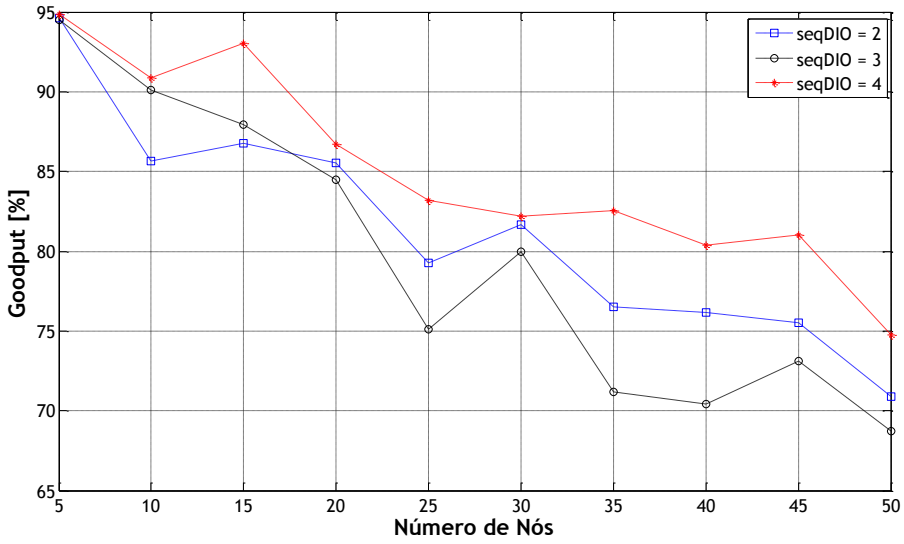

Figura 6.4 – *Goodput* em percentagem para diferentes sequências em função do número de nós.

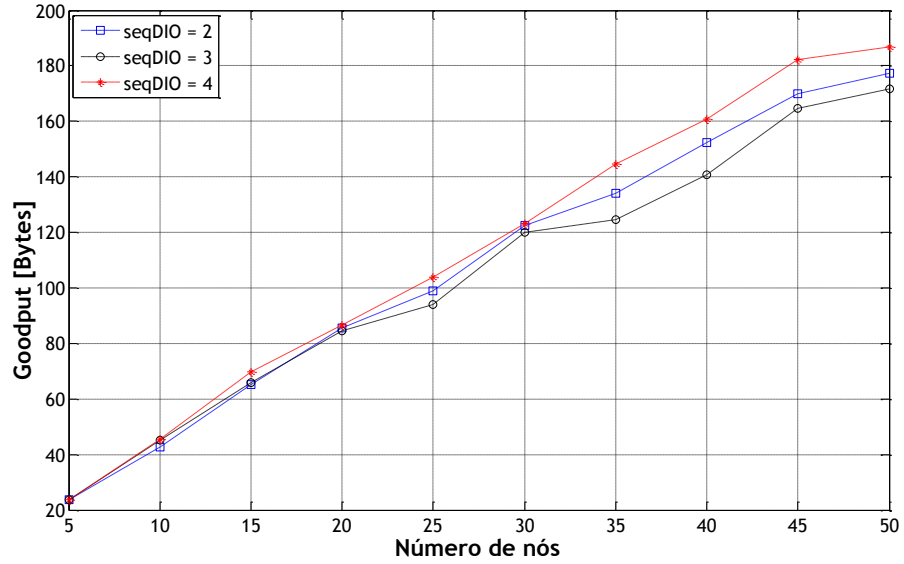

Figura 6.5 – *Goodput* em bytes para diferentes sequências em função do número de nós.

A Figura 6.6 apresenta a medição do tempo de convergência médio de toda a rede em função do número de nós. As sequências de dois DIOs apresentam os melhores tempos de convergência. Por outro lado, as sequências de quatro DIOs apresentam os tempos de convergência mais elevados. Estes resultados devem-se à existência de trocas de DIOs mais frequêntes quando a sequência é de quatro e, portanto, maior é o tempo até um nó determinar o *rank* e os nós estarem associados aos respectivos nós progenitores.

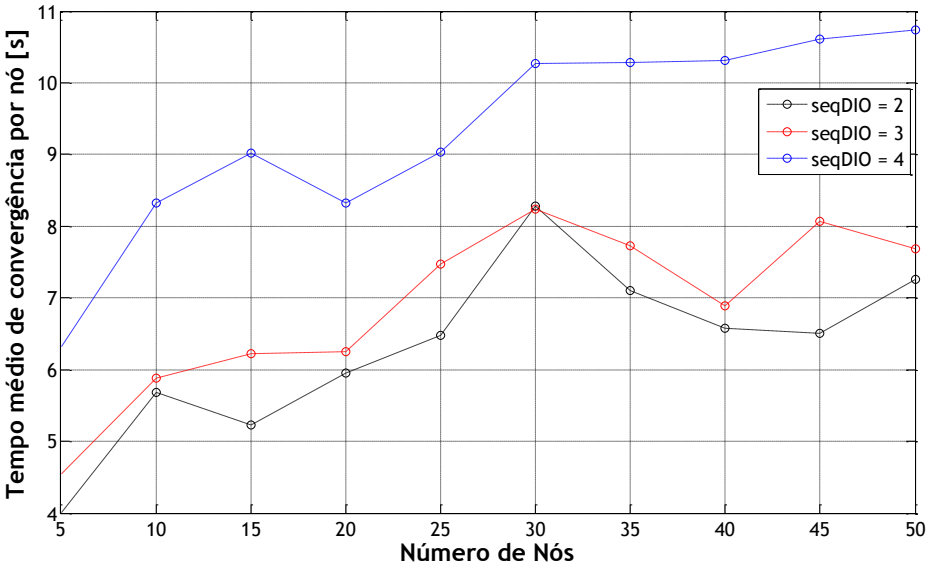

Figura 6.6 – Tempo de convergência por nó para diferentes tipos de sequência em função do número de nós.

Por outro lado, a Figura 6.7 apresenta o número médio de reconvergências por nó. Quando um nó actualiza um encaminhamento ou começa a participar noutros encaminhamentos, poderão ter ocorrido pelo menos duas situações. O nó pai pode ter deixado de comunicar e, portanto, o nó em questão associa-se a outro nó pai, ou o nó pode ter descoberto um nó pai que lhe oferece melhores condições para encaminhar os seus pacotes. Nas duas situações existe uma troca de nó pai, portanto, ocorrerá uma actualização do *rank* e, consequentemente, a actualização do encaminhamento (ou até mesmo em situações mais extremas, de toda a rede). Esta actualização de posição (*rank*) é uma reconvergência.

As reconvergências são extremamente importantes. A actualização do *rank* de um determinado nó pode não incidir na escolha de um novo nó pai. Isto é, apesar da actualização do *rank* (para piores valores) o nó pai pode continuar a ser o mesmo (desde que todos os outros nós apresentem piores escolhas). No entanto, estes encaminhamentos apresentam um "efeito dominó". Significa que apesar do nó permanecer com o mesmo nó pai, só o simples facto de actualizar o seu *rank*, pode afectar as escolhas e actualizações de todos os outros nós "abaixo" de si, ao longo do encaminhamento, levando à reconvergência destes.

A Figura 6.8 apresenta a latência média em função do número de nós. Este é o tempo decorrido durante a criação, emissão e recepção de um pacote, sucessivamente ao longo de um encaminhamento, até ser entregue com sucesso no nó raiz. A diferença entre os três tipos de sequências não é elevada. No entanto, notam-se atrasos menores quando a sequência é três DIOs, e, em algumas situações, latências mais longas quando a sequência é de quatro DIOs.

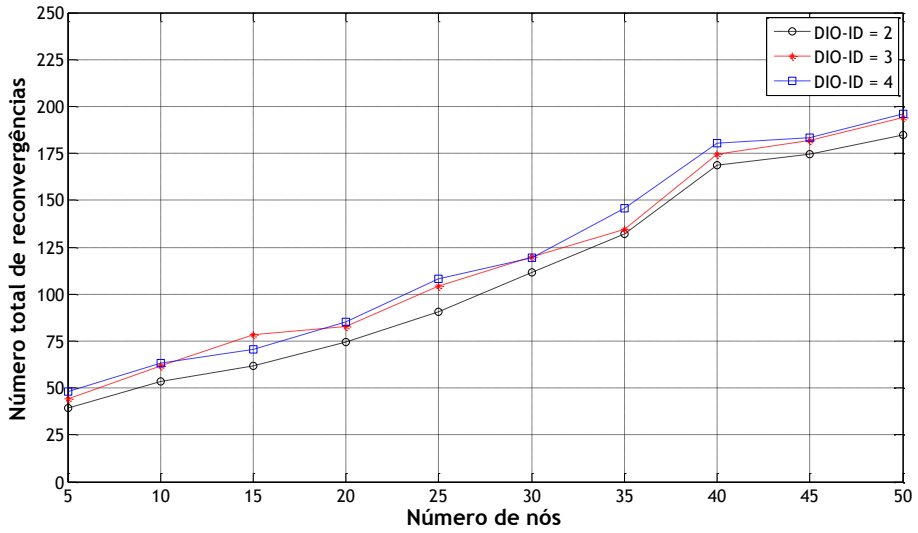

Figura 6.7 – Número total de reconvergências para diferentes tipos de sequência em função do número de nós.

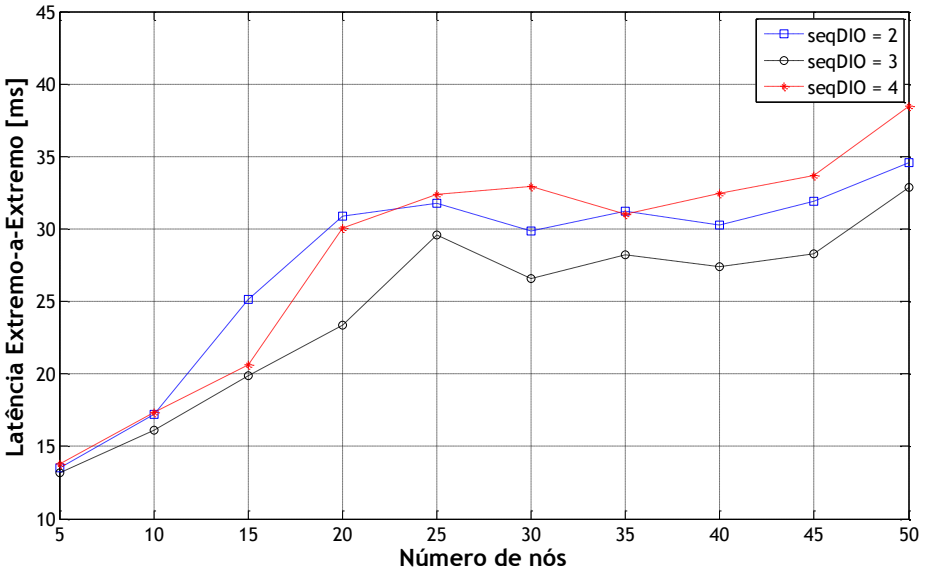

Figura 6.8 – Latência média extremo-a-extremo para diferentes tipos de sequência em função do número de nós.

Latências superiores (atraso extremo-a-extremo) significam que o mesmo pacote percorre mais nós e/ou maiores distâncias até chegar ao destino. Este estudo é importante na avaliação de aplicações onde o tempo de entrega de dados é crítico, isto é, necessita de ser bastante reduzido. Na mesma medida, mais nós participam na entrega de um só pacote. Portanto, mais energia é consumida pela rede. No entanto, latências superiores podem significar melhores débitos binários pois, normalmente, verifica-se que esses encaminhamentos são mais precisos.

6.1.2 Intervalos de tempo entre DIOs dentro de uma sequência

A escolha do intervalo de tempo $\Delta \tau_l$ entre cada DIO-ID pode ser bastante relevante, consoante os equilíbrios (*trade-offs*) que se pretendam manter dentro dos limites aceitáveis tais como tempos de latência, débito binário (*throughput*), consumo de energia, quantidade de tráfego. O estudo realizado nesta Secção considera vários intervalos $\Delta \tau_l$ com um intervalo de tempo $\tau_h = 15\ s$ e um número total de 26 nós, incluindo o nó raiz, para um tempo de operação total igual a 5 minutos (300 segundos). O intervalo mais reduzido está incluído no intervalo aleatório $[0.5 < \tau_l < 1]$ segundos. O intervalo mais longo encontra-se no intervalo aleatório $[5.5 < \tau_l < 6]$ segundos. A Figura 6.9 apresenta o tempo médio de convergência da rede para diferentes intervalos de tempo $\Delta \tau_l$. À medida que o intervalo de tempo entre DIO-IDs aumenta, maior será o tempo até os nós calcularem a sua posição e associarem-se a um nó pai e, portanto, participarem num encaminhamento.

Por outro lado, a Figura 6.10 apresenta o número total de reconvergências da rede. Os resultados obtidos são extremamente relevantes. Primeiro, quanto maior o intervalo de tempo entre os DIO-IDs de uma só sequência, menor será o número de actualizações ao longo dos 5 minutos de simulação. Segundo, uma sequência de quatro DIOs possui um menor número de reconvergências. Na prática, a sequência de quatro DIOs tem um número mais reduzido de actualizações devido, não só, ao número mais elevado de DIOs numa sequência, mas também devido ao alto grau de precisão no cálculo do *rank*. Quanto mais preciso e rigoroso for o cálculo do ETX, menos actualizações sofrerão os nós, pois escolherão mais rapidamente os melhores nós sensores progenitores, à medida que os forem descobrindo.

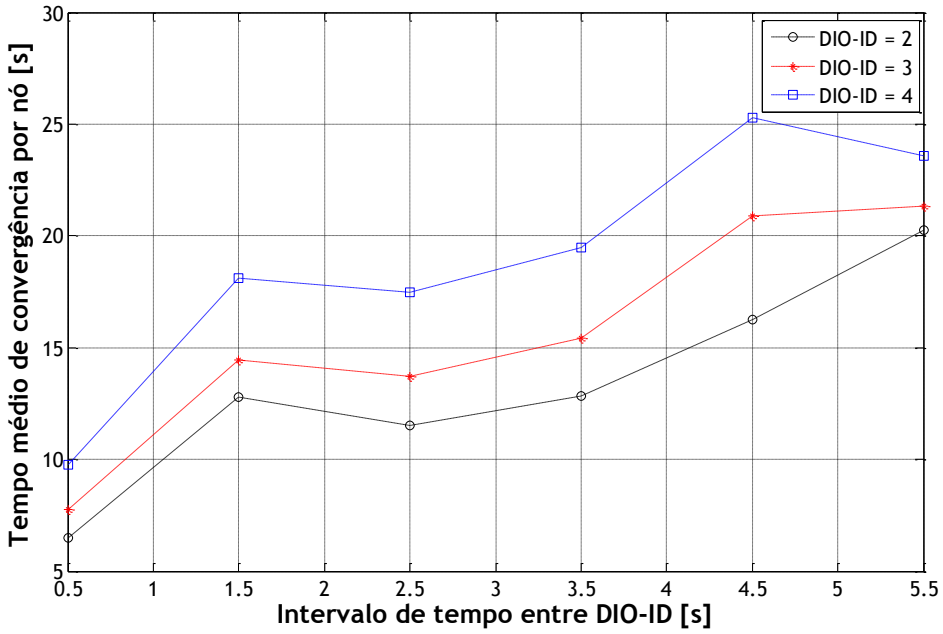

Figura 6.9 – Tempo médio de convergência por nó em função do intervalo de tempo entre DIOs de uma sequência.

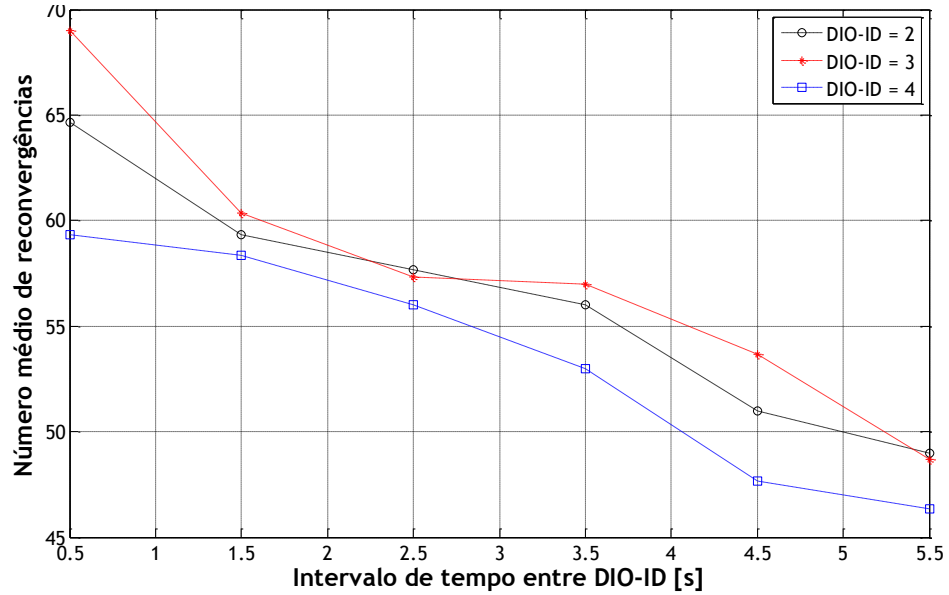

Figura 6.10 – Número total de reconvergências da rede em função do intervalo de tempo entre DIOs de uma sequência.

As Figuras 6.11 e 6.12 apresentam o número de DIOs emitidos e recebidos por nó, respectivamente. Quanto maior o número de DIOs emitidos por sequência, maior será o número total de DIOs emitidos por nó. Consequentemente, maior será o número total de DIOs recebidos. Novamente, quanto mais DIOs forem recebidos, mais preciso é o cálculo das posições e melhor é o encaminhamento. Por outro lado, mais energia é desperdiçada com um número mais elevado de DIOs a serem trocados entre todos os nós.

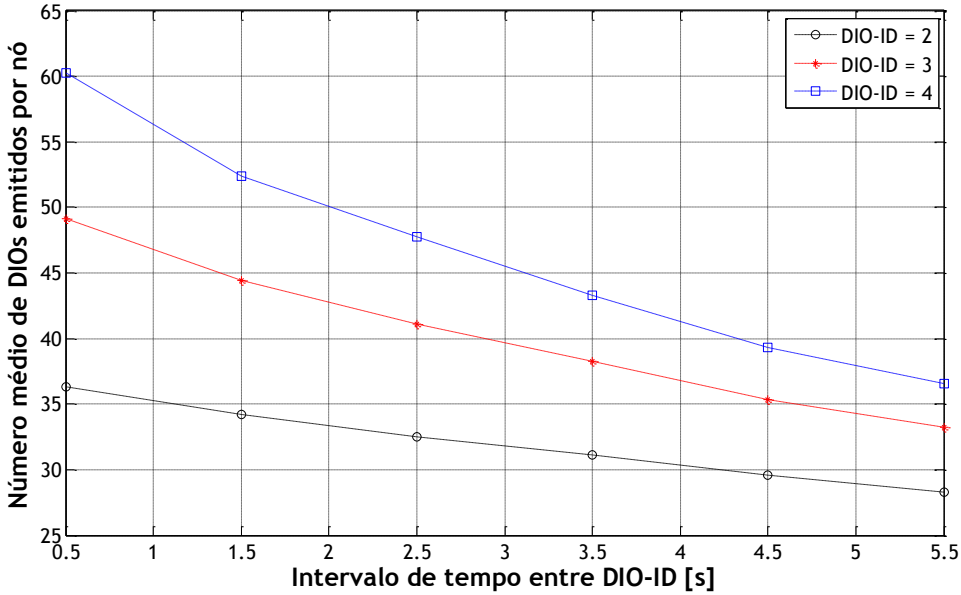

Figura 6.11 – Número médio de DIOs emitidos por nó em função do intervalo de tempo entre DIOs de uma sequência.

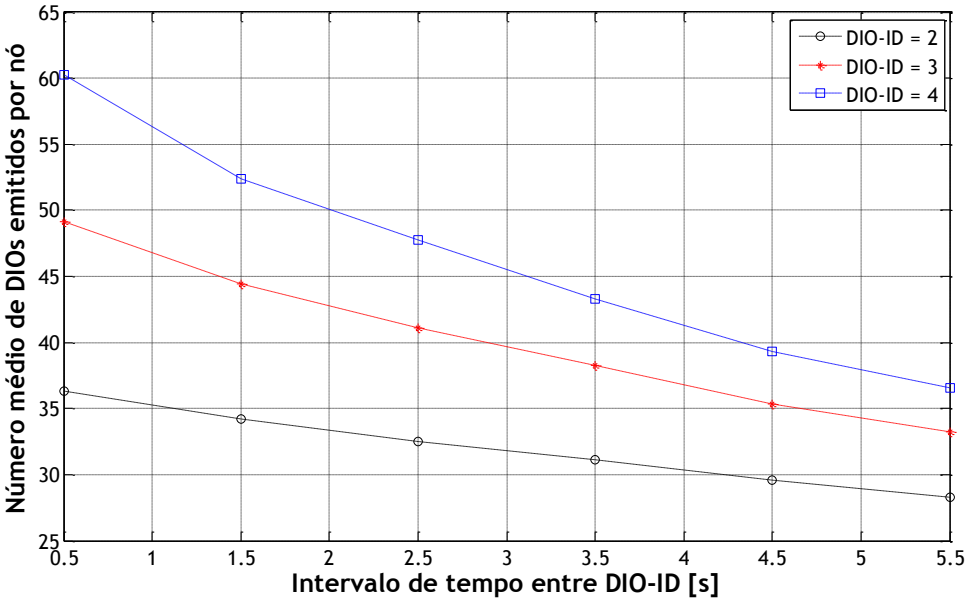

Figura 6.12 – Número médio de DIOs recebidos por nó em função do intervalo de tempo entre DIOs de uma sequência.

A Figura 6.13 mostra que, quanto maior for a troca de DIOs por sequência, mais preciso é o cálculo das posições e melhores são os encaminhamentos. Para sequências de quatro DIOs, o *goodput* é superior, principalmente quanto o intervalo de tempo entre DIOs está acima do intervalo [1.5, 2] segundos, decrescendo à medida que o intervalo aumenta.

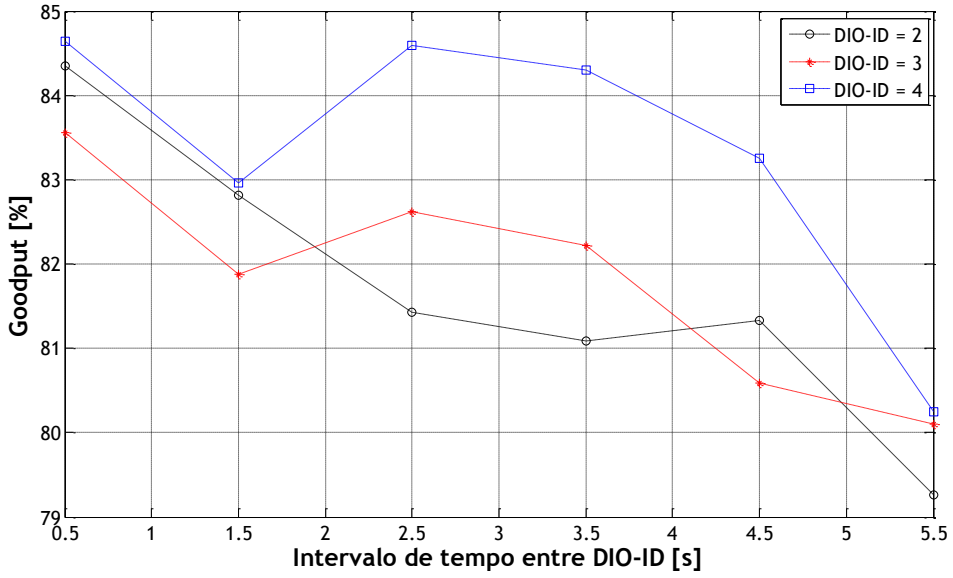

Figura 6.13 – *Goodput* em percentagem em função do intervalo de tempo entre DIOs de uma sequência.

6.1.3 Intervalos de tempo entre sequências de DIOs (SEQ-DIO)

Nesta Secção são apresentados resultados de simulação variando o intervalo de tempo entre sequências de DIO (SEQ-DIO), isto é, o intervalo de tempo máximo $\Delta\,\tau_h$. Considera-se um intervalo de tempo fixo entre DIOs de uma só sequência, neste caso o intervalo mais reduzido estudado na Secção 6.1.2, $\tau_l = [0.5, 1]\text{segundos}$. As simulações foram realizadas para 7 intervalos de tempo máximo entre sequências. Consideram-se $\tau_h = \{3, 7, 10, 12, 15, 17, 20\}$ s.

A Figura 6.14 apresenta os resultados para o tempo médio de convergência por nó. A diferença de tempos de convergência ao longo de diferentes tempos $\Delta\,\tau_h$ não é elevada, e pouco ou nada variam para intervalos maiores. Confirma-se mesmo assim que o tempo de convergência é superior para sequências com maior número de DIOs.

Por outro lado, a Figura 6.15, que apresenta o número total de reconvergências da rede, confirma, em parte, as conclusões retiradas da Secção 6.1.2. Para um número mais elevado de DIOs de uma sequência, o número de reconvergências é menor devido ao cálculo mais preciso dos *ranks*. No entanto, os resultados apontam para uma leve melhoria desta precisão quando o número de DIOs é igual a dois para intervalos de tempo τ_h mais elevados.

As Figuras 6.16 e 6.17 apresentam os resultados para o número médio de DIOs emitidos e recebidos, respectivamente. Como seria de esperar, à medida que o intervalo de tempo entre SEQ-DIOs aumenta, menos DIOs são emitidos e consequentemente, menos DIOs são recebidos.

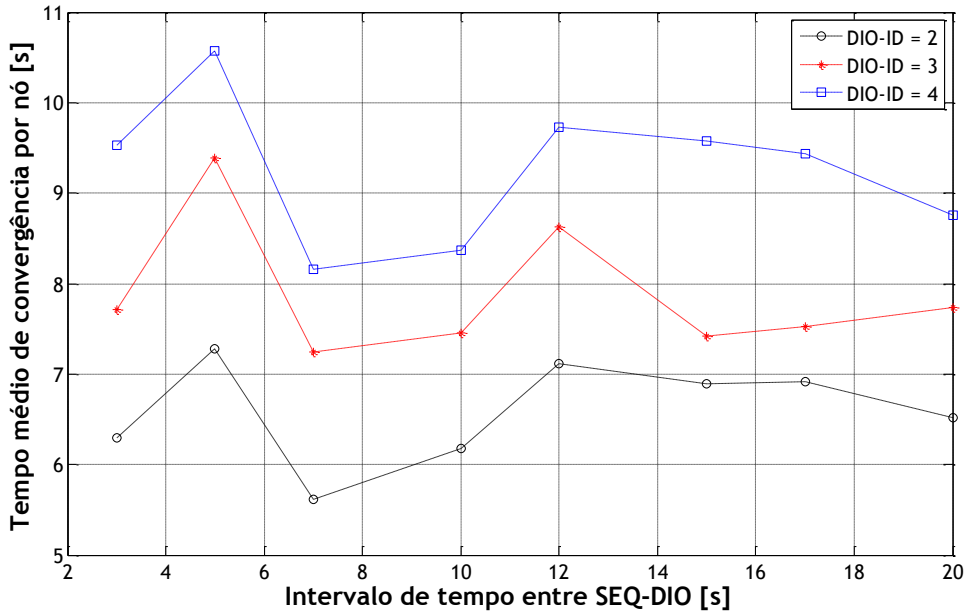

Figura 6.14 - Tempo médio de convergência por nó em função do intervalo de tempo entre sequências SEQ-DIO.

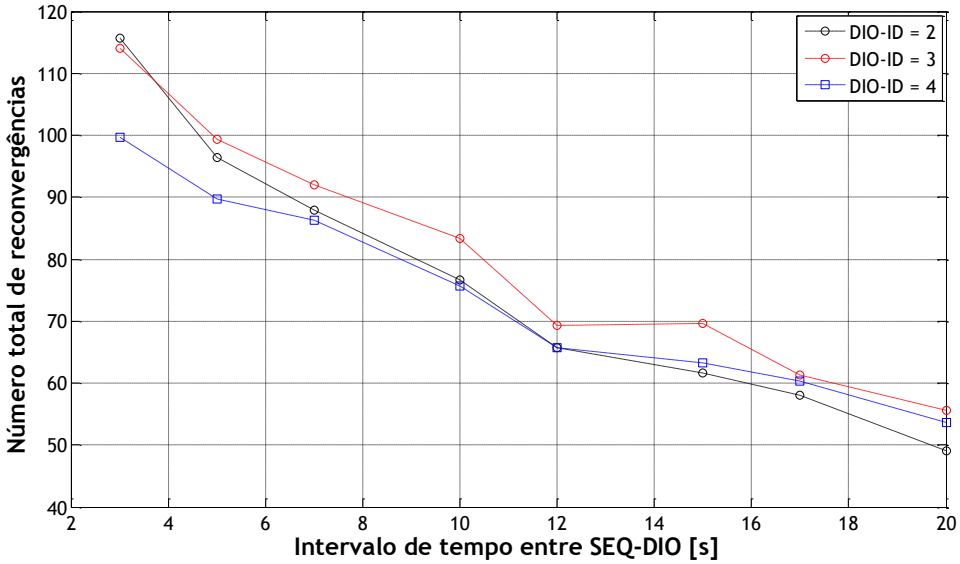

Figura 6.15 – Número total de reconvergências da rede em função do intervalo de tempo entre sequências SEQ-DIO.

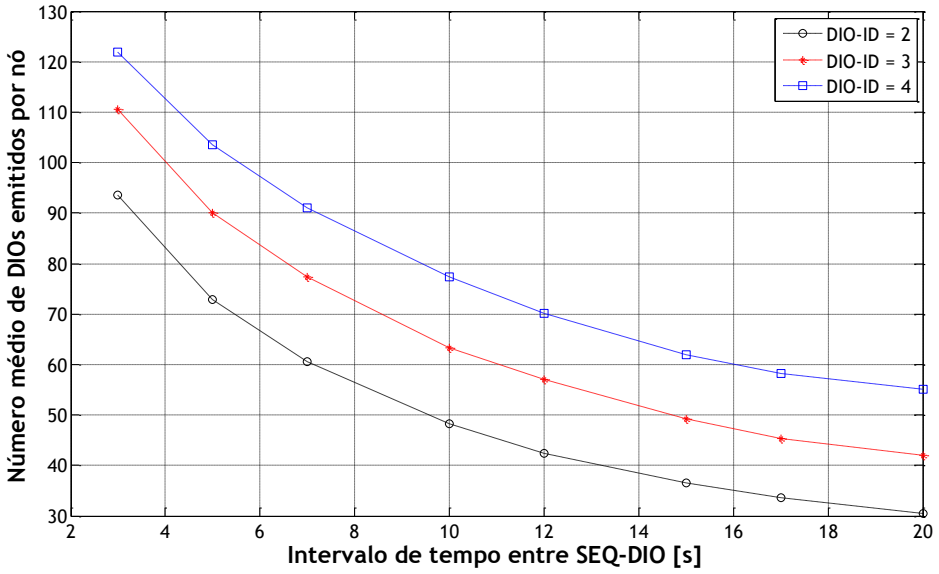

Figura 6.16 – Número de DIOs emitidos por nó em função do intervalo de tempo entre sequências SEQ-DIO.

A Figura 6.18 apresenta os resultados para o *goodput* em percentagem. Estes resultados mostram que a precisão elevada do *rank* e dos encaminhamentos é elevada para as sequências de quatro DIOs. Novamente, devido à troca elevada de DIOs e ao consequente cálculo mais preciso das posições dos nós e escolha dos nós progenitores, os resultados apontam para elevada fiabilidade das comunicações, com elevada taxa de sucesso na entrega de dados e portanto, elevado nível de QoS.

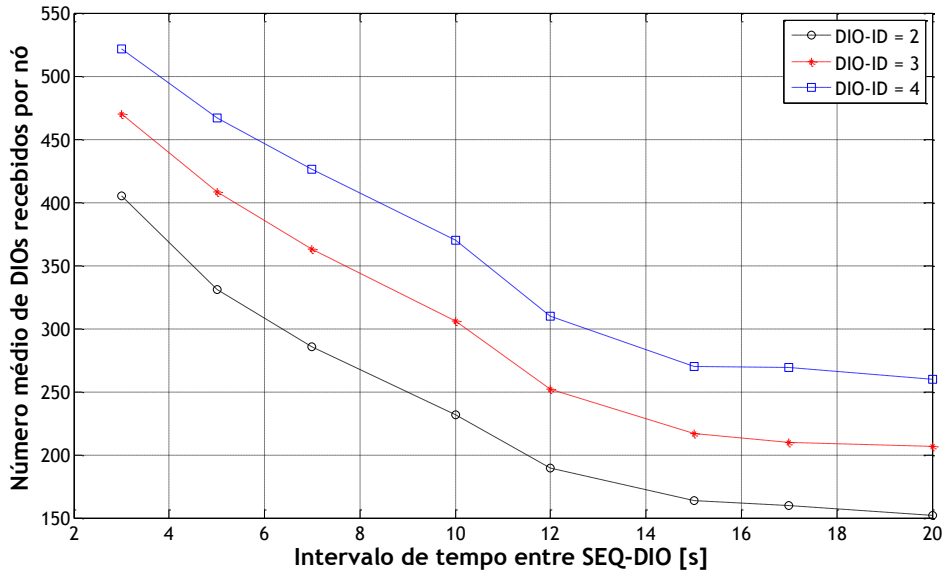

Figura 6.17 – Número médio DIOs recebidos por nó em função do intervalo de tempo entre sequências SEQ-DIO.

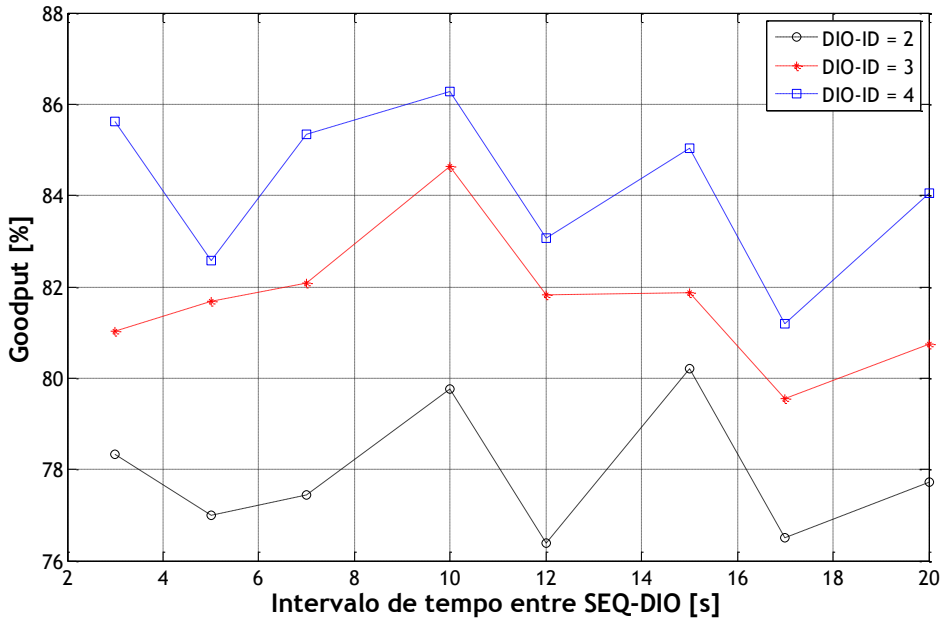

Figura 6.18 – *Goodput* agregado no *nó raiz*, em percentagem, em função do intervalo de tempo entre sequências SEQ-DIO.

6.2 Métrica HOP-COUNT

Como foi referido no Capítulo 5, a métrica HOP-COUNT utiliza apenas o número de saltos entre os nós geradores de dados e o *nó raiz* para calcular o custo do encaminhamento. É uma métrica simples que pretende essencialmente minimizar o número de nós participantes entre a origem e o destino. Os resultados com a métrica HOP-COUNT são comparados com os resultados da métrica ETX-WSN. Para o HOP-COUNT, o intervalo de tempo mínimo foi colocado no intervalo $\tau_l = [0.5, 1]$ segundos enquanto os nós não se encontram "convergidos".

Por outro lado, o tempo máximo é $\tau_h = 15$ segundos quando os nós estão totalmente convergidos. As simulações foram realizadas durante 10 minutos de operação, variando o número de nós presentes e activos na rede.

Como apresentado na Figura 6.19, o HOP-COUNT alcança tempos extremamente reduzidos de convergência. Comparativamente com a métrica ETX, o HOP-COUNT não necessita de partilhar números elevados de pacotes de sonda para calcular o *rank*. Os DIOs são apenas partilhados para informar quais os *ranks* dos nós vizinhos e somar uma unidade por cada salto (*hop*) a esses *ranks*, ao contrário do ETX, que também troca vários DIOs de forma a calcular a qualidade das ligações.

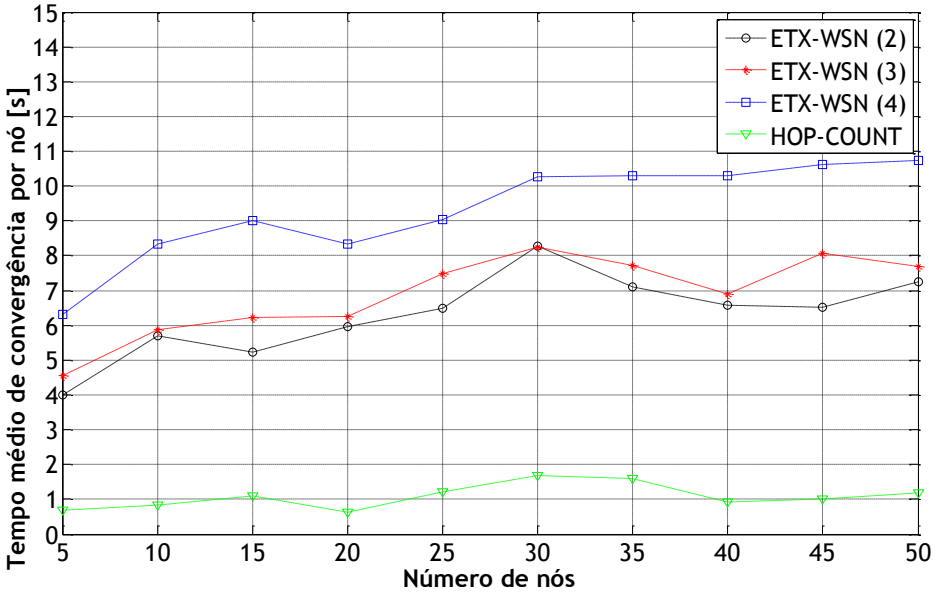

Figura 6.19 – Comparação do tempo de convergência médio em função do número de nós entre HOP-COUNT e ETX.

Na mesma medida, a métrica HOP-COUNT apresenta latências extremamente reduzidas quando comparada com o ETX, como pode ser verificado na Figura 6.20. Desde que dois nós estejam ao alcance um do outro, estes constroem um caminho, não tendo em conta a qualidade dessa ligação. Significa isto que um pacote de dados percorrerá menos nós até chegar ao *nó raiz*, comparativamente com o ETX.

A Figura 6.21 apresenta o número total de reconvergências, comparando o HOP-COUNT com o ETX. Observa-se que ocorre um número de actualizações mais reduzido devido ao facto dos nós não procurarem ligações mais fiáveis ou outros aspectos relevantes. O objectivo da métrica HOPCOUT é procurar e encontrar o nó vizinho que se encontra mais próximo do *nó raiz*.

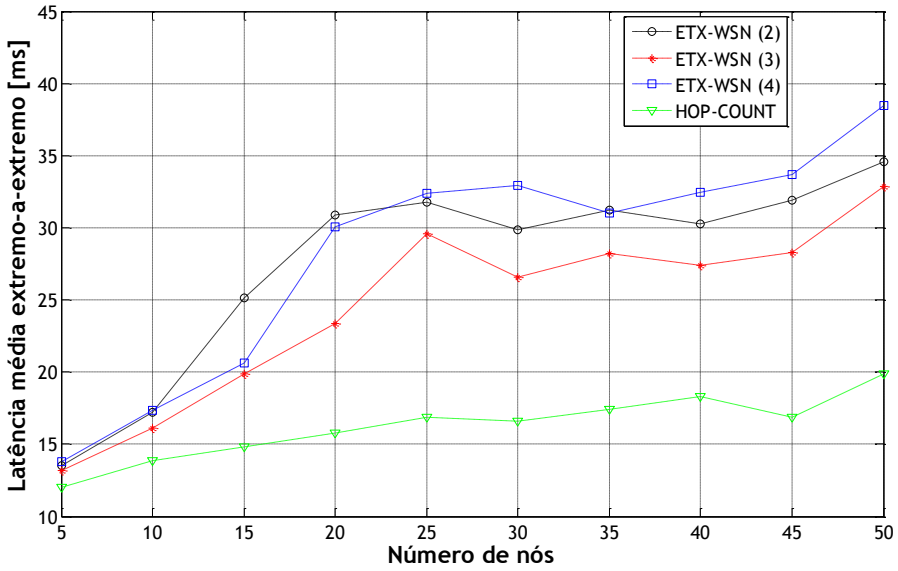

Figura 6.20 – Comparação da latência média extremo-a-extremo para o HOP-COUNT e ETX.

As únicas actualizações realizadas com o HOP-COUNT é precisamente a descoberta de nós mais próximos do nó raiz e que estão no raio de alcance do nó que se auto-actualiza. A Figura 6.22 apresenta a média das reconvergências por nó. Para um número reduzido de nós, o HOP-COUNT verifica muito poucas actualizações, pois os nós mais próximos do nó raiz são imediatamente descobertos. No entanto, à medida que o número de nós aumenta, potenciais nós mais próximos do nó raiz podem ser descobertos ao longo do tempo, pelos nós que se encontram bastante longínquos. Estes resultados para o HOP-COUNT são, no entanto, contraditórios quando comparados com os resultados do ETX. Verifica-se que quantos mais nós existirem na rede, menor (e mais estável) é a média de actualizões por nó com o ETX.

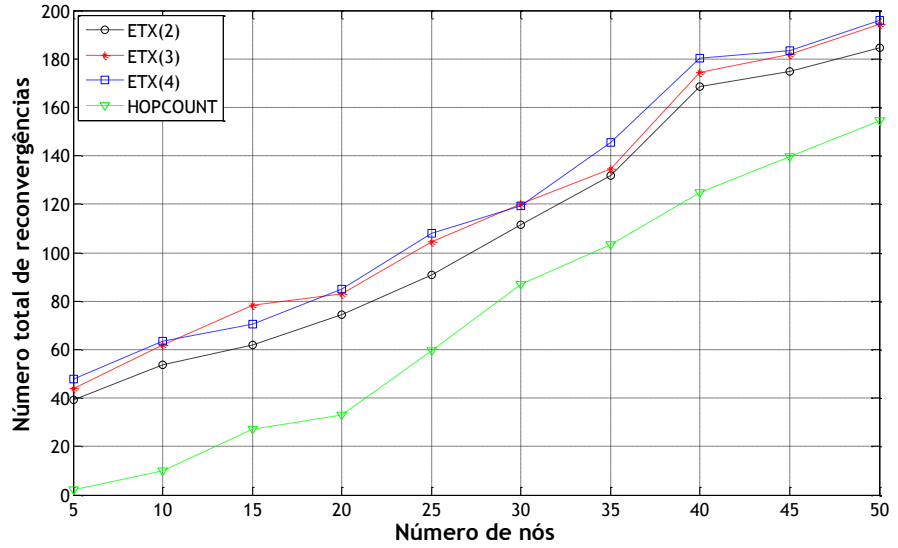

Figura 6.21 – Comparação do número total de reconvergências para o HOP-COUNT e ETX.

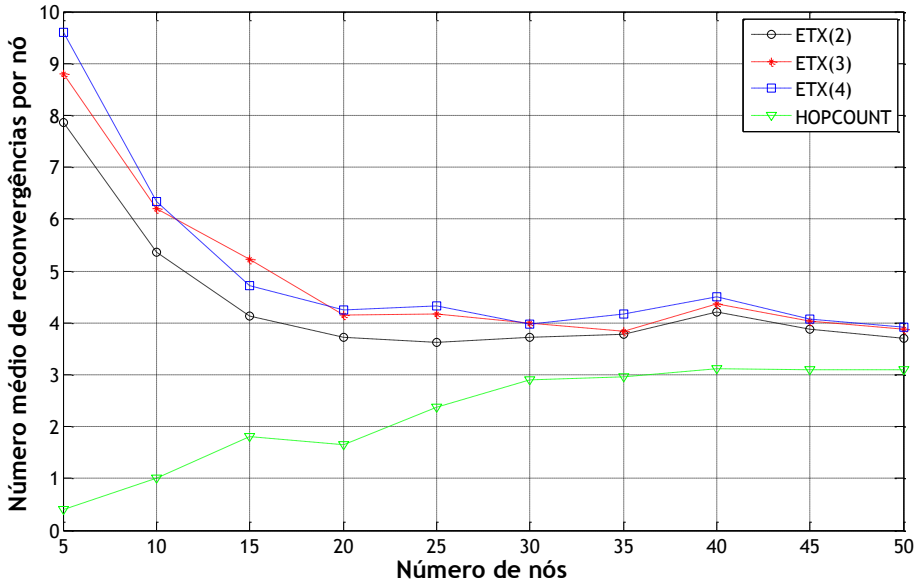

Figura 2.22 – Comparação do número médio de reconvergências em função do número de nós para o HOP-COUNT e ETX.

Por fim, as Figuras 6.23 e 6.24 apresentam os resultados de *goodput*, em percentagem e em bytes, respectivamente. Os resultados apresentam diferenças extremas para o HOP-COUNT e conclusões muito importantes. Primeiro, a fiabilidade das ligações com a métrica HOP-COUNT é extremamente reduzida. O número de pacotes de dados entregues situa-se abaixo dos 50 % quando o número de nós é superior que 40. Em contrapartida a métrica ETX nunca apresentou resultados inferiores a 70 %. Estes resultados são importantes para aplicações onde é mais prioritário considerar o número de pacotes correctamente recebidos no nó raiz do que a própria convergência ou latência da rede. Uma segunda inferência pode ser feita em relação à eficiência energética da rede.

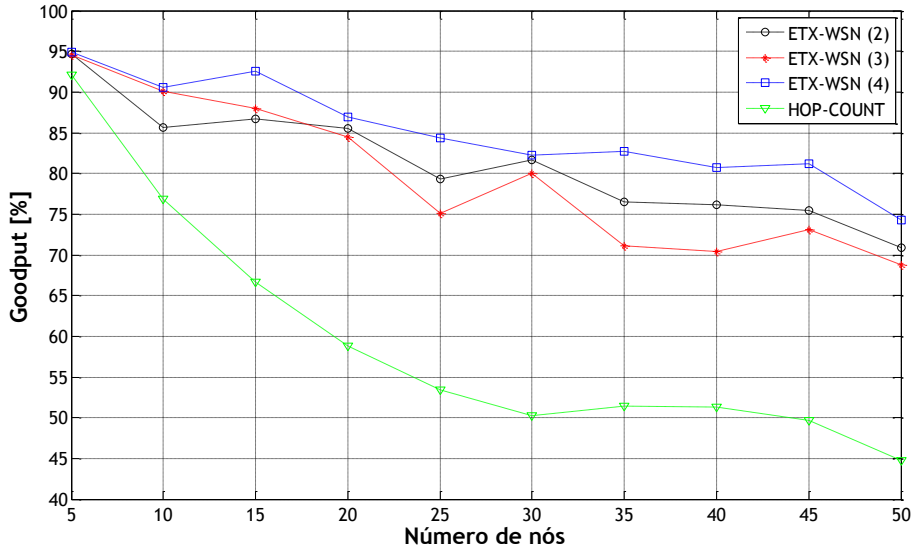

Figura 6.23 – Comparação do *goodput*, em percentagem, para o HOP-COUNT e ETX.

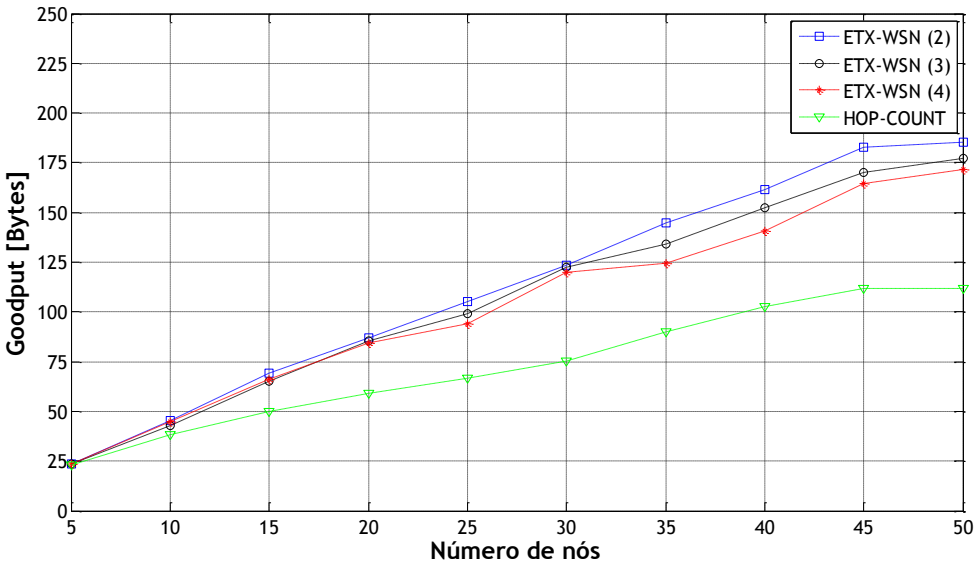

Figura 6.24 – Comparação do *goodput*, em bytes, para o HOP-COUNT e ETX.

Tendo o RPL falhado o encaminhamento de grande parte dos pacotes de dados com a métrica HOP-COUNT, o seu consumo é igual ou maior quando comparado com o ETX. Falhar a emissão/recepção dos pacotes de dados conduz a números mais elevados de retransmissões desses pacotes de dados. As retransmissões provocam maiores *duty cycles* superiores no estado activo e *duty cycles* superiores no estado TX/RX. Se os nós mais afastados do nó raiz falharem sucessivamente a emissão dos dados e/ou a recepção de ACKs mais energia irão desperdiçar. A mesma conclusão pode ser inferida para os nós intermédios.

6.3 Métrica RSSI

O RSSI é uma métrica importante quando se trata de avaliar as condições e qualidade das ligações. Para além das plataformas *hardware* disponibilizarem esse valor instantâneo e médio, podendo este ser armazenado numa tabela com os nós vizinhos associados, o cálculo do *rank* não exige muita complexidade. Para as simulações foi variado o número de nós como realizado nas Secções anteriores e comparados os resultados com o ETX e RSSI.

Foi adoptada uma estratégia base bastante simples para a construção dos encaminhamentos e convergência dos nós com RSSI. Os DIOs partilhados servem de sondas para que os nós calculem o RSSI das ligações com os nós vizinhos. Os DIOs são portanto medidores de qualidade das ligações. Para os encaminhamentos ascendentes se formarem, todos os nós que tenham recebido um DIO com um *rank* maior ou igual a 1 (valor do *rank* do nó raiz), começam também por partilhar os seus DIOs com um *rank* nulo. Os nós que já possuem um *rank* e participem activamente na rede, recebem os DIOs dos vizinhos com *rank* nulo. De forma análoga ao ETX-WSN, os nós activos da rede contam para cada nó vizinho com *rank* nulo pelo menos uma sequência de dois DIOs de cada vez, antes de emitirem um DIO-ACK para um nó vizinho correspondente. O DIO-ACK informa quais os valores médios de RSSI calculados na

recepção dos DIOs de cada um dos nós vizinhos, informando, cada um, qual o RSSI médio que conseguem atingir na ligação para esse determinado candidato a se tornar nó pai. À medida que os DIOs são emitidos, os valores de RSSI vão sendo actualizados entre todos os nós da rede. Após calcularem o *rank* e associarem-se a um nó pai, os nós aguardam DIOs com *rank* nulo para realizarem o mesmo procedimento. Da mesma maneira, sempre que receberem DIOs com *rank* não nulo mas com valores superiores aos seus, realizam o mesmo procedimento de armazenamento dos RSSI e emissão de DIO-ACK. Mesmo após a convergência, os nós progenitores continuam a actualizar os RSSI dos nós mais descendentes e a emitirem os DIO-ACK com essa informação actualizada, permanentemente de dois em dois DIOs (para um determinado nó). Os resultados são apresentados a seguir.

A Figura 6.25 apresenta o tempo médio de convergência por nó. Comparativamente ao ETX, os nós com a métrica RSSI convergem mais depressa. No entanto, a construção inicial dos encaminhamentos é mais lenta do que com o HOP-COUNT.

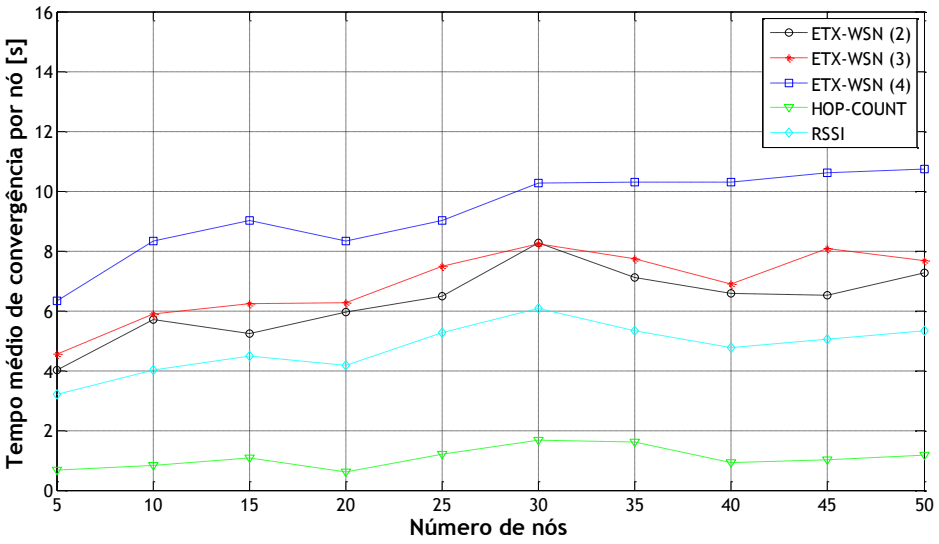

Figura 6.25 – Comparação do tempo médio de convergência em função do número de nós para o RSSI, HOP-COUNT e ETX.

As Figuras 6.26 e 6.27 apresentam o número médio e o total de reconvergências, respectivamente. O RSSI apresenta resultados favoráveis, tanto em relação ao ETX como ao HOP-COUNT. O número de actualizações reduzido mostra a precisão do RSSI a calcular o *rank* e a construir os encaminhamentos. Na prática, com a métrica RSSI, a rede tem um número elevado de actualizações no início, pois os nós vão-se descobrindo mutuamente nos primeiros períodos de operação. Após estarem praticamente todas as descobertas concluídas, os valores de RSSI não se alteram tão frequentemente como se verifica no caso do ETX, e portanto, os *ranks* permanecem basicamente iguais até ao final. Assim conclui-se que, a maior incidência de actualizações ocorre no início, isto é, as sucessivas actualizações por toda a rede acontecem mais rápido e mais cedo do que no caso do ETX.

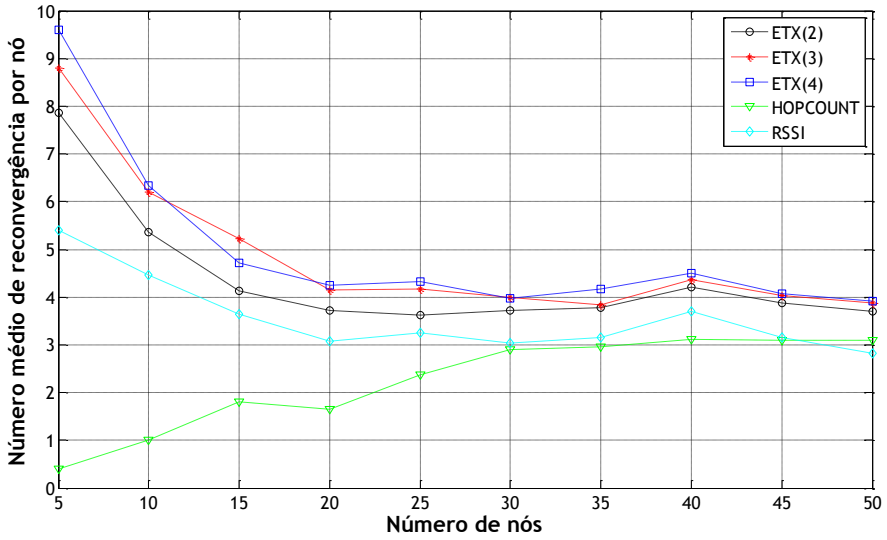

Figura 6.26 – Comparação do tempo médio de reconvergência em função do número de nós para o RSSI, HOP-COUNT e ETX.

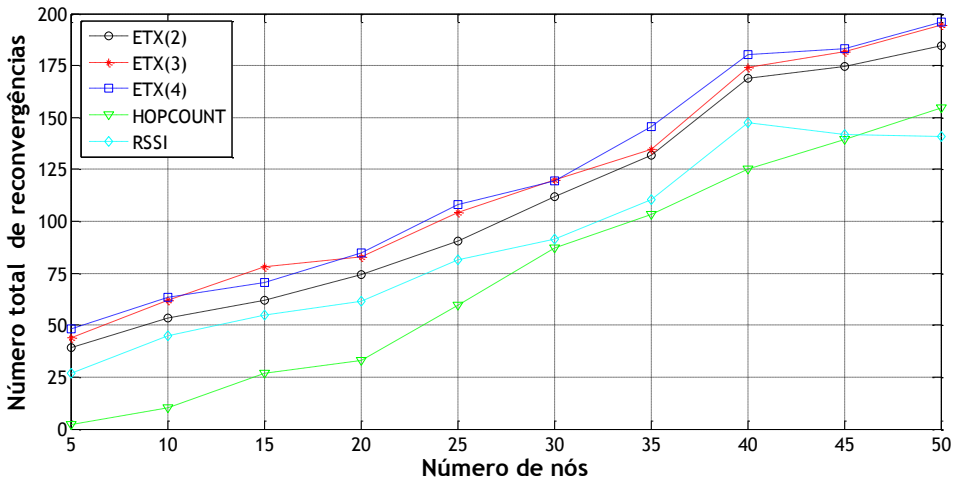

Figura 6.27 – Comparação do tempo total de convergências em função do número de nós para o RSSI, HOP-COUNT e ETX.

A Figura 6.28 apresenta a latência em função do número de nós. O RSSI apresenta valores aceitáveis de latência, menores do que no caso do ETX mas superiores ao do HOP-COUNT.

A partir destes resultados e dos resultados apresentados nas Figuras 6.29 e 6.30 para o *goodput*, podem-se extrair algumas conclusões.

Os resultados do *goodput* com a métrica RSSI são razoáveis. Apesar de serem piores comparativamente ao ETX, apresentam resultados bem mais elevados do que o HOP-COUNT. O RSSI mostra ser uma métrica que consegue "equilibrar" as características necessárias dos encaminhamentos. Quer isto dizer que apresenta equilíbrios adequados entre a latência de convergência, o número de nós participantes no encaminhamento dos pacotes e o *goodput*.

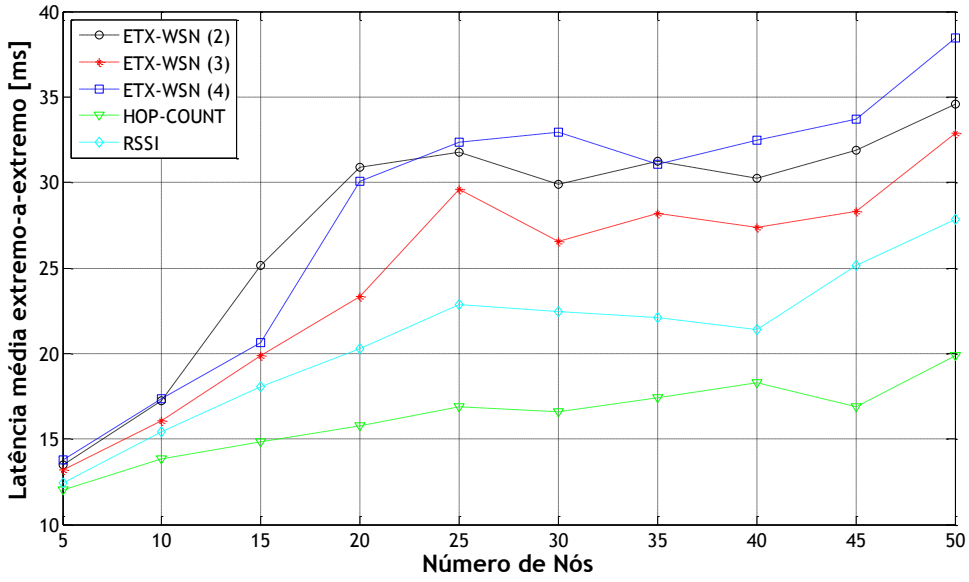

Figura 6.28 – Comparação da latência extremo-a-extremo em função do número de nós para o RSSI, HOP-COUNT e ETX.

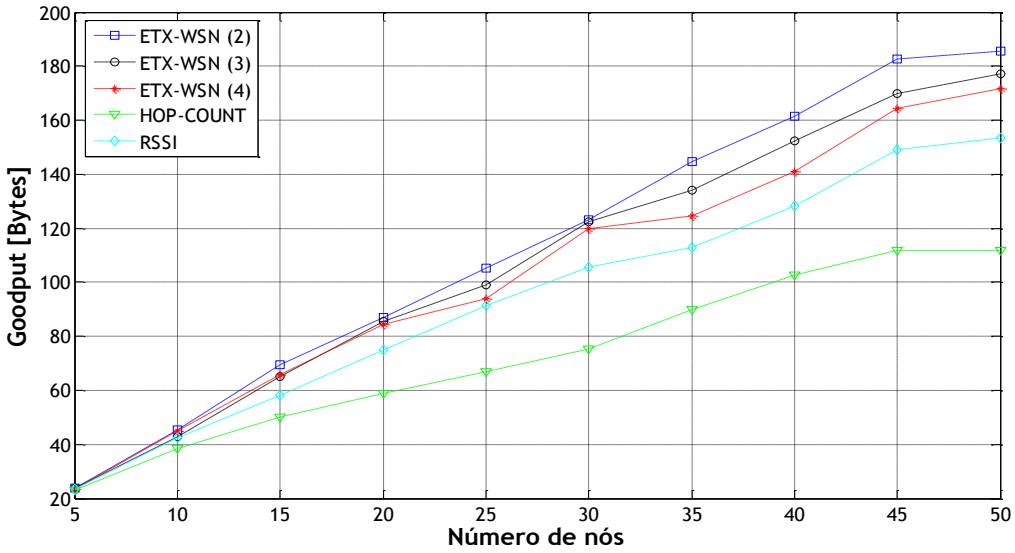

Figura 6.29 – Comparação do *goodput*, em bytes, em função do número de nós para o RSSI, HOP-COUNT e ETX.

Os resultados para a latência e *goodput* mostram que existem ligações mais ou menos razoáveis entre nós geradores de dados e nós progenitores, com um número mínimo possível de nós participantes num dado encaminhamento.

A percentagem de *goodput* alcançada com a métrica RSSI é apenas 10 % inferior do que com a métrica ETX (versão 4 DIOs por sequência). Ainda assim, tem uma latência 20 ms inferior.

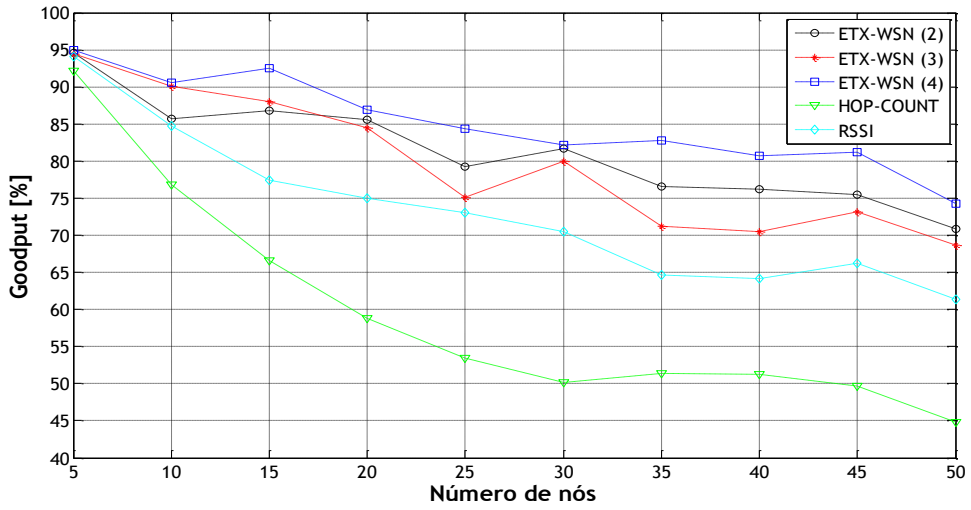

Figura 6.30 – Comparação do *goodput*, em percentagem, em função do número de nós para o RSSI, HOP-COUNT e ETX.

6.3 Sumário e Conclusões

Neste Capítulo apresentaram-se e discutiram-se os resultados de simulação para três métricas principais, de cálculo do custo dos encaminhamentos descendentes para o protocolo RPL, implementado de raiz no simulador OMNeT++. Apresentado em trabalhos anteriores para redes IEEE 802.11, a métrica ETX, tem resultados muito acima da média, com excelentes resultados de *goodput*. Essencialmente, o ETX é uma métrica indicada para aplicações que requerem taxas elevadas de sucesso na entrega de pacotes de dados, bem como fiabilidade elevada.

O HOP-COUNT apresenta os piores resultados, mas consegue com que a rede convirja rapidamente, utilizando o mínimo número de saltos possível, isto é, o número mínimo de nós participantes num dado encaminhamento. Apesar de evitar que muitos nós participem nos encaminhamentos, mantendo-os o mínimo de tempo no estado activo, as sucessivas falhas de entrega de pacotes de dados traduzem-se num maior número de tentativas de envio desses pacotes. Em suma, quanto mais tentativas forem realizadas, mais energia será desperdiçada.

A métrica baseada no RSSI revela ter resultados adequados a todos os níveis. Com o débito binário útil mais ou menos aceitável, apresenta um número reduzido de reconvergências devido ao elevado grau de precisão quando os nós da rede não possuem mobilidade e estão ao longo do tempo fixos no mesmo local e mantêm a mesma distância entre os nós vizinhos. A métrica possui melhor eficiência energética que o ETX pois não existe uma troca tão acentuada de DIOs como pacotes de sonda. Por outro lado, não tem um nível de perdas de pacotes de dados tão acentuado como no caso do HOP-COUNT.

Capítulo 7

Conclusões e Sugestões de Trabalho Futuro

7.1 Conclusões

O século XXI é palco de uma progressiva revolução tecnológica. A rápida evolução das tecnologias de informação e comunicação permite que surjam novas aplicações relacionadas com os Ambientes Inteligentes (AmI), oferecendo novos paradigmas de partilha de informação útil entre entidades físicas (por exemplo, objectos) e as pessoas. Os AmI surgem a partir da convergência de sistemas de microelectrónica com tecnologias de redes de comunicação sem fios.

As tecnologias de comunicação são o grande suporte dos AmI. A computação ubíqua só é possível através das redes de comunicação que possibilitam a partilha de informação a qualquer momento e em qualquer lugar. Os AmI utilizam um vasto potencial de tecnologias, que juntas, formam redes heterogéneas interligadas, desde redes de potência reduzida embutidas nos ambientes e nos objectos, até às redes celulares que partilham e disponibilizam toda a informação, dos ambientes para as pessoas, com elevados ritmos de transmissão e elevada capacidade, a partir de *data centers*.

A *Internet of Things* (IoT) é a uma estrutura baseada na Internet tradicional que identifica objectos e entidades físicas de forma única e global, com representação virtual na web. A "nova Internet" não se refere unicamente à Internet das coisas e dos objectos, mas sim, à junção desta com a Internet tradicional das pessoas, alcançando novos paradigmas de partilha de informação nas sociedades humanas. Esta aproximação entre serviços virtuais e físicos possibilita o aparecimento e evolução dos AmI, sendo toda a informação proveniente dos objectos dos ambientes inteligentes envolventes, partilhada de forma transparente, através de aplicações e novos serviços disponíveis para as pessoas.

A potencialidade das redes de comunicação heterogéneas nos AmI, utilizada pela IoT, não se resume apenas à partilha de informação e serviços entre as tecnologias embutidas nos ambientes e as pessoas. Este novo paradigma possibilita a interligação dos próprios objectos e dispositivos inteligentes, para que sem qualquer intervenção humana, estas entidades físicas possam partilhar os seus dados e responder a determinados comportamentos e acontecimentos dentro dos ambientes e dentro do contexto no qual estão inseridos. As comunicações *Machine-to-Machine* (M2M) completam todo o cenário criado pela IoT e as redes de comunicação das AmI, com aplicações de automação e monitorização em cidades, casas, indústria, saúde e agricultura.

Nesta dissertação, no Capítulo 2, são descritas as várias tecnologias utilizadas na IoT e M2M. A IoT reutiliza os protocolos da Internet, e a estrutura da web, para disponibilizar os seus serviços. As comunicações M2M reutilizam as estruturas e tecnologias das redes de comunicação. Vários projectos estão centrados neste conceito de IoT-M2M, e algumas entidades como o ETSI e o IEEE, propõe normas, recomendações e métodos para que os vários desafios que se colocam sejam ultrapassados. Ao nível da capacidade das comunicações, prevê-se que os AmI sejam cada vez mais uma fonte exponencial de dados. A geração de informação de milhares (ou milhões) de dispositivos embutidos nos ambientes, representa um acréscimo brutal de quantidade de dados que as redes conseguem suportar. Apesar da maioria destes dados ocuparem muito pouco espaço em memória, com comunicações bastante rápidas, espera-se que sejam em grande número. Por esta razão, é essencial que as tecnologias de comunicação implementadas tenham capacidades adicionais para manterem os serviços actuais e os da IoT.

As redes de comunicação M2M dividem-se em duas sub-redes: celulares e capilares. As redes capilares são as redes de comunicação inseridas nos ambientes que possibilitam a comunicação dos dados produzidos para pontos de acesso próximos. As redes celulares são a espinha dorsal destas várias redes capilares que suportam os ambientes inteligentes. A ubiquidade e omnipresença da informação é possível a partir das redes celulares. Por outro lado, devido aos consumos reduzidos e capacidade de auto-organização, as tecnologias das redes capilares possibilitam a formação de pequenos grupos de dispositivos embutidos nos objectos e nos ambientes.

No Capítulo 3, propõe-se a adopção das redes de sensores sem fios (RSSF) para as redes capilares. As redes de sensores sem fios têm características muito próprias, facilmente adaptadas aos ambientes inteligentes. São redes que nos últimos anos têm vindo a ganhar bastante interesse, devido ao grande leque de aplicações inteligentes que suporta, e oportunidades de negócio aliciantes. Classificam-se como redes de dispositivos de consumo e potência reduzida, constituídos por sensores que recolhem dados dos ambientes e partilham os mesmos, através de comunicações bastante simples. As suas características possibilitam consumos reduzidos, um factor determinante para redes com um número gigantesco de dispositivos permanentemente activos. Consequentemente, os dispositivos em redes de sensores sem fios têm capacidades reduzidas em termos de processamento, mas suficientes para realizarem as suas funções. Além disso, são dispositivos com tamanhos bastante reduzidos (conceito *smart dust*), facilmente implementáveis nos ambientes. Estes dispositivos têm portanto, custo muito reduzidos, tanto de *hardware*, como de implementação e desenvolvimento. Todas estas características traduzem-se em vantagens importantes na adopção das redes de sensores sem fios para as redes capilares, integradas nos ambientes e nos objectos inteligentes.

A norma IEEE 802.15.4 especifica as camadas físicas e MAC das redes de sensores sem fios. É uma norma com especial foco em comunicações com custos reduzidos, ritmos de transmissão reduzidos e são redes com pouca ou nenhuma infraestrutura subjacente, alcançando a máxima eficiência energética possível. Mantendo o mínimo de recursos energéticos e espectrais necessários à realização das comunicações, introduz técnicas para maximizar a eficiência da partilha de dados com métodos *beacon-enable* e *nonbeacon*, transição de estados activos para estados "adormecidos" dos componentes electrónicos dos dispositivos para minimizar o consumo de energia, tempos de emissão/recepção extremamente reduzidos e topologias de rede simples.

O vasto número de trabalhos relacionados com a norma IEEE 802.15.4 propõe protocolos que maximizem a eficiência energética e o tempo de vida dos dispositivos de potência reduzida, sem reduzir a fiabilidade e a eficiência das próprias comunicações. A optimização das redes de sensores sem fios através destes protocolos é possível através de paradigmas de *cross-layer*, protocolos colaborativos, auto-organização e métodos de acesso ao meio híbridos.

No Capítulo 4 foram selecionadas algumas plataformas *hardware* de dispositivos utilizados nas redes de sensores sem fios e disponíveis no mercado. O estudo realizado pretendeu identificar quais as plataformas mais eficientes ao nível do consumo de energia, baseado nos tempos de comunicação e transição de estados da norma IEEE 802.15.4. A plataforma WirelessHart mostrou ter o tempo de vida superior devido aos seus consumos de energia extremamente reduzidos, comparada com as outras plataformas. Algumas plataformas com consumos de energia idênticos ao WirelessHart (por exemplo, o Waspmote) conseguem atingir tempos de vida elevados, mesmo sem recorrerem a qualquer tipo de recolha de energia do ambiente. Por outro lado, o SunSpot e o Imote2 provam ter os tempos de vida mais elevados, com consumos muito acima da média. Contudo, a frequência de operação do microcontrolador do SunSpot e Imote2 é bastante mais elevada em comparação à maioria plataformas, incluindo o WirelessHart. Esta capacidade de processamento acima da média possibilita o desenvolvimento de aplicações de voz e vídeo. A eficiência energética nestas aplicações continua a ser um desafio a ultrapassar.

A interligação das redes de sensores sem fios com redes externas, nomeadamente a Internet, é possível através do protocolo IP. A atribuição dos endereços IP a estes dispositivos de potência reduzida é um passo fulcral para a convergência das redes capilares com as redes celulares. Esta atribuição permite identificar os dispositivos/objectos como entidades únicas na rede, e portanto, facilmente identificáveis e localizáveis como os serviços virtuais. Devido à integração de milhares de dispositivos em todo o mundo, a solução mais viável centra-se na utilização do IPv6. A camada de adaptação 6LoWPAN permite atribuir endereços IPv6 aos dipositivos das redes de potência reduzida (como as redes de sensores sem fios) com um *overhead* muito reduzido dos cabeçalhos IPv6. O 6LoWPAN utiliza métodos de compressão do

cabeçalho IPv6, capazes de reduzir mais de 80 % do seu comprimento, com atribuições *stateless* ou *stateful*, técnicas de fragmentação e encaminhamento tanto local como global.

A camada de rede que oferece suporte à camada de adaptação 6LoWPAN utiliza o protocolo de rede RPL. O RPL é um protocolo da categoria hierárquica, com uma construção topológica em árvore, especificado directamente para as redes de sensores sem fios. São descritos três tipos de pacotes de controlo. O DIO constrói encaminhamentos ascendentes, o DAO constrói encaminhamentos descendentes e o DIS solicita informação para os nós se juntarem à rede. O intervalo de tempo entre a emissão destes pacotes de controlo é calculado através do algoritmo de propagação de código, *Trickle*. O RPL utiliza um processo de construção da topologia e dos encaminhamentos, idêntico ao utilizando pelo protocolo CTP, actualmente desenvolvido em aplicações reais.

Foram identificados alguns problemas relacionados com a manutenção dos encaminhamentos e detecção de falhas após a convergência de toda a rede. Nesse âmbito, foram também identificadas algumas propostas para a resolução desses problemas.

A construção e manutenção dos encaminhamentos são processos realizados mediante custos associados à posição dos nós em relação ao nó central. Estes custos, calculados através de métricas relacionadas com as dinâmicas da rede, são partilhados entre os nós de maneira a estes construírem as suas tabelas de topologia e de encaminhamento, e conhecerem a posição e função que ocupam nos encaminhamentos da rede. Ao longo do tempo, estas métricas podem alterar os seus valores devido a vários factores tal como mobilidade, perdas de ligação e estados prolongados de inactividade da parte dos nós vizinhos. Devido à importância das métricas na construção e manutenção dos encaminhamentos, foram descritas algumas métricas propostas para as redes de sensores sem fios e adaptadas ao protocolo RPL. Após avaliadas as características de cada métrica sugeriram-se para cada uma, funções objectivo que calculam o custo das ligações entre os nós. Em RPL, este custo é denominado *rank*.

Os comportamentos e os desempenhos de cada métrica foram avaliados, após o desenvolvimento e implementação do protocolo RPL no simulador OMNeT++ em conjunto com a *Framework* MiXiM. Para as métricas ETX, HOP-COUNT e RSSI, foram criados vários cenários variando aleatoriamente o número de nós na rede com parâmetros de tráfego e tempo iguais. As simulações foram realizadas no âmbito das comunicações locais 6LoWPAN, sem fragmentação de pacotes (sem exceder o comprimento máximo de 127 bytes) e apenas com encaminhamento do tipo *mesh*. Sendo a métrica ETX, uma métrica proposta para redes Wi-Fi, foi proposto um novo algoritmo ETX adaptado às características das RSSF. Os resultados provam que a métrica ETX apresenta os melhores resultados de débito binário útil, com taxas de entrega de pacotes com sucesso entre 95 % e 80 %. O RSSI mantem resultados aceitáveis ao longo das simulações, mas o HOP-COUNT tem um decréscimo bastante acentuado de débito binário útil à medida que o número de nós na rede aumenta. Ao contrário da métrica ETX,

que procura estabelecer ligações mais fiáveis para construir os encaminhamentos, o HOP-COUNT mantém um número mínimo de nós a participar num dado encaminhamento, por vezes com distâncias demasiado longas entre cada salto, o que se traduz na perda de mais de metade dos pacotes de dados enviados quando o número de nós numa rede é maior que 45 nós. A desvantagem do ETX consiste essencialmente na necessidade de existirem transmissões periódicas de sequências de pacotes de sonda para construir os encaminhamentos. A taxa de entrega destes pacotes de sonda é o indicador de qualidade das ligações entre um par de nós. O algoritmo ETX-WSN, proposto para as RSSF, prova calcular custos e construir encaminhamentos que correspondem às expectativas adquiridas na análise e nos resultados para redes Wi-Fi, em trabalhos anteriores.

7.2 Sugestões de Trabalho Futuro

Para trabalho futuro propõe-se o desenvolvimento de uma métrica que contabilize a energia consumida pelos nós ao longo do tempo, ou a energia residual das baterias de cada nó. Alguns protocolos de encaminhamento *energy-aware* mostram obter bons resultados relacionados com o tempo de vida da rede. As métricas baseadas em valores de energia têm capacidade para equilibrar o consumo de energia entre os nós, substituindo à vez, os nós participantes num encaminhamento ao longo do tempo, consoante a energia residual das baterias.

Outra proposta é a fusão de duas ou mais métricas numa só função objectivo e posterior simulação e obtenção de resultados. Por exemplo, a métrica ETX de duas sequências (SEQ-DIO = 2) juntamente com a métrica RSSI pode ajudar a diminuir a latência extremo-a-extremo e o tempo de convergência, ao mesmo tempo que, com um número aceitável de pacotes de sonda partilhados, pode manter ou aumentar o débito binário útil. Por um lado, o ETX selecciona as ligações mais fiáveis em termos de débito binário, por outro lado, o RSSI ajuda a escolher de um conjunto de ligações fiáveis, qual a que apresenta melhor qualidade do sinal. Outro exemplo, é a junção do ETX com uma métrica do tipo *energy-aware*. A partir de um equilíbrio entre ligações fiáveis e energia residual em cada nó, encaminhamentos energeticamente eficientes e equilibrados podem, mesmo assim, obter resultados excepcionais em termos de débito binário útil.

Um outro estudo importante relacionado com o protocolo RPL é o seu comportamento face a cenários de mobilidade elevada, isto é, cenários onde os nós estão em movimento, com trajectos aleatórios ou constantes. Muitas aplicações requerem que os nós não estejam fixos num só local, mas que se movam livremente. A mobilidade é uma característica que provoca profundas dinâmicas e alterações do estado dos nós e das ligações entre estes. Portanto, é importante verificar quais as métricas mais eficazes neste tipo de cenários, e que estratégias de manutenção e resolução de problemas de encaminhamento melhor se adaptam a RSSF móveis.

Relativamente ao desenvolvimento da camada de adaptação 6LoWPAN, nesta dissertação foi apenas considerado o encapsulamento para comunicações locais, isto é, encaminhamento *mesh*. Na mesma medida, foram só consideradas transmissões de tramas de dados que não excediam 127 bytes de comprimento, portanto, não foi considerada a fragmentação de pacotes. Sugere-se para trabalho futuro, simulações que incluam compressão *stateful*, fragmentação de pacotes e encaminhamento *route*. É importante avaliar o desempenho do protocolo RPL quando estão duas ou mais sub-redes interligadas com redes externas heterogéneas, através de vários nós raíz.

Apesar do relevo e conclusões que as simulações em tempo real oferecem, é imperativo que estes resultados sejam apresentados em aplicações reais. Portanto, sugere-se como trabalho futuro o desenvolvimento e implementação do protocolo RPL com estas métricas em plataformas *hardware*, e posterior construção de redes reais. Actualmente, os sistemas operativos TinyOS e ContikiOS têm soluções reais baseadas no protocolo RPL. Portanto, a aplicação destas métricas nestes sistemas operativos é uma forte contribuição para o aperfeiçoamento de resultados obtidos em aplicações reais.

Ao nível das comunicações M2M, são necessárias técnicas mais eficientes para a comunicação e partilha de dados provenientes das redes capilares em rajada. Existem várias propostas neste sentido, mas continuam a faltar desenvolvimentos reais, para a obtenção e análise de resultados concretos.

Anexo A

Simulador OMNeT++

O OMNeT++ é um simulador de eventos discretos para redes de comunicação. É um simulador de código aberto, baseado em módulos e orientado a objectos, construído na linguagem de programação C++. Na prática, o OMNeT++ não é um simulador de algo em concreto. O OMNeT++ fornece sim, uma estrutura e um conjunto de ferramentas genéricas e flexíveis, que possibilitam o desenvolvimento de simulações em várias áreas para além das redes de comunicação, tal como simulação de processos de negócio e sistemas complexos de IT.

A infraestrutura do OMNeT++ é composta por uma arquitectura hierárquica baseada em componentes. Estes componentes, denominados "módulos", podem ser reutilizados e combinados de maneira a construir modelos de simulação. Estes módulos são referidos como redes. O módulo *top-level* é o *system module*, ou módulo de rede (*network module*). Este módulo superior pode conter vários sub-módulos, e estes por sua vez, podem conter mais sub-módulos. Módulos que contenham sub-módulos são denominados módulos compostos. Por sua vez, módulos que não contenham sub-módulos chamam-se módulos simples. No modelo hierárquico, os módulos simples situam-se nas camadas mais inferiores e estão directamente associados a um ficheiro C++ que fornece os comportamentos desejados da rede. Os módulos compostos agregam vários módulos simples. A Figura A.1 mostra como os sub-módulos são agrupados nos módulos compostos.

Os módulos comunicam através de mensagens. Cada mensagem pode conter dados arbitrários relacionados com os parâmetros dos módulos. Estas mensagens são partilhadas através de interfaces denominadas portais de entrada e saída (*input/output gates*), ou enviadas directamente para os módulos simples, baseado nas identificações (IDs) únicas dos módulos. Quando um módulo recebe uma mensagem de outro módulo, ou de si mesmo, ocorre um evento. Numa rede, as mensagens representam pacotes ou tramas. As *auto-mensagens* são mensagens geradas num módulo e enviadas para si mesmo para calendarizar o disparo de eventos num intervalo de tempo interno e definido no próprio módulo.

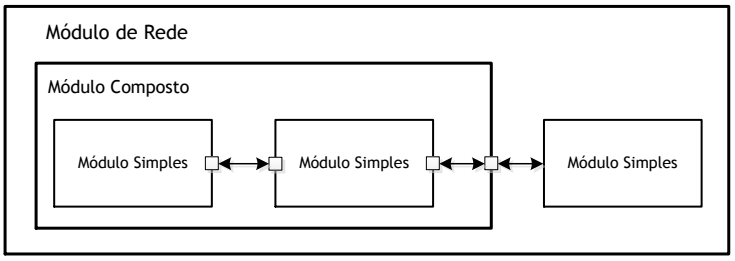

Figura A.1 – Módulos compostos e simples.

Os modelos de simulação são estruturados pelos módulos e interligações. O OMNeT++ utiliza uma linguagem descritiva denominada *Network Description* (NED) para descrever a estrutura dos modelos. Parâmetros globais de todos os módulos são especificados num ficheiro NED, possibilitando a passagem de dados de conFiguração para módulos simples.

Além de usarem as suas próprias funções, os módulos simples herdam as funções e os parâmetros dos módulos compostos a que pertencem. O envio, tratamento e recepção de mensagens são exemplos de funções implementáveis em módulos simples e herdadas do módulo composto *cSimpleModule*.

A escolha do simulador OMNeT++ para esta dissertação deveu-se a vários factores:

- Simulador de código aberto;
- Informação disponível em documentos e comunidades de utilizadore;
- Simulações paralelas;
- Interpretação e análise de resultados;
- Controlo sobre a execução das simulações;
- Debugging;
- Extremamente bem estruturado;
- Implementações realistas de cenários, modelos de propagação e redes sem fios.

Anexo B

Framework MiXiM

O MiXiM é uma *framework* de modelação para o OMNeT++ criada para simulações de redes móveis, redes de sensores sem fios, redes de área corporal, redes Ad-Hoc, Wi-Fi, entre outras. O MiXiM oferece um conjunto de módulos que descrevem detalhadamente modelos de propagação, estimativas de interfência, consumos de energia de rádios transmissores/receptores de nós RSSF e protocolos da camada MAC.

Como parte integral do OMNeT++, o MiXiM tem uma estrutura hierárquica com vários módulos básicos que proporcionam a criação de módulos simples.

O protocolo RPL foi desenvolvido e implementado dentro da *Framework* MiXiM, utilizando módulos descritores das características das redes de sensores sem fios, e da camada Física e MAC dos nós.

Uma rede MiXiM é formada por três componentes:

- **Rede** (*BaseNetwork*) – contem a rede de simulação;
- **Nó** (*BaseNode*) - contem o módulo composto que define o tipo de nós na rede*;*
- **NIC** (*BaseNic*) – contem o módulo composto que define a placa de rede utilizada pelos nós.

O MiXiM tem definidos vários tipos de nós, incluindo para redes de sensores sem fios. Os nós são definidos pela camada física e MAC, como apresentado na Figura B.1. A camada física é um módulo composto que define como são enviadas e recebidas as mensagens, detectadas colisões e calculados os erros binários das transmissões. Os vários módulos simples existentes no módulo da camada física são apresentados na Figura B.2 e descritos como:

- **Sinal** – A classe "sinal" agrega uma representação das características de um sinal como a potência de emissão, atenuação e ritmo de transmissão, no tempo, na frequência e no espaço. O nó receptor adiciona atenuação e calcula os erros binários segundo a representação do sinal recebido;
- **Modelos analógicos** – A atenuação do sinal é calculada simulando características reais de *path loss*, *shadowing* e *fading*. Existem vários modelos de propagação desenvolvidos e que podem ser implementados na camada física, consoante as características dos cenários e ambientes que se pretendem simular;
- ***Decider*** – O *decider* classifica as mensagens recebidas como tramas de dados ou simplesmente ruído. Esta decisão é suportada por modelos que determinam quando

um sinal é classificado como ruído. Após decidir que recebeu um trama de dados, o *decider* calcula os erros binários da mensagem calculando a relação sinal/ruído (SNR) e, posteriormente, a taxa de erros binários (BER). Este calculado leva a conclusões simples, tal como, se a trama foi recebida com sucesso ou não. Por último, o *decider* fornece informação sobre o estado do canal de comunicação à camada MAC para ser utilizada em protocolos e métodos de acesso ao meio;

- ***BasePhyLayer*** – Este módulo actua como uma interface entre as mensagens transmitidas no meio de comunicação (*AirFrames*) e o modelo analógico e o *decider*. Assim, é possível interligar diferentes modelos analógicos com diferentes tipos de *deciders*, com máxima flexibilidade. Após a recepção da mensagem, o *BasePhyLayer*, passa a mensagem para o modelo analógico (para que seja calculada a atenuação do sinal) e simula o atraso de propagação e de transmissão da mensagem. Depois da mensagem passar pelo *decider* e calculados os erros binários é decidido ou não enviar a mensagem para a camada MAC.

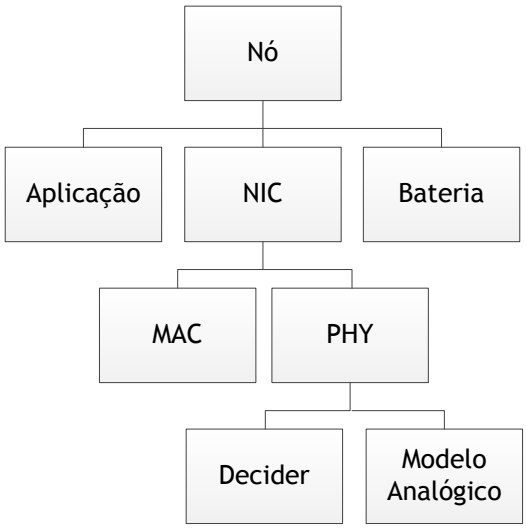

Figura B.1 – Modelo hierárquico de um nó.

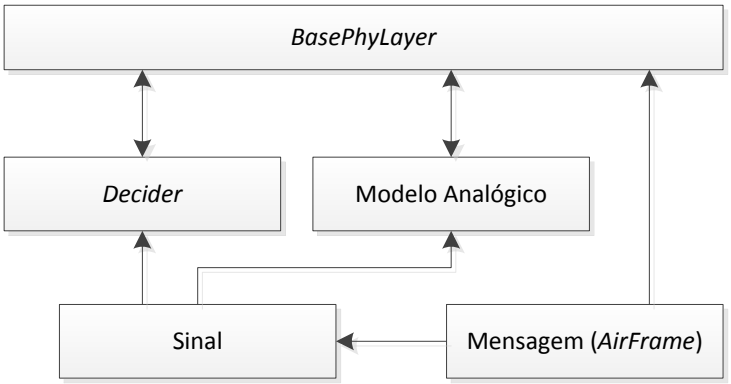

Figura B.2 – Módulos da camada física.

Os módulos da camada MAC realizam o encapsulamento e desencapsulamento de pacotes, passam a mensagem aos módulos das camadas superiores, definem os endereços físicos dos nós e são utilizados para implementar protocolos MAC. Os protocolos MAC utilizam parâmetros e valores adquiridos dos módulos que compõe a camada física para decidir sobre vários comportamentos dos nós. Por exemplo, a emissão de tramas ACK, *beacons*, transições de estado (entre o estado activo e o estado SLEEP), comprimento da janela de contenção (CW), número de tentativas de emissão de uma trama, entre outros. As camadas superiores recebem a mensagem através da camada MAC. O MiXiM oferece módulos desenvolvidos para a camada MAC, de rede, transporte e aplicação. No entanto, esta abstração e hierarquia entre os módulos permite o desenvolvimento de protocolos MAC e de encaminhamento, independentemente dos modelos utilizados nos módulos da camada física. O MiXiM é, portanto, uma forte ferramenta para a simulação em tempo real de todo o tipo de protocolos.

Anexo C

Desenvolvimento no simulador

O RPL foi desenvolvido para funcionar como um módulo da camada de rede. A camada física utilizada nas simulações é composta por módulos que implementam as características da norma IEEE 802.15.4. O ficheiro NED utilizado que compõe o tipo de nó da rede é o *Host802154_2400MHz*. Este nó é composto por um NIC que simula as características do rádio emissor/receptor CC2420 da Texas Instruments, cujo módulo denomina-se *Nic802154_TI_CC2420*. Este módulo especifica correntes de consumo, tempos de transição de estado, potência de emissão, ruído térmico e o método de acesso ao meio implementado pela camada MAC. Os valores escolhidos para este parâmetros são os de origem e encontram-se no Anexo C.1. Este módulo especifica a utilização da técnica de modulação OQPSK da parte do *decider*.

O módulo *CSMA802154* simula o acesso ao meio por CSMA como especificado na norma IEEE 802.15.4 e é apresentado no Anexo C.2. O método de *back-off* utilizado denomina-se "*exponential*" e está desenvolvido de origem no ficheiro C++ "csma.cc". O *exponential back-off* simula o aumento exponencial do comprimento da janela de contenção à medida que são realizadas tentativas para transmitir mensagens para outros nós.

O modelo analógico utilizado para calcular a atenuação do sinal é o *BreakpointPathlossModel*, com um limite mínimo de atenuação igual a -100 dBm e um coeficiente alfa iguala a 2. A sensibilidade da recepção do sinal na camada física é igual a -92 dBm.

São definidas quatro tipos de mensagens na camada de rede para o protocolo RPL: DIO, DIOACK, DATA e DIS. A mensagem DIO corresponde aos pacotes de controlo DIO para partilha de informação sobre *ranks*. No caso do ETX, o número de DIOs transmitidos com sucesso traduz-se no cálculo do *rank* para cada nó. Cada DIO enviado tem associado um ID (nbDIO) e a sequência na qual o respectivo DIO faz parte (seqDIO). Após o a recepção de um número fixo de DIOs duma sequência, o nó receptor emite um DIOACK que informa quantos DIOs numa dada sequência recebeu (Anexo C.3). Assim, o nó calcula o ETX associado a esse nó vizinho (Anexo C.4), e utiliza esse resultado para calcular o *rank* e seleccionar um nó vizinho.

Ainda no caso do ETX, são construídas quatro tabelas diferentes:

- ***nbDIOTable*** – Armazena o ID do DIO recebido referente a um determinado nó vizinho. Esta tabela é sempre reinicializada para um nó vizinho quando um DIO com o ID máximo (último DIO de uma sequência) é recebido;

- ***seqDIOTable*** – Armazena o número da sequência do DIO recebido referente a um determinado nó vizinho. Sempre que um nó recebe o último DIO de uma sequência, ou calcula o *rank* associado ao nó vizinho emissor do DIO em questão, a sequência é incrementada;
- ***etxTable*** – Após receber um DIOACK, o nó calcula o valor de ETX para o nó vizinho e armazena esse valor em memória para posterior calculo do *rank*. Esta tabela é actualizada à medida que os DIOACKs são recebidos ao longo do tempo;
- ***rankTable*** – Todos os *ranks* calculados por um nó, associados aos nós vizinhos no seu raio de alcance, são armazenado nesta tabela. Assim, sempre que um nó perca ligação com o seu nó pai, procurará o vizinho com o qual tem o menor custo de encaminhamento.

No caso do HOP-COUNT e do RSSI, é apenas utilizada a tabela *rankTable* para armazenar os valores do *rank* associado a cada nó vizinho. Os ficheiros .h do ETX, HOP-COUNT e RSSI são apresentados nos Anexos C.7, C.8 e C.9, respectivamente.

O algoritmo *Trickle* desenvolvido é apresentado no Anexo C.6.

Periodicamente, de 10 em 10 segundos, a camada de aplicação envia 50 bytes de dados. O cabeçalho RPL ocupa no total 25 bytes, incluindo os campos adicionados pelo 6LoWPAN. Com os encapsulamentos da camada MAC e PHY, uma trama tem no total 90 bytes de comprimento.

C.1 Módulo Nic802154_TI_CC2420

```
module Nic802154_TI_CC2420 extends WirelessNicBattery
{
  parameters:
    macType = "CSMA802154";

    sleepCurrent        = 0.000021mA;
    rxCurrent           = 18.8 mA;
    decodingCurrentDelta = 0 mA;
    txCurrent           = 17.4 mA;
    setupRxCurrent      = 0.6391mA;
    setupTxCurrent      = 0.6845mA;
    rxTxCurrent         = 18.8 mA;
    txRxCurrent         = 18.8 mA;

    phy.decider = xmldoc("Nic802154_TI_CC2420_Decider.xml");
    phy.headerLength = 48 bit;
    phy.thermalNoise = default(-110 dBm);
    phy.timeSleepToRX = 0.001792 s;
    phy.timeSleepToTX = 0.001792 s;
    phy.timeRXToTX = 0.000192 s;
    phy.timeTXToRX = 0.000192 s;
    phy.timeRXToSleep = 0 s;
    phy.timeTXToSleep = 0 s;
```

```
        mac.rxSetupTime = 0.001792 s;
        mac.txPower = default(1 mW);
}
```

C.2 Módulo CSMA802154

```
simple CSMA802154 extends csma
{
    parameters:
        @class(CSMA802154);

        headerLength @unit(bit) = default(72 bit);
        macMaxCSMABackoffs = default(4);  // takes values between 0 and 5
        //txPower @unit(mW);
        useMACAcks = default(true);

        backoffMethod = "exponential";
        macMaxBE = default(5);    // takes values between 3 and 8
        macMinBE = default(3);    // takes values between 0 and macMaxBE
}
```

C.3 Recepção de pacotes DIO em ETX-WSN

```
            tRouteNbDIOTable::iterator pos0;
            tRouteSeqDIOTable::iterator pos1;
            pos0 = nbDIOTable.find(initialSrcAddr);
            pos1 = seqDIOTable.find(initialSrcAddr);

            if ((pos0 == nbDIOTable.end()) && (pos1 == seqDIOTable.end()))
            {
                int newEntry = 1;
                nbDIOTable.insert(make_pair(initialSrcAddr, newEntry));
                seqDIOTable.insert(make_pair(initialSrcAddr, seqNodeDIO));
            }
            else
            {
                int lastEntry = pos0->second;
                int preSeqDIO = pos1->second;

                if((preSeqDIO < seqNodeDIO) || (nbNodeDIO < lastEntry))
                {
                    if(rank != 0 && initialSrcAddr != 0)
                    {
                        handleACKDIOMsg(initialSrcAddr, lastEntry, preSeqDIO, rank);
                    }
                    nbDIOTable[initialSrcAddr] = 1;
                    seqDIOTable[initialSrcAddr] = seqNodeDIO;
                }
                else
                {
                    if((lastEntry < 4) && (nbNodeDIO < 4))
                    {
                        nbDIOTable[initialSrcAddr] = lastEntry+1;
                        int newEntry = pos0->second;
                    }
                    else if((lastEntry <= 4) && (nbNodeDIO == 4))
```

```
            {
              nbDIOTable[initialSrcAddr] = lastEntry+1;
              int newEntry = pos0->second;

              if(rank != 0 && initialSrcAddr != 0)
              {
                 handleACKDIOMsg(initialSrcAddr, newEntry, preSeqDIO, rank);
              }
              nbDIOTable[initialSrcAddr] = 0;
              seqDIOTable[initialSrcAddr] = seqNodeDIO + 1;
            }
          }
        }
```

C.4 Cálculo do ETX (versão DIO-ID = 4)

```
        int res = seqDIO - seq; //Cálcula possíveis atrasos
        float number = (float) numbDIO;
        if(numbDIO == 0)

        if (res == 0)
         {
                float etx = (1/(number/4));
                return etx;
        }
        else
        {
                float etx = 1/((number/4)/res);
                return etx;
        }
}
```

C.5 Cálculo do *rank* e selecção do nó pai em ETX-WSN

```
if(rankParent != 0)
{
   tETXTable::iterator pos0;
   tRankTable::iterator pos1;
   pos0 = etxTable.find(origin);
   pos1 = rankTable.find(origin);

   if((pos0 == etxTable.end()) || (pos1 == rankTable.end()))
   {
      etxTable.insert(make_pair(origin, etxResol));
      float r = rankParent*etxResol + 1;
      rankTable.insert(make_pair(origin, r));

      if(rank == 0)
      {
         parentAdd = origin;
         rank = r;
         converged = true;
      }

      else
```

```
        {
            std::pair<LAddress::L3Type, float> min = *min_element(rankTable.begin(),
rankTable.end(), Compare());

            rank = min.second;
            LAddress::L3Type parentAddEXEM = min.first;
            parentAdd = parentAddEXEM;
        }
    }

    else
    {
        float e = pos0->second;
        float selfTempRank = rankParent*(0.5*etxResol + (1-0.5)*e) + 1;

        if (selfTempRank >= rank)
        {
            if(origin == parentAdd)
            {
                rank = selfTempRank;
                EV<<"Mantido o mesmo nó pai com rank  != "<<rank<<endl;
            }

            else
            {
                EV<<"Mantido o mesmo nó pai com rank = "<<rank<<endl;
            }
        }

        else if (selfTempRank<rank)
        {
            rank = selfTempRank;
            parentAdd = origin;
        }

        etxTable[origin] = etxResol;
        float t = pos0->second;
        rankTable[origin] = selfTempRank;
    }
}

else{
    EV<<"Nó emissor do ACK-DIO não possui qualquer rank"<<endl;
}
```

C.6 Trickle

```
        float f = 0;
        float t = trickleTime(f);
        if(rank!=0 && converged == true && nbDIO == 4)
        {
                scheduleAt(simTime()+t+15, DIOTimer);
        }
        else
        {
                scheduleAt(simTime()+t, DIOTimer);
        }
```

```
        float RPL::trickleTime(float f)
        {
        while (f<0.5)
        {
                f = ((float)rand()/((float)(RAND_MAX)));
        }
        return f;
}
```

C.7 RPLETX.h

```
/* file:    RPLETX.h
 *
 *  Created on: 22 de Jun de 2013
 *      Author: Paulo Torres Gouveia
 */
#ifndef RPLETX_H_
#define RPLETX_H_
#include <map>
#include <omnetpp.h>
#include "MiXiMDefs.h"
#include <BaseNetwLayer.h>
#include <BaseLayer.h>
#include <BaseApplLayer.h>
#include <SimpleBattery.h>
#include <RPLPkt_m.h>
class SimTracer;
using std::vector;

class RPLETX: public BaseNetwLayer
{
private:
   RPLETX(const RPL&);
   cMessage* DIOTimer;
   int seqDIO;
   float rank;
   int nbDIOPacketsReceived;
   int nbDIOPacketsSend;
   LAddress::L3Type sinkAddress;
   LAddress::L2Type macaddress;
   int headerLength;
   bool converged;
```

```
    int nbDIO;
    cMessage* DIOACKResponse;
    cMessage* DISMsg;
    LAddress::L3Type parentAdd;
    LAddress::L3Type desigParentAdd;
    int nbDataPacketsReceived;
    int nbDataPacketsForwarded;
    int nbDataPacketsSend;
    bool useSimTracer;
    bool trace, stats, debug;
    SimTracer *tracer;
    simtime_t timeToData;
    simtime_t timeReceivedData;
    simtime_t timeToConverge;
    cOutVector vectorTimeReceivedData;
    cOutVector vectorTimeToData;
    cOutVector dataPacketNodeAdd;
    cOutVector vectorTimeToConverge;

    bool compar(std::pair<LAddress::L3Type, float> i, std::pair<LAddress::L3Type, float> j)
    {
       return i.second < j.second;
    }
public:
    RPLETX()
    : BaseNetwLayer(),
      DIOTimer(NULL),
      seqDIO(0),
      rank(0),
      nbDIOPacketsReceived(0),
      nbDIOPacketsSend(0),
      sinkAddress(),
      macaddress(),
      headerLength(),
      converged(false),
      nbDIO(0),
      DIOACKResponse(NULL),
      DISMsg(NULL),
      parentAdd(),
      desigParentAdd(NULL),
```

```
    nbDataPacketsReceived(0),
    nbDataPacketsForwarded(0),
    nbDataPacketsSend(0),
    useSimTracer(false),
    trace(true), stats(false), debug(false),
    tracer(NULL),
    timeToData(),
    timeReceivedData(),
    timeToConverge(),
    vectorTimeReceivedData(),
    vectorTimeToData(),
    dataPacketNodeAdd(),
    vectorTimeToConverge()
  {}

  virtual void initialize(int);
  virtual void finish();

protected:
  enum messagesTypes {
        UNKNOWN=0,
        DATA,
        DIO,
        DIOACK,
        DIS
     };
  typedef std::map<LAddress::L3Type, int> tRouteNbDIOTable;
  tRouteNbDIOTable nbDIOTable;

  typedef std::map<LAddress::L3Type, int> tRouteSeqDIOTable;
  tRouteSeqDIOTable seqDIOTable;

  typedef std::map<LAddress::L3Type, float> tETXTable;
  tETXTable etxTable;

  typedef std::map<LAddress::L3Type, float> tRankTable;
  tRankTable rankTable;

  struct Compare
  {
```

```
        bool operator()(std::pair<LAddress::L3Type, float> i, std::pair<LAddress::L3Type, float>
j) const
        {
            return i.second < j.second;
        }
    };

    /** @brief Handle messages from upper layer */
    virtual void handleUpperMsg(cMessage* msg);

    /** @brief Handle messages from lower layer */
    virtual void handleLowerMsg(cMessage* msg);

    /** @brief Handle self messages */
    virtual void handleSelfMsg(cMessage* msg);

    /** @brief Handle control messages from lower layer */
    virtual void handleLowerControl(cMessage* msg);

    /** Handle Parent Selection */
    virtual void parentSelection(const tRouteNbDIOTable::key_type& origin, float rankParent,
float etxResoldeDIO, int seqNodeDIO, int seqNeigDIO);

    /** Handle DIO-ACK Messages */
    virtual void handleACKDIOMsg(const LAddress::L3Type srcAddr, int nb, int seq, float
selfRank);

    /** Handle DIS Messages */
    virtual void handleDISMsg(const LAddress::L3Type srcAddr);

    cMessage* decapsMsg(RPLPkt *msg);

    /** Calculate the Trickle Time and return it*/
    float trickleTime(float f);

    /** Calcualte the ETX and return it*/
    float ETXCalc(int nb, int seq); //RPL_v1
};
#endif /* RPLETX_H_ */
```

C.8 hopCountRPL.h

```
/*
 * hopCountRPL.h
 *
 *  Created on: 30 de Jul de 2013
 *      Author: Paulo Torres
 */

#ifndef HOP-COUNTRPL_H_
#define HOP-COUNTRPL_H_

#include <map>
#include <omnetpp.h>

#include "MiXiMDefs.h"
#include "BaseNetwLayer.h"
#include "SimpleAddress.h"
#include <BaseLayer.h>
#include <BaseApplLayer.h>
#include <SimpleBattery.h>
#include <RPLPkt_m.h>

class SimTracer;

class MIXIM_API hopCountRPL : public BaseNetwLayer
{
private:

   hopCountRPL(const hopCountRPL&);

   cMessage* DIOTimer;
   cMessage* DISTimer;
   LAddress::L3Type sinkAddress;
   LAddress::L2Type macaddress;
   LAddress::L3Type parentAdd;
   float rank;
   int headerLength;
   int nbDIOPacketsSend;
   int nbDIOPacketsReceived;
```

```
    int nbDataPacketsSend;
    int nbDataPacketsReceived;
    int nbDataPacketsForwarded;
    int nbHops;
    int nbDIS;
    bool converged;
    bool useSimTracer;
    bool trace, stats, debug;
    SimTracer *tracer;
    simtime_t timeToData;
    simtime_t timeReceivedData;
    simtime_t timeToConverge;
    simtime_t parentTimeElapse;
    cOutVector vectorTimeReceivedData;
    cOutVector vectorTimeToData;
    cOutVector dataPacketNodeAdd;
    cOutVector vectorTimeToConverge;
    int initialHOP;

public:
    hopCountRPL()
    : BaseNetwLayer(),
      DIOTimer(NULL),
      DISTimer(NULL),
      sinkAddress(),
      macaddress(),
      parentAdd(),
      rank(),
      headerLength(),
      nbDIOPacketsSend(),
      nbDIOPacketsReceived(),
      nbDataPacketsSend(),
      nbDataPacketsReceived(),
      nbDataPacketsForwarded(),
      nbHops(),
      nbDIS(),
      converged(),
      useSimTracer(),
      trace(false), stats(false), debug(false),
      tracer(NULL),
```

```
    timeToData(),
    timeReceivedData(),
    timeToConverge(),
    parentTimeElapse(),
    vectorTimeReceivedData(),
    vectorTimeToData(),
    dataPacketNodeAdd(),
    vectorTimeToConverge(),
    initialHOP()
  {}

  virtual void initialize(int);
  virtual void finish();

protected:

  enum messagesTypes {
        UNKNOWN=0,
        DATA,
        DIO,
        DIOACK,
        DIS
     };


  typedef std::map<LAddress::L3Type, float> tRankTable;
  tRankTable rankTable;

  /** @brief Handle messages from upper layer */
  virtual void handleUpperMsg(cMessage* msg);

  /** @brief Handle messages from lower layer */
  virtual void handleLowerMsg(cMessage* msg);

  /** @brief Handle self messages */
  virtual void handleSelfMsg(cMessage* msg);

  /** @brief Handle control messages from lower layer */
  virtual void handleLowerControl(cMessage* msg);
```

```
    /** Handle DIS Messages */
    virtual void handleDISMsg(const LAddress::L3Type srcAddr);

    cMessage* decapsMsg(RPLPkt *msg);

    /** Handle Parent Selection */
    virtual void parentSelection(const LAddress::L3Type srcAddr, float rankParent);

    /** Calculate the Trickle Time and return it*/
    float trickleTime(float f);

    struct Compare
    {
        bool operator()(std::pair<LAddress::L3Type, float> i, std::pair<LAddress::L3Type, float>
j) const
        {
            return i.second < j.second;
        }
    };

};

#endif /* HOP-COUNTRPL_H_ */
```

C.9 RPLRSSI.h

```
#ifndef RPLRSSI_H_
#define RPLRSSI_H_

#include <map>
#include <omnetpp.h>

#include "MiXiMDefs.h"
#include <BaseNetwLayer.h>
#include <BaseLayer.h>
#include <BaseApplLayer.h>
#include <SimpleBattery.h>
#include <RSSIPkt_m.h>

class SimTracer;
```

```
using std::vector;

class RPLRSSI: public BaseNetwLayer
{
private:
   RPLRSSI(const RPLRSSI&);

   cMessage* DIOTimer;
   float rank;
   int nbDIOPacketsReceived;
   int nbDIOPacketsSend;
   LAddress::L3Type sinkAddress;
   LAddress::L2Type macaddress;
   int headerLength;
   bool converged;
   int nbDIO;
   cMessage* DIOACKResponse;
   cMessage* DISMsg;
   LAddress::L3Type parentAdd;
   LAddress::L3Type desigParentAdd;
   int nbDataPacketsReceived;
   int nbDataPacketsForwarded;
   int nbDataPacketsSend;
   bool useSimTracer;
   bool trace, stats, debug;
   SimTracer *tracer;
   simtime_t timeToData;
   simtime_t timeReceivedData;
   simtime_t timeToConverge;
   cOutVector vectorTimeReceivedData;
   cOutVector vectorTimeToData;
   cOutVector dataPacketNodeAdd;
   cOutVector vectorTimeToConverge;
   double actualRSSI;

   bool compar(std::pair<LAddress::L3Type, float> i, std::pair<LAddress::L3Type, float> j)
   {
      return i.second < j.second;
   }
```

```
public:
    RPLRSSI()
    : BaseNetwLayer(),
      DIOTimer(NULL),
      rank(0),
      nbDIOPacketsReceived(0),
      nbDIOPacketsSend(0),
      sinkAddress(),
      macaddress(),
      headerLength(),
      converged(false),
      nbDIO(0),
      DIOACKResponse(NULL),
      DISMsg(NULL),
      parentAdd(),
      desigParentAdd(NULL),
      nbDataPacketsReceived(0),
      nbDataPacketsForwarded(0),
      nbDataPacketsSend(0),
      useSimTracer(false),
      trace(true), stats(false), debug(false),
      tracer(NULL),
      timeToData(),
      timeReceivedData(),
      timeToConverge(),
      vectorTimeReceivedData(),
      vectorTimeToData(),
      dataPacketNodeAdd(),
      vectorTimeToConverge(),
      actualRSSI()
    {}

    virtual void initialize(int);
    virtual void finish();

protected:

    enum messagesTypes {
          UNKNOWN=0,
          DATA,
```

```
        DIO,
        DIOACK,
        DIS
    };

  typedef std::map<LAddress::L3Type, float> tRankTempTable;
  tRankTempTable rankTempTable;

  typedef std::map<LAddress::L3Type, double> tRSSI;
  tRSSI RSSITable;

  struct Compare
  {
    bool operator()(std::pair<LAddress::L3Type, float> i, std::pair<LAddress::L3Type, float>
j) const
    {
      return i.second < j.second;
    }
  };

  virtual void handleUpperMsg(cMessage* msg);

  /** @brief Handle messages from lower layer */
  virtual void handleLowerMsg(cMessage* msg);

  /** @brief Handle self messages */
  virtual void handleSelfMsg(cMessage* msg);

  /** @brief Handle control messages from lower layer */
  virtual void handleLowerControl(cMessage* msg);

  /** Handle Parent Selection */
  virtual void parentSelection(const LAddress::L3Type srcAddr, float rankParent, float
rssiRec);

  /** Handle DIO-ACK Messages */
  virtual void handleACKDIOMsg(const LAddress::L3Type srcAddr, float selfRank, double
actualRSSI);
```

```
    /** Handle DIS Messages */
    virtual void handleDISMsg(const LAddress::L3Type srcAddr);

    cMessage* decapsMsg(RPLPkt *msg);

    /** Calculate the Trickle Time and return it*/
    float trickleTime(float f);

};

#endif /* RPLRSSI_H_ */
```

Referências

[A11] Ahmed Amokrane, Adlen Ksentini, Yassine Hadjadj-Aoul, "Congestion Control in the context of Machine type communications in 3gpp lte networks", *Dionysos, Inria Bretagne Atlantique*, University of Rennes, Rennes, France, Jan. 2011.

[ADM2M] "OWA21A-Track Datasheet", *Datasheet*. Available from: http://www.adaptivemodules.com/assets/File/datasheet-owa21a.pdf.

[ADV] "Advantech's First M2M GPS/GPRS Product for Fleet Management", 2008. Informação disponível em: http://www.advantech.com/sector/vehicle/News.aspx?doc_id={106EE45A-9FA5-4DE5-A7D6-127F75CDBB0.

[AGB10] Darie Angela, Mihai Ghenghea, Ion Bogdan, "Supporting environmental surveillance by using wireless sensor networks", *in Proc. of the 3rd International Symposium on Electrical and Electronics Engineering (ISEEE)*, pp.216-219, Sept. 2010.

[AGK11] Sergey Andreev, Olga Galinina, Yevgeni Koucheryavy, "Energy-Efficient Client Relay Scheme for M2M Communication" in *Proc. of the IEEE Global Telecommunications Conference (GLOBECOM 2011)*, Houston, Texas, USA, Dec. 2011.

[AK04] Ahmed E. Kamal, Jamal N. Al-Karaki, "Routing Techniques in Wireless Sensor Networks: A Survey", *IEEE Transactions on Wireless Communications*, Vol. 11, No. 6, Dec. 2004, pp. 6-28.

[AMP11] Nurul Amirah Ali, Micheal Drieberg, Patrick Sebastian, "Deployment of MICAz mote for Wireless Sensor Network applications", in *Proc. of the IEEE International Conference Computer Applications and Industrial Electronics (ICCAIE)*, Penang, Malaysia, Dec. 2011.

[AY03] Kemal Akkaya, Mohamed Younis, "A survey on routing protocols for wireless sensor networks", *Internationa Journal of Ad Hoc Networks*, Vol. 3, Issue 3, March 2005, pp 325-349.

[B01] David L. Brock, “The Electronic Product Code (EPC): A Naming Scheme for Physical Objects”, Auto-ID Center White Paper WH-002, MIT, Cambridge, MA, 2001.

[B06] John Buckley, “From RFID To the internet of things networked systems”, *Final Report of Information Society Technologies*, European Commission Directorate “Network and Communications Technologies”, March 2006.

[B12] M. Beale, "Future Challenged in Efficiently Supporting M2M in the LTE Standards”, in *Proc. of IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, Paris, France, April 2012.

[BatStuff] “Peukert’s Law | A Nerd’s Attempt to Explain Battery Capacity”. Informação disponível em: http://www.batterystuff.com/kb/tools/peukert-s-law-a-nerds-attempt-to-explain-battery-capacity.html.

[BCS12] Carsten Bormann, Angelo P. Castellani, Zach Shelby, “CoAP: An Application Protocol for Billions of Tiny Internet Nodes”, *IEEE Transactions on Internet Computing*, Vol. 16, No. 2, April 2007, pp. 62-67.

[BDWL10] Abdelmalik Bachir, Mischa Dohler, Thomas Watteyne, Kin K. Leung, “MAC Essentials for Wireless Sensor Networks”, *IEEE Transactions on Communications Surveys & Tutorials*, Vol.12, No.2, Abril-Junho 2010, pp.222-248.

[BGV13a] Norberto Barroca, Paulo T. Gouveia, Fernando J. Velez, “Impact of Switching Latency Times in Energy Consumption of IEEE 802.15.4 Radio Transceivers”, in *Proc. of the 9th Conference on Telecommunications (ConfTele)*, Castelo Branco, Portugal, March 2013.

[BGV13b] Norberto Barroca, Paulo T. Gouveia, Fernando J. Velez, “Impact of Switching Latency Times in Energy Consumption of IEEE 802.15.4 Radio Transceivers”, submitted to the Best Student Paper Award, *7th URSI Seminar of the Portuguese Committee*, Lisbon, Portugal, Nov. 2013.

[BGZR11] M.J. Booysen, J.S. Gilmore, S. Zeadally, G.J.van Rooyen, “Machine to Machine (M2M) for Vehicular Networks”, *KSII Transactions on Internet and Information Systems*, Vol. 6, Issue 2, Dec. 2011, pp. 529-549.

[BSGT13a] Norberto Barroca, Henrique M. Saraiva, Paulo T. Gouveia, Jorge Tavares, Luís M. Borges, Fernando J. Velez, Caroline Loss, Rita Salvado, Pedro Pinho, Ricardo Gonçalves, Nuno Borges de Carvalho, Raúl Chavéz-Santiago, Ilangko

Balasingham, “Opportunities, Antennas and Circuits for RF Energy Harvesting in Wireless Body Area Nerworks”, in *Proc. of the* 24^{th} *International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC)*, London, UK, Sept. 2013.

[BSGT13b] Norberto Barroca, Henrique M. Saraiva, Paulo T. Gouveia, Jorge Tavares, Luís M. Borges, Fernando J. Velez, Caroline Loss, Rita Salvado, Pedro Pinho, Ricardo Gonçalves, Nuno Borges de Carvalho, Raúl Chavéz-Santiago, Ilangko Balasingham, “Antennas and Circuits for Ambient RF Energy Harvesting in Wireless Body Area Nerworks”, in *Proc. of the* 8^{th} *Meeting of the Management Committee of COST IC 1004-Cooperative Radio Communications for Green Smart Environments*, TD(13)08068, Ghent, Belgium, Sept. 2013.

[BTGV13] Norberto Barroca, Jorge Tavares, Paulo T. Gouveia, Fernando J. Velez, “Wireless Sensor Network Platforms”, *Internal Report under PROENERGY-WSN project*, Covilhã, Portugal, June 2013.

[CABM03] Douglas S. J. De Couto, Daniel Aguayo, John Bicket, Robert Morris, “A High-Throughput Path Metric for Multi-Hop Wireless Ruting”, in *Proc. of the* 9^{th} *Annual International Conference on Mobile Computing and Networking (MobiCom)*, San Diego, Californa, USA, Sept. 2003.

[CCG12] Cosmin Cirstea, Mihail Cernaianu, Aurel Gontean, “Packet loss analysis in wireless sensor networks routing protocols”, in *Proc. of the* 35^{th} *International Conference on Telecommunications and Signal Processing (TSP)*, St. Petersburg, Florida, USA, July 2012.

[CERN1] “The birth of the web”. Informação disponível em: http://home.web.cern.ch/about/birth-web.

[CERN2] “CERN Data Centre passes 100 petabytes”. Informação disponível em: http://home.web.cern.ch/about/updates/2013/02/cern-data-centre-passes-100-petabytes.

[CICRSY] Aminul Haque Chowdhury, Muhammad Ikram, Hyon-Soo Cha, Hassen Redwan, S.M. Saif Shams, Ki-Hyung Kim, Seung-Wha Yoo, “Route-Over vs Mesh-Under Routing in 6LoWPAN”, in *Proc. of* 5^{th} *International Wireless Communications and Mobile Computing Conference (IWCMC)*, Leipzig, Germany, June 2009.

[CLSB12] Nadia Catenazzi, Vanessa De Luca, Lorenzo Sommaruga, Massimo Botta, “Guidelines to design inclusive Ambient Intelligence solutions for human

activity sharing", in *Proc. of the 6th International Conference on Complex, Intelligent, and Software Intensive Systems (CISIS)*, Palermo, Italy, July 2012.

[CLPKEAH11] Alexandru Caracas,Clemens Lombriser,Yvonne Anne Pignolet,Thorsten Kramp,Thomas Eirich,Rolf Adelsberger,Urs Hunkeler, "Energy-efficiency through micro-managing communication and optimizing sleep", in *Proc of the 8th Annual IEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*, Salt Lake City, Utah, USA, June 2011.

[COBIS] "CoBIs Project: Collaborative Bussiness Items". Available from: http://www.cobis-online.de.

[ConOS] Contiki Operating System Website: www.contikios.com.

[CREa13] CREaTION, "Cognitive Radio Transceiver Design for Energy Efficient Data Transmission". Available from: http://www.av.it.pt/creation/.

[DFMM06] Henri Dubois-Ferrière, Laurent Fabre, Roger Meier, Pierre Metrailler, "TinyNode: a comprehensive platform for wireless sensor network applications", in *Proc. of the 5th International Conference on Information Processing in Sensor Networks (IPSN)*, Nashville, Tennessee, USA, April 2006.

[DGAZ10] Nelson I. Dopico, Carlos Gil-Soriano, I˜nigo Arrazola, Santiago Zazo, "Analysis of IEEE 802.15.4 Throughput in Beaconless Mode on micaZ under TinyOS 2", in *Proc. of the IEEE 75th Vehicular Technology Conference Fall (VTC)*, Ottawa, Canada, Sept. 2010.

[DigiM2M] "Digi M2M Solution Builder Kit: Cloud-enabled end-to-end solution for M2M applications". Informação disponível em: http://www.digi.com/products/wireless-routers-gateways/kits/m2m-solution-builder-kit#overview.

[DRAFT18] Z. Shelby, K. Hartke, C. Bormann, "Constrained Application Protocol (CoAP)", *IETF Standards Track RFC Proposed Standard*, CORE Working Group, 2013. Available from: http://datatracker.ietf.org/doc/draft-ietf-core-coap/.

[ETSIM2M] Machine-to-Machine Communications, ETSI Standards. Available from: http://www.etsi.org/technologies-clusters/technologies/m2m.

[EZ] "eZ430-RF2500 Application Report", *Report*. Available from: http://www.ti.com/lit/an/slaa378d/slaa378d.pdf.

[F08] Shahin Farahani, "ZigBee Wireless Networks and Transceivers", Newnes, 2008.

[Fa10] H. Farhangi, "The path of the Smart Grid", *IEEE Transactions on Power and Energy Magazine*, Vol.8, No.1, Jan-Feb. 2010, pp.18-28.

[FASL01] Fan Ye, A. Chen, Songwu Lu, Lixia Zhang, "A scalable solution to minimum cost forwarding in large scale sensor networks", in *Proc. of the 10th International Conference on Computer Communications and Networks (ICCCN)*, Scottsdale, Arizona, USA, Oct. 2001.

[FSM2M] Iain Davidson, "Machine-to-Machine (M2M) Gateway: Trusted and Connected Intelligence", *Freescale Semicondutor Document*. Available from: http://www.freescale.com/files/32bit/doc/brochure/PWRARBYNDBITSMTM.pdf.

[FV12] João M. Ferro, Fernando J. Velez, "Combined Hop Count and Received Signal Strength Routing Protocol for Mobility–enabled", in *Proc. of the IEEE Vehicular Technology Conference (VTC-Fall)*, Québec City, Canada, Sept. 2012.

[GH09] Vehbi C. Gungor, Gerhard P. Hancke, "Industrial Wireless Sensor Networks: Challenges, Design Principles, and Technical Approaches", *IEEE Transactions on Industrial Electronics*, Vol. 56, No. 10, Oct. 2009, pp. 4258-4265.

[GFJML09] Omprakash Gnawali, Rodrigo Fonseca, Kyle Jamieson, David Moss, Philip Levis "Collection Tree Protocol", in *Proc. of the* 17th *ACM Conference on Embedded Networked Sensor Systems (SenSys)*, Berkeley, California, USA, Nov. 2009.

[GLA12] Antonis G. Gotsis, Athanasios S. Lioumpas, Angeliki Alexiou "M2M Scheduling over LTE – Challenges and New Prespectives", *IEEE Transactions on Vehicular Technology Magazine*, Vol. 7, No. 3, Sept. 2012, pp.34-39.

[GTPL09] Dominique Guinard, Vlad Trifa, Thomas Pham, Olivier Liechti, "Towards Physical Mashups in the Web of Things", in *Proc. of the 6th International Conference on Networked Sensing Systems (INSS),* Pittsburgh, Pennsylvania, USA, June 2009.

[HARBOR] Harbor Research, "Machine-To-Machine (M2M) & Smart Systems Market Opportunity 2010-2014", 2011. Available from: http://www.windriver.com/m2m/edk/Harbor_Research-M2M_and_Smart_Sys_Report.pdf.

[HBB11] Ziaul Hasan, Hamidreza Boostanimehr, Vijay K. Bhargava, "Green Cellular Networks: A Survey, Some Research Issues and Challenges", *IEEE Transactions on Communications Surveys & Tutorials*, Vol. 13, No. 4, Oct-Dec. 2011, pp.524-540.

[HBE12] Olivier Hersent, David Boswarthick, Omar Elloumi, "The Internet of Things: Key Applications and Protocols", John Wiley & Sons, 2012.

[HC08] Jonathan W. Hui, David E. Culler, "Extending IP to Low-Power, Wireless Personal Area Network", *IEEE Transactions on Internet Computing*, Vol. 12, No. 4, July-Aug. 2008, pp.37-45.

[HC10] Jonathan W. Hui, David E. Culler, "IPv6 in Low-Power Wireless Networks", *IEEE Transactions on Proceedings of the IEEE*, Vol. 98, No. 11, Nov. 2010, pp.1865-1878.

[HCB00] Wendi Rabiner Heinzelman, Anantha Chandrakasan, Hari Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks", in *Proc. of the 33th Annual Hawaii International Conference on System Sciences (HICSS)*, Maui, Hawaii, USA, Jan. 2000.

[ICTSE] "Sensei Project: Integrating the Physical with the Digital World of the Network of the Future". Available from: http://www.ict-sensei.org.

[IEE03] IEEE Standard for Information technology-Telecommunications and information exchange between systems-PART 15.4:Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs): Amendment to add alternate PHY (Amendment of IEEE Std 802.15.4), *IEEE Std 802.15.4a/D7*, 2007.

[IEEE80211] IEEE Standard for Information technology--Telecommunications and information exchange between systems Local and metropolitan area networks--Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, *IEEE Std 802.11-2012*.

[IEEE80216P] IEEE Standard for Air Interface for Broadband Wireless Access Systems - Amendment 1: Enhancements to Support Machine-to-Machine Applications, *IEEE Std 802.16p-2012*.

[IEEEM2M] IEEE 802.16's Machine-to-Machine (M2M) Task Group. Available from: http://grouper.ieee.org/groups/802/16/m2m/index.html.

[IJM11] Ozlem Durmaz Incel, Pierre Jansen, Sape Mullender, "MC-LMAC: A Multi-Channel MAC Protocol for Wireless Sensor Networks", *International Journal of Ad Hoc Network*, Vol. 9, Issue 1, Jan. 2011, pp.73-94.

[IM2] "IMote2 Datasheet", *Datasheet*. Available from: http://wsn.cse.wustl.edu/images/e/e3/Imote2_Datasheet.pdf.

[IPSO09] Jonathan Hui, David Culler, Samita Chakrabarti, "6LoWPAN: Incorporating IEEE 802.15.4 into the IP architecture", *Internet Protocol for Smart Objects Alliance Whitepaper #3 (IPSO)*, Jan. 2009.

[ITREU] IERC Project: European Reserach Cluster on the Internet of Things. Available from: http://www.internet-of-things-research.eu

[IS] "IRIS Datasheet", *Datasheet*. Available from: http://www.memsic.com/userfiles/files/Datasheets/WSN/IRIS_Datasheet.pdf

[JG08] M.R. Jongerden, B.R. Haverkort, "Battery Modeling", *Report of the Centre for Telematics and Information Technology,* University of Twente, Enschede, Netherlands. Available from: http://doc.utwente.nl/64556/.

[JPL10] Kwang-Ryul Jung, Aesoon Park, Sungwon Lee, "Machine-Type-Communication (MTC) Device Grouping Algorithm for Congestion Avoidance of MTC Oriented LTE Network", in *Proc. of the 1st International Conference on Security-Enriched Urban Computing and Smart Grid (SUComS)*, Daejeon, Korea, Sept. 2010.

[KA12] Kaivan Karimi, Gary Atkinson, "What the Internet of Things Need to Become a Reality", *Freescale Semicondutor Inc and ARM Inc Whitepaper*, Sept. 2012.

[KETHVDTDC11] JeongGil Ko, Joakim Eriksson, Nicolas Tsiftes, Stephen Dawson-Haggerty, Jean-Philippe Vasseur, Mathilde Durvy, Andreas Terzis, Adam Dunkels, David Culler, "Beyond Interoperability – Pushing the Performance of Sensor Network IP Stacks", in *Proc. of the International Conference on Embedded Networked Sensor Systems (ACM SenSys),* Seattle, Washington, Nov. 2011.

[KKSBM12] MinJi Kim, Thierry Klein, Emina Soljanin, João Barros, Muriel M´edard, "Trade-off between cost and goodput in wireless", *Computing Research Repository (CoRR)*, abs/1203.2841, March 2012.

[KKVBSG12] Anupama K., Nishad Kamdar, Dhruv Vyas, Ishaan Baokar, Siddharth Sahu, Philip George, "Design and Implementation of a Cross Layered Protocol Stack

for Sensor Networks in an Indoor Environment", in *Proc. of the 9th International Conference on Wireless and Optical Communications Networks (WOCN)*, Indore, India, Sept. 2012.

[KSW08] Andreas Köpke, Michael Swigulski, Karl Wessel, Daniel Willkomm, Klein Haneveld, T.E.V. Parker, Otto Visser, Hermann Simon Lichte, Stefan Valentin, "Simulating Wireless and Mobile Networks in OMNeT++ The MiXiM Vision", in *Proc. of the 1st International Workshop on OMNeT++*, Cannes, French Riviera, March 2008.

[KTHHCHL11] JeongGil Ko, Andreas Terzis, Stephen Dawson-Haggerty, David E. Culler, Jonathan W. Hui, Philip Levis, "Connecting Low-Power and Lossy networks to the Internet", *IEEE Transactions on Communications Magazine*, Vol. 49, No. 4, April 2011, pp.96-101.

[KW05] Holger Karl, Andreas Willig. "Protocols and Architectures for Wireless Sensor Networks", John Wiley & Sons, 2005.

[LFZS09] Xiang Li, Ling Feng, Lizhu Zhou, Yuanchun Shi, "Learning in an Ambient Intelligent World: Enabling Technologies and Practices", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 21, No. 6, June 2009, pp.910-924

[LKY11] Ki-Dong Lee, Sang Kim, Byung Yi, "Throughput Comparison of Random access methods for m2m service over lte networks", in *Proc. of IEEE GLOBECOM Workshops (GC Wkshps)*, Houston, Texas, USA, Dec. 2011.

[LO] "Lotus Datasheet", *Datasheet*. Available from: http://www.memsic.com/userfiles/files/Datasheets/WSN/6020-0705-01_A_LOTUS.pdf.

[LPCS05] Philip Levis, Neil Patel, David Culler, Scott Shenker, "Trickle: A Self-Reguling Algorithm for Code Propagation and Maintenance in Wireless Sensor Networks", in *Proc. of the 1st Conference on Symposium on Networked Systems Design and Implementation (NSDI)*, San Diego, California, March 2004

[LS08] Gérald Santucci, Sebastian Lange, "Internet of things in 2020: A Roadmap for the Future", *RFID Working Group of the European Technology Platform on Smart Systems Integration* (EPOSS), 2008.

[LSYZ10] Hui Li, Jianghong Shi, Qi Yang, Dezhong Zhang, "An Energy-efficient hierarchical routing protocol for long range transmission in wireless sensor

networks", in *Proc. of the 2nd International Conference on Education Technology and Computer (ICETC)*, Shanghai, China, June 2010.

[LXCDW11] Jialiang Lu, Zhengzheng Xu, Lionel Croix, Anis Darwich, Min-You Wu, "Interfering Mobile Target Motion Planning in Wireless Sensor Networks", in *Proc. of the IEEE Global Telecommunication Conference*, Houston, Texas, USA, Dec. 2011.

[LXKK11] Li Li, Hu Xiaoguang, Chen Ke, He Ketai, "The Applications Of WiFi-based Wireless Sensor Network In Internet Of Things And Smart Grid", in *Proc. of the 6th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, Beijing, China, June 2011.

[MAI11] Kazumine Matoba, Ken-ichi Abiru, Tomohiro Ishihara, "Service Oriented Network Architecture for Scalable M2M and Sensor Network Services", in *Proc. of the 15th International Conference on Intelligence in Next Generation Networks (ICIN)*, Berlin, Germany, Oct. 2011.

[MBUS] "The M-Bus: A Documentation ". Available from: http://www.m-bus.com/mbusdoc/md3.php.

[MeshX] "Meshlium Xtreme Datasheet", *Datasheet*. Available from: http://www.libelium.com/development/meshlium/documentation/.

[MHD08] Geoffrey G. Messier, Jennifer A. Hartwell, and Robert J. Davies, "A Sensor Network Cross-Layer Power Control Algorithm that Incorporates Multiple-Access Interference", *IEEE Transactions on Wireless Communication*, Vol. 7, No. 8, Aug. 2008.

[MLK11] Boran Morvaj, Luka Lugaric, Slavko Krajcar, "Demonstrating smart buildings and smart grid features in a smart energy city", in *Proc. of the 3rd International Youth Conference on Energetics (IYCE)*, Leiria, Portugal, June 2011.

[Multiv6] "IPv6 Multicast Address Space Registry". Available from: http://www.iana.org/assignments/ipv6-multicast-addresses/ipv6-multicast-addresses.xhtml.

[MYM11] Gerard Rudolph Mendez, Mohd Amri Md Yunus, Subhas Chandra Mukhopadhyay, "A WiFi based Smart Wireless Sensor Network for an Agricultural Environment", in *Proc. of the 5th International Conference on Sensing Technology (ICST)*, Palmerston North, New Zealand, Nov-Dec., 2011

[MZ] "MicaZ Datasheet", *Datasheet*. Available from: http://www.openautomation.net/uploadsproductos/micaz_datasheet.pdf.

[NLM08] Xian Ni, Kun-chan Lan, Robert Malaney, "On the Performance of Expected Transmission Count (ETX) for Wireless Mesh Network", in *Proc. of the 3rd International Conference on Performance Evaluation Methodologies and Tools*, Athens, Greece, Oct. 2008.

[NR12] Alaparthi Narmada, Parvataneni Sudhakara Rao, "ZigBee Based WSN with IP Connectivity", in *Proc. of 4th International Conference on Computational Intelligence, Modelling and Simulation (CIMSiM),* Kuantan, Malaysia, Sept. 2012.

[OECD12] "Machine-to-Machine Communications – Connecting Billions of Devices", *Working Party on Communication Infrastructures and Services Policy*, OECD Digital Economy Papers, No. 192, 2012.

[OH10] Jongtaek Oh, Zygmunt J. Haas, "Personal environment service based on the integration of mobile communications and wireless personal area networks", *IEEE Transactions on Communications Magazine*, Vol. 48, No. 6, June 2010, pp. 66-72.

[OSGM2M] Walt Bowers, "Enabling Smart Data on M2M Gateways and Aggregators", in *Proc. of the OSGi DevCon*, Boston, Massachusetts, USA, March 2013.

[OSR11] Luís M. L. Oliveira, Amaro F. de Sousa, Joel J. P. C. Rodrigues, "Routing and mobility approaches in IPv6 over LoWPAN mesh networks", *International Journal of Communication Systems*, Vol. 24, Feb. 2011, pp. 1445–1466.

[P06] Gilles Privat, "From Smart Devices to Ambient Communication", in *Proc. of the Workshop on From RFID to the Internet of Things*, Brussels, Belgium, May 2006.

[PB94] Charles Perkins, Pravin Bhagwat, "Highly dynamic Destination-Sequenced Distance-Vector routing (DSDV) for mobile computers", in *Proc. of the Conference on Communications Arquitectures, Protocols and Applications*, London, UK, Aug-Sept. 1994.

[PC11] S. Petersen, S. Carlsen, "WirelessHart versus ISA100.11 a: The Format War Hits the Factory Floor", *IEEE Transactions on Industrial Electronics Magazine*, Vol. 5, No. 4, Dec. 2011, pp.23-34.

[PHCSS03] Jianping Pan, Y. Thomas Hou, Lin Cai, Yi Shi, Sherman X. Shen, "Locating Base-Stations for Video Sensor Networks", in *Proc. of the 58th Vehicular Technology Conference (VTC)*, Orlando, Florida, USA, Oct. 2003.

[PR99] Charles Perkins, Elizabeth Royer, "Ad-hoc on-demand distance vector routing", in *Proc of the 2nd IEEE Workshop on Mobile Computing Systems and Applications (WMCSA)*, New Orleans, Louisiana, USA, Fev. 1999.

[PSS01] Sung Park, Andreas Savvides, Mani B. Srivastava, "Battery Capacity Measurement And Analysis Using Lithium Coin Cell Battery", in *Proc. of the International Symposium on Low Power Electronics and Design*, Huntington Beach, California, USA, Agosto 2001.

[PWSN13] PROENERGY-WSN, "Prototypes for Efficient Energy Self-sustainable Wireless Sensor Networks". Available from: http://www.e-projects.ubi.pt/proenergy-wsn/.

[QC02] Daji Qiao, Sunghyun Choi, "Goodput analysis and link adaptation for IEEE 802.11a wireless LANs", *IEEE Transactions on Mobile Computing*, Vol.1, Oct-Dec. 2002, pp. 278-292.

[Ram01] Siddharth Ramesh, "Protocol Architecture for Wireless Sensor Networks" in *Proc. of the 1st ACM International Workshop on Wireless Sensor Networks and Applications (WSNA)*, Atlanta, Georgia, USA, Sept. 2002.

[R1072277] "Downlink interference coordination", *3GPP R1-072277*, 3GPP TSG RAN WG1 #49 Meeting, March 2007.

[RCS05] Saikat Ray, Jeffrey B. Carruthers, David Starobinski, "Evaluation of the Masked Node Problem in Ad Hoc LANs", IEEE Transactions on Mobile Computing, Vol. 4, No. 5, Sept-Oct. 2005, pp. 430-442.

[RFC4944] G. Montenegro, N. Kushalnagar, J. Hui, D. Culler "Transmission of IPv6 Packets over IEEE 802.15.4 Networks", *IETF Standards Track RFC 4944*, 6LoWPAN Working Group, 2007. Available from: http://datatracker.ietf.org/doc/rfc4944/.

[RFC6550] P. Thubert, A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, JP. Vasseur, R. Alexander, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks", *IETF Standards Track RFC 6550*, ROLL Working Group, 2012. Available from: http://datatracker.ietf.org/doc/rfc6550/.

[RFC6550] P. Levis, T. Clausen, J. Hui, O. Gnawali, J. Ko, "The Trickle Algorithm", *IETF Standards Track RFC 6206,* ROLL Working Group, 2011. Available from: http://www.rfc-base.org/txt/rfc-6206.txt.

[RL09] Rodrigo Roman, Javier Lopez, "Integrating Wireless Sensor Networks and the Internet: A Security Analysis", *International Journal of Internet Research*, Vol. 19, Issue 2, 2009, pp. 246-259.

[RNL11] Rodrigo Roman, Pablo Najera, Javier Lopez, "Securing the Internet of Things", *IEEE Transactions on Computer Society*, Vol. 44, No. 9, Sept. 2011, pp. 51-58.

[RT99] Elizabeth M. Royer, Chai-Keong Toh, "A Review of current routing protocols for ad hoc mobile wireless networks", *IEEE Transactions on Personal Communications*, Vol. 6, No. 2, April 1999, pp. 46-55.

[SB09] Zach Shelby, Carsten Bormann, "6LoWPAN: the wireless embedded internet", Wiley, 2009.

[SMZ07] Kazem Sohraby, Daniel Minoli, Taieb Znati, "Wireless Sensor Networks: Technology, Protocols, and Applications", Wiley, 2007.

[SS] "Sun™ Small Programmable Object Technology (Sun SPOT) Theory of Operation", *Datasheet*. Available from: http://www.sunspotworld.com/docs/Purple/SunSPOT-TheoryOfOperation.pdf.

[SS10] Wenchao Sun, Meina Song, "A General M2M Device Model", in *Proc. of IEEE 2nd Symposium on Web Society (SWS)*, Beijing, China, Aug. 2010.

[SVFF12] Ian G Smith, Ovidiu Vermesan, Peter Friess, Anthony Furness, "The Internet of Things 2012 – New Horizons", *Internet of Things European Reserch Cluster (IERC)*, 2012.

[TB] "TelosB Datasheet", *Datasheet*. Available from: http://www.willow.co.uk/TelosB_Datasheet.pdf.

[TM09] K. Thanigaivelu, K. Murugan, "Reduced energy dissipation using Beacon Based Data Collection algorithm for mobile sink in wireless sensor networks", in *Proc. of the 1st International Conference on Advanced Computing (ICAC)*, Chennai, India, Dec. 2009.

[TMS] "T-Mote Sky Datasheet", *Datasheet*. Available from: http://www.eecs.harvard.edu/~konrad/projects/shimmer/references/tmote-sky-datasheet.pdf.

[TN] "Tinynode Fact Sheet", *Datasheet*. Available from: http://www.tinynode.com/?q=system/files/TN584_Fact_Sheet_v_1_1.pdf.

[TS102690] ETSI TS 102 690, "Machine-to-Machine communications (M2M): Functional Architecture", Technical Specification, 2011. Available from: http://www.etsi.org/deliver/etsi_ts/102600_102699/102690/01.01.01_60/ts_102690v010101p.pdf.

[V01] András Vargas, "The OMNeT++ Discrete Event Simulation System", in *Proc. of the 15th European Simulation Multiconference (ESM)*, Prague, Czech Republic, June 2001.

[UHM11] Dieter Uckelann, Mark Harrison, Florian Michahelles, "Architecting the Internet of Things", Springer Berlin Heidelberg, 2011.

[VS10] Krishnan V, Bhaswar Sanya, "M2M Technology: Challenges and Opportunities", *Tech Mahindra Whitepaper*, 2010. Available from: http://www.techmahindra.com/Documents/WhitePaper/M2MTechnology_ChallengesandOpportunities_Sept10.pdf.

[WA09] Klaus-Dieter Walter, "Implementing M2M applications via GPRS, EDGE and UMTS", in *Whitepaper of M2M Community*, 2009. Available from: http://m2m.com/docs/DOC-1003.

[WC01] Alec Woo, David E. Culler, "A Transmission control scheme for media access in sensor networks", in *Proc. of the 7th Annual International Conference on Mobile Computing and Networking (MobiCom)*, Rome, Italy, July 2001.

[WH] "WirelessHart LTP590 Datasheet", *Datasheet*. Available from: http://cds.linear.com/docs/en/datasheet/5903-WHRf.pdf.

[WK05] C. Westphal, R. Koodli,"Stateless IP Header Compression", in *Proc. of IEEE International Conference on Communications*, Vol. 5, May 2005, pp.3236-3241.

[WM] "Waspmote Datasheet", *Datasheet*. Available from: http://www.libelium.com/development/waspmote/documentation/.

[WSN802G] "WSN802G ESeries Wi-Fi Radio Module", *Datasheet*. Available from: http://www.rfm.com/products/data/wsn802g-e_sheet.pdf.

[WTZA10] Di Wang, Zhifeng Tao, Jinyun Zhang, Alhussein Abouzeid, "RPL Based for Advanced Metering infrastructure in Smart Grid", in *Proc. of IEEE International Conference on Communications Workshops (ICC)*, Cape Town, South Africa, May 2010.

[XHE01] Ya Xu, John Heidemann, Deborah Estrin, "Geography-informed Energy Conservation for Ad Hoc Routing", in *Proc. of the 7th Annual International Conference on Mobile Computing and Networking (ACM MobiCom)*, Rome, Italy, July 2001.

[YF04] Ossama Younis, Sonia Fahmy, "HEED: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networksh", *IEEE Transactions on Mobile Computing*, Vol. 3, No. 4, Oct-Dec. 2004, pp.366-379.

[Z1] "Z1 Datasheet", *Datasheet*. Available from: http://zolertia.sourceforge.net/wiki/images/e/e8/Z1_RevC_Datasheet.pdf.

[ZB2] "ZigBee IP Specification Overview", Descrição em: http://www.zigbee.org/Specifications/ZigBeeIP/Overview.aspx.

[ZigSE] "ZigBee Smart Energy FAQ", Descrição em: http://www.zigbee.org/Standards/ZigBeeSmartEnergy/FAQ.aspx.