



**Tiago José  
Carvalho Cantão**

**Proposta de Gateway para Redes de Sensores Sem  
Fios**





**Tiago José  
Carvalho Cantão**

**Proposta de Gateway para Redes de Sensores Sem  
Fios**

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Electrónica e Telecomunicações (Mestrado Integrado), realizada sob a orientação científica do Dr. Pedro Fonseca, Professor Auxiliar do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro, e coorientação do Dr. José Luís Azevedo, Professor Auxiliar do Departamento de Electrónica, Telecomunicações e Informática e do Eng. Manuel Loureiro da Exatronic- Engenharia Electrónica, LDA.



Aos meus pais e irmã, por tudo.



## **O júri**

Presidente	<b>Prof. Dr. Manuel Bernardo Salvador Cunha</b> Professor Auxiliar do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro
Vogal - Arguente Principal	<b>Prof. Dr. Francisco Manuel Madureira e Castro Vasques de Carvalho</b> Professor Associado da Faculdade de Engenharia da Universidade do Porto
Vogal - Orientador	<b>Prof. Dr. Pedro Nicolau Faria da Fonseca</b> Professor Auxiliar do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro
Vogal - Co-Orientador	<b>Prof. Dr. José Luís Costa Pinto de Azevedo</b> Professor Auxiliar do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro





## **Agradecimentos**

Agradeço ao Prof. Dr. Pedro Fonseca e ao Prof. Dr. José Luís Azevedo pelos conselhos, conhecimentos transmitidos e pelo à-vontade criado durante as nossas conversas.

Agradeço ao Eng. Manuel Loureiro pela disponibilidade, colaboração e sugestões, assim como aos restantes colaboradores e mestrandos, presentes na Exatronic, pelo saudável ambiente criado e pela interajuda.

Agradeço também aos meus pais e irmã pelo incentivo, apoio, confiança e preocupação demonstrados, durante todo o meu percurso académico e ao longo de toda a minha vida.

Agradeço, por fim, aos meus amigos e colegas, presentes nos momentos de trabalho e de lazer, pela amizade e camaradagem.



**Palavras-chave**

Redes de sensores sem fios, *gateway*, controlo do acesso ao meio, colisão de mensagens, sincronização.

**Resumo**

O presente trabalho apresenta um estudo acerca das redes de sensores sem fios e propõe uma solução para um *gateway* a ser integrado numa rede Zigbee. Para além da arquitetura deste dispositivo, sugerem-se soluções para a sua implementação, esquemas de troca de mensagens e de controlo do acesso ao meio, tendo em vista a eficiência energética e a garantia de entrega dos dados. Apresentam-se também resultados de simulações e protótipos de aplicações com vista a implementação do dispositivo e sua integração na rede.



**Keywords**

Wireless sensor networks, gateway, medium access control, message collision, synchronization.

**Abstract**

This work presents a study on wireless sensor networks and proposes a solution for a gateway to be integrated in a ZigBee network. In addition to the device architecture solutions for implementation, message exchange and medium access control schemes are also proposed; bearing in mind energetic efficiency and data delivery guarantee. Simulation results and application prototypes are also presented aiming the implementation of the device and its network integration.



# Índice

---

Índice.....	I
Índice de Figuras .....	III
Índice de Equações.....	V
Índice de Tabelas.....	VII
Acrónimos .....	IX
1. Introdução .....	1
1.1. Motivação .....	1
1.2. Objetivos.....	1
1.3. Estrutura.....	2
2. Redes de Sensores Sem Fios .....	3
2.1. Background histórico .....	3
2.2. Aplicações.....	6
2.3. Arquitetura .....	8
2.4. Protocolos de comunicação.....	16
2.5. Desafios e trabalho futuro.....	21
3. Sistema .....	23
3.1. Definição do sistema.....	23
3.2. Especificação de requisitos .....	25
3.3. Hardware.....	26
3.4. <i>Software</i> .....	30
3.5. Conclusão.....	34
4. Implementação .....	35
4.1. Arquitetura .....	35
4.2. Integração.....	36
4.3. Integração em rede.....	41
4.4. Conclusão.....	48
5. Soluções Propostas.....	49
5.1 Controlo do acesso ao meio .....	49
5.2. Firmware.....	56
5.3. Gestão da rede.....	58
6. Conclusões e Trabalho Futuro.....	61

Bibliografia .....	63
Anexo A .....	65
Anexo B .....	67
Anexo C .....	71



# Índice de Figuras

---

Figura 1 - Dois exemplos de telégrafo ótico .....	4
Figura 2 - Principais componentes de um nó de uma rede WSN.....	8
Figura 3 - Nó sensor SenseNode, da Genetlab, para sistemas de vigilância e nó sensor da família Mica, derivado de projetos da Universidade da Califórnia em Berkeley .....	10
Figura 4 - Topologia estrela .....	11
Figura 5 - Topologia malha.....	12
Figura 6 - Topologia híbrida .....	12
Figura 7 - <i>Stack</i> protocolar .....	13
Figura 8 - Princípio de funcionamento de um <i>gateway</i> .....	14
Figura 9 - Protocolos MAC.....	18
Figura 10 - Topologia da rede de sensores sem fios .....	23
Figura 11 - Esquema geral do <i>gateway</i> .....	24
Figura 12 - Interações <i>gateway</i> .....	25
Figura 13 - Placa de desenvolvimento FriendlyARM.....	26
Figura 14 - Módulos XBee.....	28
Figura 15 - Placas de desenvolvimento para os módulos XBee.....	29
Figura 16 - Diagrama do fluxo interno de dados do módulo XBee .....	29
Figura 17 - Modos de operação do módulo XBee.....	30
Figura 18 - Espaço de memória virtual .....	32
Figura 19 - Arquitetura Windows CE .....	33
Figura 20 - Elementos BSP .....	33
Figura 21 - Arquitetura física <i>gateway</i> .....	35
Figura 22 - Uso de <i>driver</i> para comunicação série.....	36
Figura 23 - Trama API .....	37
Figura 24 - Trama API TX Request .....	38
Figura 25 - Trama API RX Packet .....	38
Figura 26 - Exemplo de leitura de ficheiro através da Internet .....	39
Figura 27 - Ambiente Microsoft Visual Studio 2005 .....	40
Figura 28 - Interface X-CTU.....	41
Figura 29 - Etapas referentes a uma troca de mensagens.....	42
Figura 30 - Diagrama temporal de uma sequência de comunicação .....	44
Figura 31 - Probabilidade de colisão em função do período entre transmissões.....	45
Figura 32 - Probabilidade de colisões em função do tamanho da rede, para diferentes períodos entre transmissões .....	45
Figura 33 - Esquema do consumo de corrente do módulo XBee .....	46
Figura 34 - Consumo de corrente em função do período entre transmissões.....	47
Figura 35 - Esquema por pedido do <i>gateway</i> .....	50
Figura 36 - Esquema por iniciativa dos nós remotos .....	50
Figura 37 - Esquema de reserva temporal .....	51
Figura 38 - Limitações do período entre transmissões.....	52
Figura 39 - Limitações à capacidade da rede .....	52

Figura 40 - Influência da deriva do relógio.....	53
Figura 41 - Primeira comunicação .....	53
Figura 42 - Probabilidade de erro ao longo do tempo sem mecanismo de sincronização.....	54
Figura 43 - Esquema de sincronização.....	54
Figura 44 - Compensação da deriva .....	55
Figura 45 - Diagrama de <i>software</i> do <i>gateway</i> .....	56
Figura 46 - Diagrama de <i>software</i> dos nós remotos.....	57
Figura 47 - Associação de nós.....	58
Figura 48 - Tabela de nós.....	59

# Índice de Equações

---

Equação 1 .....	47
Equação 2 .....	47
Equação 3 .....	47
Equação 4 .....	47
Equação 5 .....	52
Equação 6 .....	55
Equação 7 .....	55
Equação 8 .....	59



# Índice de Tabelas

---

Tabela 1 .....	28
----------------	----



# Acrónimos

---

<b>API</b>	<i>Application Programming Interface</i>
<b>ARM</b>	<i>Advanced RISC Machines</i>
<b>BSP</b>	<i>Board Support Package</i>
<b>CCA</b>	<i>Clear Channel Assessment</i>
<b>CSMA</b>	<i>Carrier Sense Multiple Access</i>
<b>GSM</b>	<i>Global System for Mobile Communications</i>
<b>ISM</b>	<i>Industrial Scientific and Medical</i>
<b>MAC</b>	<i>Medium Access Control</i>
<b>SDK</b>	<i>Software Development Kit</i>
<b>TDMA</b>	<i>Time Division Multiple Access</i>
<b>WLAN</b>	<i>Wireless Local Area Network</i>
<b>WPAN</b>	<i>Wireless Personal Area Network</i>
<b>WSN</b>	<i>Wireless Sensor Network</i>





# 1. Introdução

---

## 1.1. Motivação

Hoje em dia, e cada vez mais, o uso de redes para monitorizar e controlar processos nas mais variadas aplicações está altamente disseminado. Entre estas, as redes de sensores sem fios, *Wireless Sensor Networks* (WSN), têm-se afirmado como a infraestrutura de comunicação de eleição em áreas como a monitorização ambiental, domótica, controlo de processos industriais, entre outros.

Contudo, se não se for capaz de fazer chegar a informação sem corrupção ao utilizador, e de acordo com as suas exigências temporais, muito do valor prometido por estas redes é perdido, o que pode reduzir o leque de aplicações possíveis.

Para tal é necessária a presença de um dispositivo na rede capaz de fazer a ligação entre esta e o utilizador final, de acordo com os requisitos pré-estabelecidos. Este dispositivo é referido como *gateway* e tem por função fazer a ponte entre dois meios de comunicação diferentes, quer a nível físico quer a nível protocolar. Dispositivos geralmente mais capazes que os restantes intervenientes da rede com funções de controlo e gestão da mesma.

É no contexto destas redes, e mais concretamente dos dispositivos previamente referidos, que este projeto de dissertação se enquadra. Em parceria com a Exatronic, empresa de Engenharia Eletrónica, pretende-se realizar uma avaliação relativa a estas redes e conceber um projeto para um *gateway* (*hardware* e *firmware*) bem como para soluções de controlo e gestão de rede, com vista a eficiência de informação e energética, passíveis de serem utilizadas em situações reais.

## 1.2. Objetivos

Com esta dissertação pretende-se descrever a conceção e desenvolvimento de um *gateway* para uma rede de sensores sem fios de acordo com uma série de requisitos.

Assim, os principais objetivos que se pretendam alcançar são:

- projeto e descrição da implementação de um *gateway* para uma rede de sensores sem fios
- projeto e descrição de uma solução protocolar para controlo e gestão da mesma rede

## Proposta de Gateway para Redes de Sensores Sem Fios

### 1.3. Estrutura

Esta dissertação está organizada em 6 capítulos cuja descrição sumária é a que se apresenta:

- Capítulo 1 - Introdução - Capítulo introdutório ao documento onde se descreve a motivação e os objetivos desta dissertação.
- Capítulo 2 - Redes de Sensores Sem Fios - É realizado um estudo do estado da arte no que diz respeito a redes de sensores sem fios. Vários tópicos são abordadas como a arquitetura, protocolos e *standards*, entre outros.
- Capítulo 3 - Sistema - Introdução ao sistema a projetar e aos requisitos que este deve cumprir.
- Capítulo 4 - Implementação - Capítulo referente à análise da implementação do *gateway*.
- Capítulo 5 - Soluções propostas - Análise das propostas referentes ao controlo e gestão da rede de sensores sem fios.
- Capítulo 6 - Conclusões e trabalho futuro - Apresentam-se as conclusões relativas ao trabalho desenvolvido e propõem-se caminhos de trabalho futuro.

## 2. Redes de Sensores Sem Fios

---

As redes de sensores sem fios, geralmente referidas como WSN, consistem numa rede constituída por dispositivos distribuídos espacialmente, dispositivos estes geralmente vocacionados para fazer a monitorização de condições físicas e ambientais.

Estes dispositivos, conhecidos como nós (*nodes*), colaboram através da rede sem fios para cumprir uma dada tarefa, interagindo com o espaço onde estão inseridos por meio de leitura e controlo de grandezas físicas. Sensores de temperatura, vibrações ou som, responsáveis por verificar as condições do meio, ajudam a rede a controlá-lo por intermédio de atuadores em servos, lâmpadas ou trincos.

Atualmente, as WSN são alvo de grande interesse e estudo pela capacidade que demonstram num grande número de aplicações, a custos mais baixos que aqueles de sistemas que fazem uso de cabos para alimentação e comunicação dos dispositivos constituintes. Estas redes assumiram, assim, um novo e importante papel em áreas como a domótica, a indústria ou mesmo a monitorização do ambiente.

Torna-se assim expectável uma forte aposta no desenvolvimento deste tipo de redes e da tecnologia subjacente aos seus constituintes, com o objetivo de se assumirem ainda mais, não só como alternativa aos sistemas atuais, mas como a norma no seu mercado.

### 2.1. Background histórico

#### 2.1.1. Comunicações sem fios

O desenvolvimento das primeiras redes de comunicações sem fios remontam a tempos antigos quando eram usados, por exemplo, sinais de fumo e fogo, foguetes, espelhos, entre outros, para permitir a passagem de mensagens entre pontos que se encontrassem em linha de vista [1].

Com a chegada da revolução industrial em meados do século XVIII, estas formas rudimentares de comunicação conheceram as suas sucessoras, começando pelas redes de telégrafos óticos. Neste sistema a informação era transmitida através de sinais visuais, gerados por intermédio de elementos mecânicos, como pás e traves, no topo de torres, Figura 1 [2].

## Proposta de Gateway para Redes de Sensores Sem Fios



Figura 1 - Dois exemplos de telégrafo ótico

Os diferentes sinais visuais gerados codificavam diferentes palavras e mensagens que eram passadas de estação em estação, as quais se encontravam, em média, separadas por uma distância de 10km, podendo nalguns casos chegar até aos 30km [3], e cujo ritmo de transmissão era condicionado principalmente pelas condições atmosféricas.

Este sistema foi precursor de um outro, que acabou por substituir estas linhas virtuais, o telégrafo elétrico. Em 1835, após já ter inventado um alfabeto elétrico, Samuel Morse apresenta o primeiro telégrafo comercial [3]. Estas linhas de telégrafo tinham custos mais reduzidos e conferiam maior segurança e privacidade que aquelas vistas anteriormente [2]. No entanto, por esta altura, dependiam de cabos para transmitir os sinais elétricos que compunham as mensagens. Apenas no final do mesmo século, Guglielmo Marconi demonstrou o funcionamento de um telégrafo sem fios marcando o nascimento do conceito de rádio. Em 1901, o mesmo dá um importante passo na história das comunicações sem fios e transmite a primeira mensagem, em código Morse, sobre o Oceano Atlântico [4].

Durante as décadas seguintes surgiram desenvolvimentos como a transmissão de voz via rádio e o conseqüente nascimento da telefonia móvel, com o primeiro sistema de telefone móvel, parcialmente automático e integrado em viaturas, a surgir em 1956 [3]. Desenvolvimentos estes, impulsionados pelas duas guerras mundiais que o mundo testemunharia. A necessidade de comunicação móvel e sem fios entre tropas aliadas originou a invenção do *walkie-talkie*, predecessor do telemóvel. Também a importância da deteção das forças inimigas levou ao nascimento do posicionamento via rádio.

Um pouco mais tarde, em 1979, é lançada no Japão a primeira rede comercial e automatizada de telemóvel, capaz de cobrir a cidade de Tokyo [5]. Nos anos seguintes, a cobertura estende-se a todo o país, tornando-se a primeira rede 1G de cobertura nacional a nível mundial [5]. Esta tendência foi seguida por diferentes países, apostando na criação de novas redes com novas capacidades, como o *roaming* internacional. Em 1991 realiza-se a primeira chamada utilizando o *standard GSM (Global System for Mobile Communications)* [6], cujas evoluções usamos ainda hoje em dia, marcando-se um importante passo para a disseminação das comunicações móveis a uma escala global.

Entretanto, surge também o conceito de WLAN (*Wireless Local Area Network*), associado ao nascimento da Internet. Com o objetivo de ligar dois ou mais dispositivos através de métodos de

## Proposta de Gateway para Redes de Sensores Sem Fios

distribuição sem fios, providenciando uma ligação via ponto de acesso à Internet, estas redes baseiam-se nos protocolos *standard* IEEE 802.11, criados pelo grupo de mesmo nome, em 1990 [4]. Comercializados sob a sigla Wi-Fi, os modems WLAN tornaram-se populares graças à facilidade de instalação e à crescente oferta de acessos sem fios para as pessoas, frequentemente de forma gratuita [7]. Os protocolos mais recentes suportam elevadas taxas de transmissão de dados, até 150Mbps/s, em ambas as bandas ISM usadas nestas redes, 2.4 GHz e 5 GHz [4].

Um dos últimos desenvolvimentos prende-se com as redes WPAN (*Wireless Personal Area Network*). Estas redes asseguram a comunicação entre variados dispositivos, com um alcance relativamente reduzido. Um exemplo é a tecnologia Bluetooth que foi usada como base para um novo *standard*, o IEEE 802.15, sobre o qual foi desenvolvido, por exemplo, o protocolo ZigBee (802.15.4), um dos *standards* de comunicação nas redes de sensores sem fios.

### 2.1.2. Redes de sensores

A evolução das redes de sensores esteve dependente de pesquisas e desenvolvimentos nas áreas de *sensing*, comunicação e computação. A partir do momento em que se visionou o uso integrado destas tecnologias, no que se viriam a chamar as redes de sensores, a evolução destas pode, essencialmente, ser caracterizada em quatro fases [8]:

#### Primeiros trabalhos em redes de sensores militares

No decorrer da Guerra Fria, um sistema de sensores acústicos de nome SOSUS (*Sound Surveillance System*), foi colocado no leito do oceano e permitia detetar e vigiar os submarinos Soviéticos. Este sistema ainda hoje é usado pela NOAA (*National Oceanographic and Atmospheric Administration*) para monitorizar eventos no oceano, como atividades sísmica e animal. Ainda durante a Guerra Fria, desenvolveram-se e instalaram-se redes de radares de defesa aérea.

#### Projecto DSN

A pesquisa moderna em redes de sensores sem fios começou por volta de 1980 com o programa DSN (*Distributed Sensor Networks*) da DARPA (*Defense Advanced Research Projects Agency*). Este programa pretendia averiguar se as estratégias, a nível de comunicação, usadas na *Arpanet* (antecessora da *Internet*) e baseadas no uso do protocolo TCP/IP, poderiam, de alguma forma, ser adaptadas para as redes de sensores. Um programa que encontrou entraves relacionados com o estado das tecnologias apontadas como essenciais para as redes de sensores sem fios tais como: sensores, comunicação (principalmente a nível protocolar), técnicas de processamento, algoritmos e *software* distribuído. Também se lançou o desenvolvimento de sistemas de operação, que permitissem um acesso flexível e transparente a recursos distribuídos, para redes tolerantes a falhas. Desenvolveram-se técnicas de processamento de sinal e algoritmos de *tracking* de múltiplos alvos, demonstrados em testes de tempo real, para deteção de aviões a baixas altitudes num ambiente distribuído constituído por sistemas de sensores acústicos. Todas estas pesquisas levaram ao desenvolvimento de novos equipamentos, necessários para fazer a integração das redes ao nível pretendido.

## Proposta de Gateway para Redes de Sensores Sem Fios

### Redes de sensores militares nas décadas de 80 e 90

Apesar da tecnologia para produzir os pequenos sensores, visionados pelos diferentes colaboradores no desenvolvimento destas redes, ainda não estar preparada, os sistemas militares continuaram a reconhecer o potencial destas. Tornaram-se um componente essencial da visão de ligação em rede do campo de batalha, com cooperação, sobre uma rede de comunicação, para melhor eficácia. Um exemplo deste tipo de integração foi o CEC (*Cooperative Engagement Capability*), da Marinha Americana, que consistia de múltiplos radares para vigilância aérea.

### Pesquisa em redes de sensores no século XXI

As novas pesquisas apontam num caminho do desenvolvimento de sensores pequenos e baratos, baseados em tecnologia MEMS (*Microelectromechanical Systems*), com integração em redes sem fios e dotados de processadores de baixo consumo e custo. Um recente programa da agência DARPA, o SensIT (*Sensor Information Technology*), procurou desenvolver novas técnicas de *networking* e de processamento de informação da rede. Estas redes SensIT apresentam novas capacidades de interatividade e programabilidade, com distribuição de tarefas e consultas dinâmicas. Com características como *multitasking* e algoritmos para maior precisão em deteção e *tracking*, aliados ao *software*, os resultantes deste programa, permitem baixa latência, alta eficiência energética, autonomia e resistência, ao mesmo tempo que se torna mais difícil detetar o seu funcionamento.

## 2.2. Aplicações

Os sensores podem ser usados para detetar ou monitorizar uma grande variedade de parâmetros físicos ou condições do meio onde se encontram, como por exemplo [9]:

- Luz
- Som
- Humidade
- Pressão
- Temperatura
- Composição do solo
- Qualidade da água e solo
- Atributos de um objeto como tamanho, peso, posição, velocidade e direção.

As WSN aliam estas características à possibilidade de criarmos uma rede destes sensores capazes de comunicar entre si e sem necessidade de fonte de energia por cabo, permitindo a sua implantação, por exemplo, em ambientes inhóspitos, como os campos de batalha ou incêndios, onde as redes com fios são impensáveis. Fazem-no também a custos mais reduzidos e com instalação mais rápida, prática e eficiente. Torna-se, assim, simples verificar que existe um grande número de aplicações onde as WSN podem substituir, e tornar-se mais competentes e fiáveis, que as redes de sensores com fios e ainda imaginar novas aplicações possibilitadas por esta tecnologia.

## **Proposta de Gateway para Redes de Sensores Sem Fios**

### **Aplicações militares**

Como foi abordado anteriormente, o desenvolvimento das WSN está intimamente ligado a programas desenvolvidos e/ou financiados pelos diferentes ramos de forças militares. O uso destes sensores permitiu o aparecimento e modernização de sistemas, principalmente vocacionados a deteção de inimigos no campo de batalha, desde grandes redes para deteção de submarinos a sistemas portáteis, instalados em veículos, que permitem a deteção do posicionamento de atiradores furtivos em caso de ataque.

### **Cenários de catástrofe**

Um exemplo deste tipo de aplicação é o caso dos incêndios florestais. Largar sensores sobre uma zona afetada, com recurso a avião, por exemplo, equipados com termómetros e capazes de determinar a sua posição (relativa entre si ou em coordenadas absolutas) para se obter um mapa de calor do incêndio [10]. Também existe um grande potencial para integrar estas redes em sistemas de previsão de cheias [11].

### **Edifícios inteligentes**

A integração das redes de sensores sem fios nos sistemas usados num edifício, para controlar certos parâmetros ambientais (temperatura ou humidade, por exemplo), com o objetivo de proporcionar conforto aos residentes ou trabalhadores, permite criar uma rede inteligente de sensores e atuadores que possibilite uma ação sobre o ambiente mais eficaz e com enormes poupanças em recursos e respetivos custos. Outra perspetiva de utilização em edifícios é na monitorização da estrutura dos mesmos, desde o controlo da fadiga mecânica ao longo do tempo à análise do estado da estrutura no caso de terremotos [10].

### **Indústria**

O uso de redes de sensores a nível industrial não é algo de novo. Processos de leitura de parâmetros e controlo, automação de edifícios e linhas de produção ou controlo de acessos, são algumas das aplicações conhecidas [11]. No entanto, são soluções que usam cabos e que apresentam custos elevados e difícil implementação. As WSN tornam-se assim, uma alternativa promissora para este tipo de sistemas pela sua facilidade de implementação e distribuição, custos mais reduzidos e elevada precisão. Uma das utilizações prende-se, por exemplo, com a manutenção de maquinaria. Sensores fáceis de aplicar e sem fios são ideais para colocação em massa e em lugares de difícil acesso. No entanto, para serem vantajosos, estes sistemas não podem interferir com o normal funcionamento do equipamento existente, e vice-versa, (ruído eletromagnético gerado por um e outro, por exemplo), e devem ter elevadas capacidades de bateria (e fazer um uso eficiente destas) ou eficazes sistemas de recolha de energia.

### **Medicina e saúde**

O uso de redes de sensores inteligentes para aplicações biomédicas tem-se tornado cada vez mais comum, como é o exemplo de um projeto, do Departamento de Energia Americano, em retinas artificiais [11]. Outras possibilidades na área da saúde dizem respeito a interfaces para pessoas incapacitadas, monitorização integrada de pacientes, técnicas de diagnóstico, administração de fármacos, monitorização e *tracking* de pacientes e pessoal hospitalar [11].

### Outras

Para além destas aplicações, também tem surgido algum interesse no uso destas redes na monitorização do ambiente, no que diz respeito a poluentes, por exemplo. Também no mapeamento da biodiversidade esta tecnologia se mostra bastante capaz pela capacidade de monitorizar de forma independente e pouco intrusiva, durante um longo período de tempo, um dado *habitat* [10]. A agricultura é um sector que pode tirar grandes proveitos desta tecnologia com a introdução de irrigação e fertilização, de alta precisão, controladas por redes de sensores sem fios. Na logística de mercadorias, em armazéns ou lojas, dispositivos que identifiquem os bens e permitam facilmente determinar a localização dos mesmos podem ser vantajosos.

## 2.3. Arquitetura

### 2.3.1. Nó individual

Os nós constituintes das redes de sensores sem fios podem apresentar, dependendo das exigências das várias aplicações, diferentes características no que diz respeito ao seu tamanho, peso ou consumo energético. No entanto, de uma forma geral, estes devem ser constituídos por cinco componentes essenciais [10]:

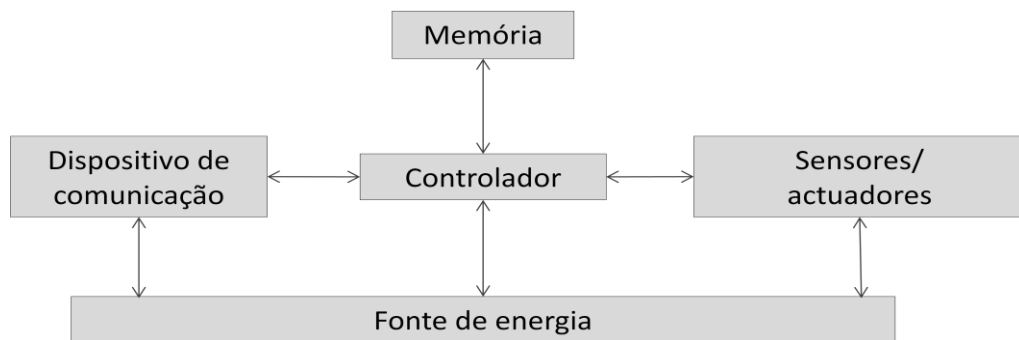


Figura 2 - Principais componentes de um nó de uma rede WSN

### Controlador

O controlador pode ser considerado o núcleo do nó e é responsável pelo processamento dos dados enviados pelos sensores, ou por outros nós, e de todas as rotinas necessárias ao funcionamento deste. É responsável pelo controlo dos períodos de amostragem, os actuadores e a comunicação.

Existem variadas soluções, no que toca a unidades de processamento, capazes de desempenhar este tipo de tarefas, desde processadores gerais, como aqueles usados nos computadores pessoais, a FPGAs (*Field-Programmable Gate Arrays*) e ASICs (*Application-Specific Integrated Circuits*). No entanto estes apresentam desvantagens como o elevado consumo energético, impraticabilidade de reprogramação ou inflexibilidade aplicacional [10]. A solução mais comum é o uso de microcontroladores. Estes apresentam grande flexibilidade em ligar e interagir com outros dispositivos, são capazes de processamento de sinal com restrições de tempo e possuem memória integrada. Outro ponto a favor destas unidades de processamento é o baixo consumo energético aliado, ainda, à possibilidade de se poder colocar o dispositivo em estado *sleep*



## Proposta de Gateway para Redes de Sensores Sem Fios

(numa tradução à letra, adormecido), mantendo este apenas algumas partes e funcionalidades ativas.

Uma das gamas de microcontroladores mais usadas recentemente, vista como detentora de um grande potencial, é a MSP 430 da *Texas Instrument's* [12], que possui características como CPU de 16 *bits*, frequência de funcionamento até 25 *GHz*, até 16KB de memória RAM, oscilador interno ou ADCs de 10/12/14 e 16 *bits* [13].

### Memória

O bloco de memória é necessário para guardar as leituras dos sensores, pacotes de dados de outros nós ou o código dos programas que regem o funcionamento do dispositivo. Um dos tipos de memória usados é a memória RAM (*Random Access Memory*), que é rápida mas volátil (perde os conteúdos quando não alimentada), sendo assim usada principalmente como armazém de informação temporária de leitura e pacotes, por exemplo. No caso de informação crítica como são os programas, esta deve ser armazenada em memórias ROM (*Read-Only Memory*), ou mais tipicamente, EEPROM (*Electrically Erasable Programmable Read-Only Memory*) e memória *flash* [10].

### Dispositivo de comunicação

O dispositivo de comunicação sem fios é, obviamente, essencial para a constituição da rede de sensores, permitindo a troca de informação entre nós, criando redes. O meio de transmissão mais usado e o que reúne maior consenso é a comunicação por Rádio Frequência. Possui um alcance e taxas de transmissão relativamente elevados, taxas de erros aceitáveis com consumos de energia razoáveis e não necessita de linha de vista entre transmissor e recetor [10]. Para além deste meio são também usados, por exemplo, comunicações óticas e por ultra sons.

Os dispositivos geralmente usados para estabelecer esta comunicação são conhecidos como *transceivers*, pois combinam as tarefas de um transmissor (*transmitter*) e de um recetor (*receiver*), transformando um *stream* de *bits* em ondas rádio e vice-versa. Existe uma grande oferta deste tipo de dispositivos e portanto, eles devem ser escolhidos para cada situação tendo em conta as suas características energéticas, frequências de transmissão, modulações, fator de ruído, entre outros. Estes dispositivos, como no caso dos microcontroladores, possuem diferentes estados para potenciar a poupança de energia.

### Sensores/atuadores

Estes componentes permitem a interação com o meio onde o dispositivo se encontra através da leitura e controlo de parâmetros físicos. Os sensores podem ser divididos em duas categorias [10]:

- **Sensores passivos:** Estes sensores realizam as suas medições sem qualquer influência no meio em que estão inseridos. Além disso, podem também ser passivos no sentido energético, retirando a energia necessária ao seu funcionamento do ambiente.
- **Sensores ativos:** Sensores que sondam de forma ativa o ambiente à sua volta, como o sonar, radares ou os sensores sísmicos, que sondam a constituição da crosta terrestre através de pequenas explosões.

## Proposta de Gateway para Redes de Sensores Sem Fios

Estes podem ainda ser divididos noutras duas categorias [10]:

- **Omnidirecionais:** As suas medições não estão focadas num ponto, plano ou direção (são exemplos o termómetro, sensores de luz, vibração ou humidade).
- **Narrow-beam:** Este tipo de dispositivos, também passivos, possuem algum tipo de diretividade nas suas medições. É um exemplo, uma câmara fotográfica que tira amostras segundo uma dada direção;

Os atuadores são, regra geral, dispositivos simples que controlam a abertura ou fecho de um *switch* ou de um *relay* ou que definem o valor de uma dada variável. Desta forma, pode-se fazer o controlo de um motor, lâmpadas ou diferentes máquinas.

### Fonte de energia

Este é um dos componentes mais importantes da constituição de um nó de uma rede de sensores sem fios. Por forma a manter a desejada mobilidade dos dispositivos, geralmente esta fonte é constituída por baterias. Para além de alimentação de dispositivos e armazenamento de energia eficientes, ou o aumento de capacidade, também se têm feito desenvolvimentos no uso de técnicas de colheita de energia do ambiente (por uso de painéis solares, coletores piezoelétricos ou por aproveitamento de gradientes de temperatura), com o objetivo de prolongar, durante o maior período de tempo possível, a autonomia do equipamento.

Para além dos aspetos relativos ao *hardware* dos nós sensores, importa também salientar alguns pontos, no que diz respeito, ao *software* destes. Os sistemas operativos tradicionais são responsáveis por controlar, proteger o acesso a recursos e fazer a gestão da alocação destes recursos aos diferentes utilizadores, assim como suportar a execução concorrente de diversos processos e comunicação entre estes [10]. No entanto, é necessário ter em conta as capacidades dos componentes vistos anteriormente, que constituem uma camada física que não está à altura deste tipo de sistemas de operação. No caso dos nós constituintes de uma rede de sensores sem fios, os sistemas de operação ou ambientes de execução devem proporcionar eficiência energética na execução, gestão fácil e eficaz de componentes externos e de informação que se torna disponível de forma assíncrona. É claro que se exige um compromisso entre os modelos de programação, a estrutura do *stack* de protocolos, suporte para gestão de energia e as capacidades dos recursos do sistema.

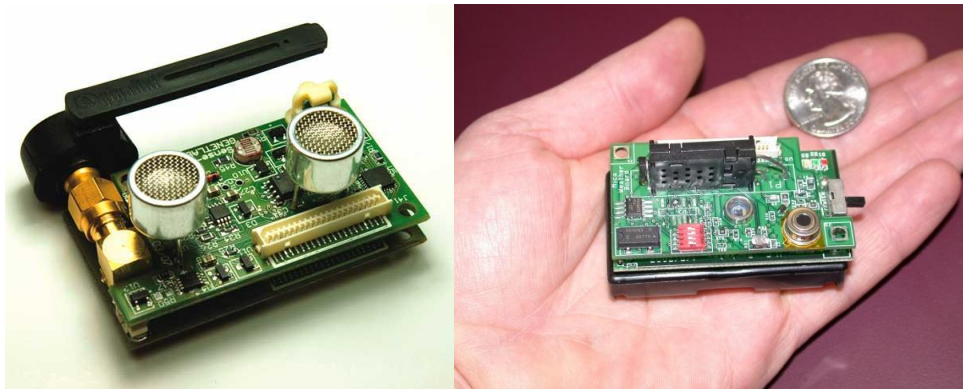


Figura 3 - Nó sensor SenseNode, da Genetlab, para sistemas de vigilância e nó sensor da família Mica, derivado de projetos da Universidade da Califórnia em Berkeley

## Proposta de Gateway para Redes de Sensores Sem Fios

### 2.3.2. Redes de sensores sem fios

Após a apresentação das principais características dos nós constituintes das WSN, importa agora discutir alguns dos princípios relativos à arquitetura que rege o desenho deste tipo de redes.

Existe uma variedade de topologias padrão para redes de comunicação rádio que estabelecem diferentes *layouts* para interconexão dos vários elementos de uma rede. Estes *layouts* pretendem cobrir diferentes situações, dependendo dos objetivos do utilizador, podendo este preferir, por exemplo, qualidade na troca de informação a favor de menores gastos energéticos. No que às redes de sensores sem fios diz respeito, são três as principais topologias associadas [14]:

#### Rede em estrela

Nesta topologia existe uma estação base que pode comunicar com vários nós mas estes não podem comunicar entre si. Este tipo de desenho é simples e permite baixos consumos de energia e reduzida latência na troca de mensagens, visto que apenas se realizam comunicações diretas entre dois pontos próximos. No entanto, necessita que a estação base esteja ao alcance de todos os nós e apresenta pouca robustez, comparada com outras topologias, devido à dependência de um único nó para ser feita a gestão da rede.

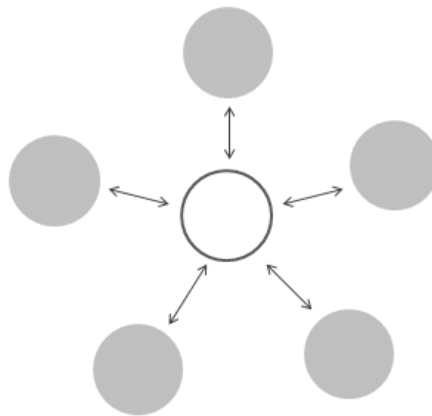


Figura 4 - Topologia estrela

#### Rede em malha

Nestas redes qualquer nó constituinte pode transmitir para qualquer outro nó que se encontre no seu alcance. Permitem-se, desta forma, as comunicações *multi-hop*, ou seja, o uso de nós que se encontrem dentro de alcance, como pontes, para transmitir informação para outros nós que não estejam. Esta topologia apresenta, assim, fácil escalabilidade, fiabilidade e redundância. Se um dos nós falhar os restantes podem ainda comunicar uns com os outros diretamente ou através de intermediários. O conceito de *multi-hop* permite, também, a fácil extensão do alcance da rede e evitar obstáculos. Como desvantagens deste desenho e principalmente das comunicações *multi-hop* apontam-se o maior consumo energético dos dispositivos, elevadas latências e o significativo aumento do tamanho das tramas a transmitir para o *routing* efetivo de mensagens entre os nós.

## Proposta de Gateway para Redes de Sensores Sem Fios

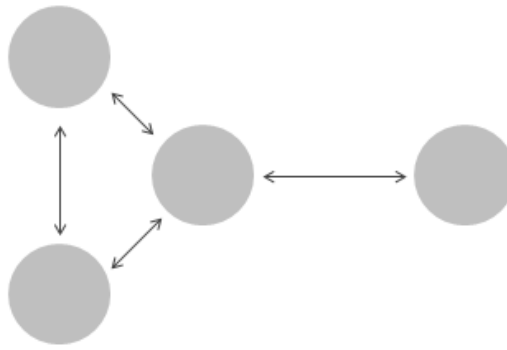


Figura 5 - Topologia malha

### Rede híbrida estrela-malha

As redes deste tipo apresentam-se como um híbrido das topologias estrela e malha, Figura 6 [14], tentando retirar as melhores características de cada uma delas. Desta junção resulta uma rede de comunicações robusta e versátil que, ao mesmo tempo, consegue manter um consumo energético relativamente baixo. Na sua constituição existem nós com características de *multi-hop*, que apresentam elevados consumos de energia, devendo, se possível, ser ligados a linhas de alimentação e outros, sem esta capacidade de encaminhamento de mensagens, com consumos reduzidos.

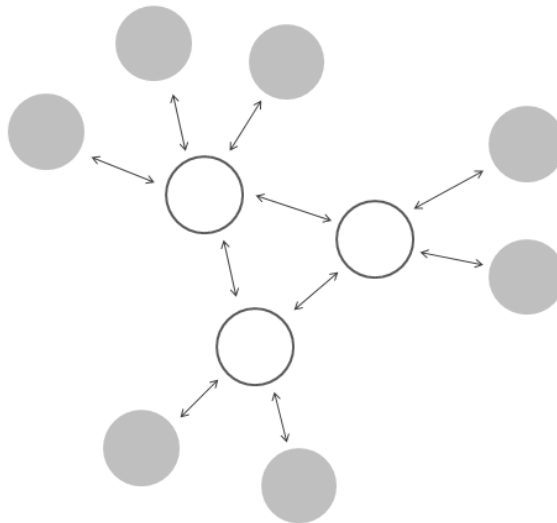


Figura 6 - Topologia híbrida

As redes de sensores sem fios consistem portanto, num elevado número de nós sensores e um ou mais *sinks* (dispositivo recetor de dados) /estações base, distribuídos numa dada região, de acordo com uma das topologias vistas ou alguma variação destas. Os *sinks* de informação enviam pedidos ou comandos aos nós sensores, implementados na região de interesse, enquanto estes colaboram para desempenhar a tarefa atribuída e enviar os dados de leituras do meio para o *sink*. No caso de uma rede *multi-hop*, os nós sensores terão a dupla tarefa de transmitir os pacotes referentes à própria informação e reencaminhar pacotes de informação de outros nós até à estação base. Esta estação pode comunicar com um gestor de tarefas ou utilizador via Internet, satélite ou qualquer tipo de rede sem fios (Wi-Fi, WiMAX). O dispositivo desempenha uma função de

## Proposta de Gateway para Redes de Sensores Sem Fios

*gateway*, servindo de intermediário entre as duas redes, situação que será analisada em mais pormenor a seguir.

No que diz respeito ao *stack* protocolar, este combina características de *routing* e controlo energético, integra dados com protocolos de rede, comunicação com eficiência energética através do meio e promove cooperação entre os nós sensores [11]. O esquema desta pilha protocolar encontra-se representado na Figura 7 [9] e apresenta as camadas física, de *link* de dados, de rede, de transporte e de aplicação. Por outro lado, o *stack* pode ser dividido num grupo de planos de gestão ao longo de cada camada. Estes planos de energia, mobilidade e tarefa monitorizam a energia, movimento e distribuição de tarefas entre os nós sensores, ajudando-os a coordenar a tarefa de *sensing* e diminuir o consumo total de energia [11].

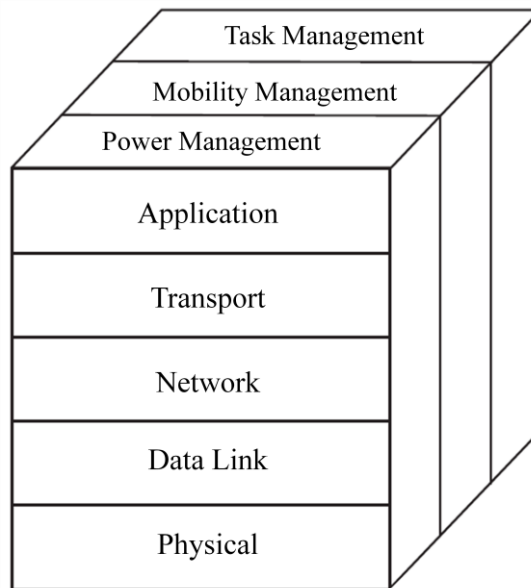


Figura 7 - Stack protocolar

### Physical Layer

A camada física é responsável por converter o *bit stream* da camada superior em sinais apropriados à transmissão no meio. Para tal trata da seleção da frequência, da geração da frequência da portadora, deteção de sinal, modulação e encriptação de dados.

### Data Link Layer

Esta camada é responsável pelo *multiplexing* dos *streams* de dados, criação e deteção dos frames de dados, acesso ao meio e controlo de erros por forma a permitir transmissões ponto-a-ponto e ponto-multiponto fidedignas [9]. Uma das funções mais importantes desta camada é o controlo do acesso ao meio (MAC) cujo objetivo é partilhar de forma justa e eficiente os recursos de comunicação ou meio pelos múltiplos nós sensores por forma a obter-se uma boa performance da rede em termos de consumo de energia, taxas de transferência e latência.

### Network Layer

Responsável pelo *routing* dos dados dos nós sensores para os *sinks* de informação. É desenhada de acordo com alguns princípios [11]: a eficiência energética; as redes de sensores são maioritariamente centradas em dados; os nós de encaminhamento, para além das tarefas de *routing*,

## Proposta de Gateway para Redes de Sensores Sem Fios

podem agregar dados de múltiplos vizinhos devido a tarefas de processamento local; os nós destas redes podem não ter um ID único, devido ao seu elevado número, sendo necessário serem referenciados com base nos seus dados ou localização.

### Transport Layer

É responsável pela entrega de dados *end-to-end* fiável entre os nós sensores e o *sink*, ou *sinks*.

### Application Layer

Como o nome sugere esta camada inclui as principais aplicações assim como várias funcionalidades de gestão. Protocolos responsáveis por *query-dissemination*, localização de nós, sincronização temporal e segurança da rede.

#### 2.3.3. Gateways

Tendo em conta o que foi dito, principalmente aquando da enumeração de aplicações, é possível perceber o potencial das redes de sensores sem fios em recolher e transportar dados de uma forma eficiente, acerca de uma ou várias grandezas físicas de um certo ponto ou região de um meio. No entanto, nem sempre é do interesse do projetista manter os dados no ambiente em que a rede está inserida, mas sim comunicá-los a um utilizador, ou mesmo a um outro sistema, que se pode encontrar a grandes distâncias e pretende aceder a estes através de vários dispositivos. É neste contexto que se percebe a importância de um dispositivo que estabeleça uma ligação física e protocolar entre a rede de sensores sem fios e uma rede exterior, por exemplo a Internet. Este dispositivo, conhecido como *gateway* (termo usado para definir o nó de uma rede que estabelece uma *interface* com outra rede que use protocolos diferentes), reúne dados dos nós sensores, podendo processá-los de alguma forma e encaminha a informação relevante através da rede desejada até ao dispositivo do utilizador final, como mostra a seguinte figura [10].

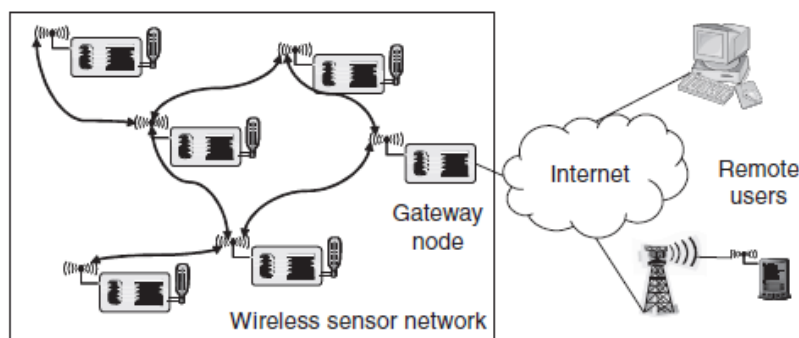


Figura 8 - Princípio de funcionamento de um gateway

Obviamente, levanta-se um grande número de dificuldades que o *gateway* deve ultrapassar. Este tem de ser responsável pela ligação entre dois meios diferentes, que muito provavelmente não partilham características das respetivas camadas do *stack* protocolar. Isto implica alterações de meio e frequência de transmissão, modulação, protocolos de transporte de informação e mapeamento de rede, entre outros.

## Proposta de Gateway para Redes de Sensores Sem Fios

No caso de uma comunicação entre uma WSN e a Internet, por exemplo, para além de servir de interface entre duas camadas físicas diferentes, o *gateway* é responsável por traduzir uma mensagem de notificação de uma rede de sensores sem fios numa mensagem de aplicação de Internet e do encaminhamento para um dado endereço IP.

### 2.3.4. Gestão de energia

Como tem sido discutido ao longo do documento, a gestão e a poupança de energia são pontos importantíssimos no que diz respeito às redes de sensores sem fios. Devem ter-se em conta dois aspetos essenciais por forma a entenderem-se os obstáculos a enfrentar: o armazenamento de energia e a sua provisão de forma eficiente, de acordo com as necessidades, e o reabastecimento da energia consumida, através de algum mecanismo de recolha do ambiente. Se as baterias dos nós constituintes de uma WSN tiverem de ser substituídas em períodos reduzidos de tempo, estas tornam-se pouco práticas e aumentam os custos de manutenção, o que dificulta a sua disseminação. Desta forma, os nós sensores devem ser desenhados para a apresentar o menor consumo de energia e, se possível, recorrerem-se do meio em que estão inseridos para recolher, pelo menos, parte da energia necessária.

As principais tarefas de um nó sensor de uma rede de sensores sem fios são: recolher dados sobre eventos, aplicar algum processamento local aos dados e transmiti-los. O seu consumo de energia pode então ser dividido em três principais áreas: escuta do meio, processamento de dados e comunicação, realizados pelos sensores, CPU e o *transceiver* rádio, respetivamente [11].

A energia necessária para fazer a leitura do meio varia com a natureza da aplicação e dos sensores usados para os diferentes casos. Para além das várias soluções de *hardware*, que por si só podem determinar grandes diferenças no que diz respeito ao consumo, outras estratégias de redução de energia destes componentes passam por [14]:

- Ligar a alimentação do sensor apenas aquando da amostragem
- Ligar a alimentação ao circuito de acondicionamento de sinal apenas quando se estiver a amostrar o sensor
- Amostrar o sensor apenas no caso de ocorrência de evento
- Baixar o ritmo de amostragem para o mínimo requerido pela aplicação.

Os gastos de energia em processamento são semelhantes àqueles de escuta do meio [11]. No entanto, a computação requer muito menos energia que a comunicação de dados. Por exemplo, de acordo com o modelo de Rayleigh de atenuação, a transmissão de um pacote de 1 kb de informação ao longo de 100 m é aproximadamente equivalente a executar 3 milhões de instruções num microprocessador típico [15]. Isto demonstra a importância do processamento local de dados, por forma a minimizar o consumo de energia numa rede *multi-hop* de sensores [11]. Mecanismos de gestão de energia por aplicação de diferentes modos de funcionamento (*active*, *idle* e *sleep*) são muito relevantes para a redução do consumo de operação dos microcontroladores. Neste modos, diferentes funcionalidades dos componentes poderão estar desligadas de acordo com os desejos do utilizador.

O *transceiver* gasta quantidades semelhantes de energia a transmitir, receber e no estado *idle* [11]. Para uma redução significativa do consumo energético devem ser aplicadas técnicas como [14]:

## Proposta de Gateway para Redes de Sensores Sem Fios

- Redução da quantidade de informação transmitida por compressão e redução de dados
- Diminuição do *duty cycle* do *transceiver* (tempo de funcionamento) e da frequência da transmissão
- Redução do *frame overhead* (*bits* adicionados a um sinal digital que providenciam funções de rede)
- Implementação de mecanismos rigorosos de gestão de energia (modos *power-down* e *sleep*)
- Implementação de uma estratégia de transmissão baseada em eventos (apenas transmite dados quando um evento de leitura do sensor ocorrer).

Como foi dito, para além destas técnicas de redução do consumo energético é também muito importante a integração de técnicas de recolha de energia a partir do ambiente. Alia-se desta forma o aumento da longevidade e a redução de manutenção das redes, permitindo que estas se tornem cada vez mais independentes, fiáveis, eficazes e, ao mesmo tempo, mais baratas. No entanto, uma autonomia total de baterias raramente é possível devido às pequenas quantidades de energia que estes sistemas, regra geral, conseguem retirar do meio, variações desta energia ao longo do tempo e imprevisibilidade dos atuais e principais métodos de recolha. Entre os métodos mais utilizados e promissores encontram-se os painéis fotovoltaicos, sistemas que retiram energia através de gradientes de temperatura e pressão, vibrações, correntes de ar ou fluidos.

## 2.4. Protocolos de comunicação

### 2.4.1. Camada física

A camada física das redes de sensores sem fios, já discutida, diz respeito a funções e componentes, de um nó sensor, que medeiam entre a transmissão e receção de formas de onda e o processamento de dados digitais no resto do nó, incluindo o processamento de protocolos de mais alto nível [10]. Esta camada, preocupa-se principalmente com a modulação e desmodulação de dados digitais. O desafio é encontrar esquemas de modulação e arquiteturas para *tranceivers* que sejam simples, de baixo custo e ainda assim robustos o suficiente para desempenhar as tarefas exigidas ao nó.

A comunicação *wireless* pode ser dispendiosa em termos energéticos e complexa em implementação. Ao desenhar a camada física das redes de sensores a minimização do consumo energético assume uma importância significativa, entre outros pormenores relacionados com atenuações, reflexões ou difrações das ondas de transmissão [11]. Como foi dito, o meio de comunicação utilizado para estas redes de sensores sem fios é a comunicação por ondas eletromagnéticas transmitidas em bandas de Rádio Frequência, bandas estas que vão dos 3Hz aos 300 GHz [11]. Num sistema sem fios baseado em rádio frequência, a frequência da portadora e a banda de funcionamento devem ser cuidadosamente escolhidas. Estas definem as características de propagação como a capacidade das ondas em ultrapassar obstáculos e características do canal de comunicação.

A maioria dos sistemas atuais baseados em Rádio Frequência trabalham a frequências abaixo dos 6 GHz [10]. A disponibilidade de frequências e bandas de frequência rádio está sujeita a



## Proposta de Gateway para Redes de Sensores Sem Fios

regulação para evitar interferências indesejadas entre os diferentes utilizadores e sistemas. Existem também bandas livres de licença sujeitas apenas a restrições a nível de potência de transmissão, densidade espectral de potência ou *duty cycle*. Deste tipo de bandas, as mais conhecidas dizem respeito às bandas ISM (*Industrial Scientific and Medical*). Estas bandas são, obviamente, populares e servem como ponto de partida para alguns dos *standards* de comunicação, que se apresentam numa secção seguinte.

### 2.4.2. Protocolos MAC

No caso das redes de sensores sem fios os protocolos MAC tratam da tarefa de regular o acesso dos nós a um meio de comunicação partilhado de forma a que certos requisitos de desempenho sejam cumpridos. Nestas redes o requisito mais importante, que estes protocolos devem procurar preencher, prende-se com a eficiência energética. Causas para perdas energéticas relacionadas com os protocolos desta camada podem dever-se a colisões de pacotes ou *overhearing*, que irão ser vistas em mais pormenor de seguida.

Pesquisas e desenvolvimentos em protocolos MAC remontam a mais de 30 anos, no entanto, as redes de sensores sem fios apresentam algumas características únicas que devem ser tidas em conta nas propostas de protocolos de controlo do acesso ao meio [9]:

- Estas redes consistem geralmente num grande número de nós sensores densamente distribuídos por uma área geográfica. Este número de sensores pode ser várias vezes superior àqueles apresentados por redes sem fios convencionais.
- Os nós sensores são geralmente alimentados por baterias que têm capacidades limitadas. Por vezes é difícil ou mesmo impossível trocar ou recarregar estas baterias, limitando a vida dos nós e, conseqüentemente, da rede.
- Os nós sensores devem poder ser implementados sem planeamento prévio. Desta forma, têm de ser capazes de se organizar entre si e criar uma rede de comunicação funcional.
- A topologia de uma WSN muda mais frequentemente devido a falhas nos nós ou a deslocalizações destes.
- Os nós sensores apresentam capacidades computacionais e de memória limitadas.

Desta forma, para além da função básica de arbitrar os acessos a um meio partilhado para evitar conflitos entre os diferentes nós, estes devem também ter em conta outros fatores para melhorar a performance da rede e dos serviços que esta presta a diferentes aplicações. Assim, nas redes de sensores sem fios, estes fatores são principalmente [9]:

- Eficiência energética - Como tem sido visto esta deve ser uma das maiores preocupações de todas as camadas protocolares.
- Escalabilidade - Capacidade de acomodar alterações no tamanho da rede.
- Adaptabilidade - Capacidade de acomodar alterações na densidade de implementação ou topologia de rede.
- Utilização de canal - Largura de banda usada para uma comunicação eficaz.
- Latência - Atraso entre o envio de um pacote até ao recetor o receber com sucesso.
- Débito - Quantidade de dados transferidos com sucesso entre um emissor e um recetor num dado período temporal.

## Proposta de Gateway para Redes de Sensores Sem Fios

- Equidade - Capacidade de vários nós partilharem um canal de transmissão de forma justa e semelhante.

Destas métricas, aquela que se mostra como de maior importância é a eficiência energética. Assim, os protocolos MAC devem ser projetados para cumprir certas exigências que levam a uma melhor utilização dos recursos, com o consumo de energia mais baixo possível. Como já foi discutido a comunicação é a principal responsável pelo consumo dos recursos energéticos de um nó e deve, desta forma, ser coordenada cuidadosamente para garantir uma operação eficiente da rede. Responsável por esta coordenação, o protocolo MAC deve lidar com as principais causas de consumo e desperdício energético associadas à comunicação [9]:

- Colisões - As colisões ocorrem quando dois ou mais nós transmitem os seus pacotes ao mesmo tempo. Em resultado disso os pacotes podem-se sobrepor e ser corrompidos tendo de ser descartados, principalmente se os emissores se encontrarem próximos. Retransmissões destes pacotes aumentam o consumo de energia e a latência da rede.
- *Overhearing* - Esta situação acontece quando um nó recebe pacotes destinados a outros nós da rede. A 'escuta' destes pacotes resulta em gastos desnecessários de energia, que são ainda mais significativos se o débito e a densidade da rede forem elevados.
- *Idle listening* - Este conceito refere-se aos casos em que o rádio é operado e não é recuperada informação útil do canal de comunicação. A principal causa deste problema, como o nome indica, é deixar a interface rádio ligada durante períodos em modo *idle*, em que não ocorrem eventos.
- *Overhead* de controlo - Um protocolo MAC requer a receção, envio e 'escuta' de certos pacotes ou *bits* de controlo, o que também contribui para o consumo de energia, sem que se estejam a enviar dados.

Tendo em conta todos estes pressupostos, os protocolos MAC para as redes de sensores sem fios podem ser divididos em três principais tipos. A Figura 9 [11] apresenta-os e mostra os diferentes protocolos propostos em cada categoria.

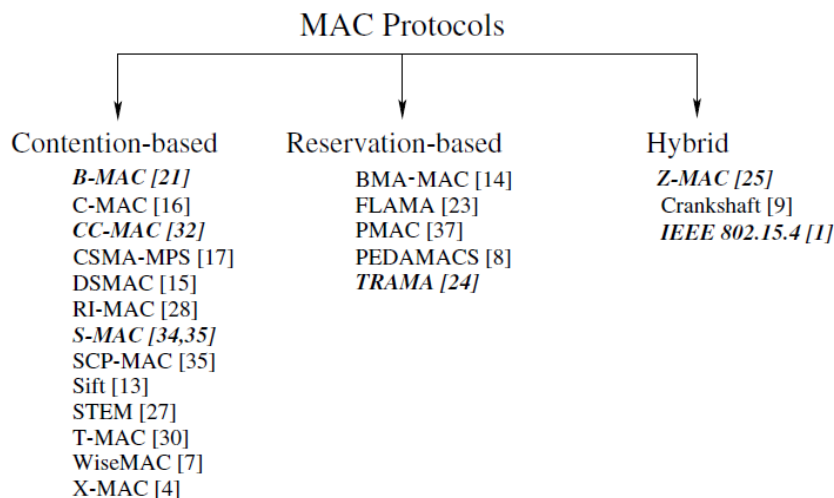


Figura 9 - Protocolos MAC

## Proposta de Gateway para Redes de Sensores Sem Fios

### Baseados em competição

Estes protocolos dependem da existência de uma contenda entre nós para estabelecer canais de comunicação, tomando cada um dos nós uma decisão em comunicar, ou não, após escuta do meio. Providenciam flexibilidade dado que cada nó pode tomar independentemente decisões de contenção sem a necessidade de troca de mensagens. Como resultado, estes protocolos geralmente não necessitam de infraestrutura, o que é importante nas redes de sensores sem fios. Ao invés, cada nó tenta o acesso ao canal baseado no mecanismo *carrier sense*, escutando o meio antes de tentar enviar os seus pacotes. Estes protocolos providenciam robustez e escalabilidade à rede.

O protocolo MAC IEEE 802.11, baseado na técnica CSMA/CA (*Carrier Sense Multiple Access with Collision Avoidance*) constitui a base para muitos dos protocolos de competição pelo meio, em desenvolvimento para as redes de sensores sem fios. No entanto, estes apresentam fracas prestações a nível de eficiência energética visto que os nós têm de estar à escuta do canal para competir por este antes de transmitir. Assim, melhorias são necessárias para cenários de aplicação neste tipo de redes.

### Baseados em reserva

Os protocolos baseados em reserva têm a vantagem de uma comunicação livre de colisões dado que cada nó transmite dados durante um período de tempo reservado. Desta forma, o período de funcionamento dos nós é diminuído resultando em maior eficiência energética. Protocolos baseados em TDMA (*Time Division Multiple Access*) têm sido propostos e seguem princípios comuns, em que cada nó comunica de acordo com uma estrutura específica designada de *superframe*. Esta consiste geralmente num período de reserva em que os nós reservam os *slots* de tempo para comunicação e um período de dados que consiste em múltiplos *slots* que são usado por cada sensor para transmitir informação.

### Híbridos

Estas técnicas protocolares têm diferentes vantagens e desvantagens para as capacidades gerais do acesso ao meio. Os protocolos que seguem um método híbrido pretendem combinar algumas das características destas técnicas como esquemas de acesso aleatório ao meio com abordagens de acesso baseado em reserva por TDMA. Os protocolos híbridos proporcionam melhorias no que diz respeito a colisões e eficiência energética através de uma melhor organização do canal e adaptabilidade à carga do mesmo.

#### 2.4.3. Standards de comunicação

Apresentam-se então os principais *standards* de comunicação, que conseqüentemente servem de base a outras normas das diferentes camadas do *stack* protocolar, usados pelas redes de sensores sem fios, que também aqui se apresentam [14]:

#### IEEE 802.11x

O IEEE 802.11 representa um conjunto de *standards* para implementação de WLANs. Este destinou-se originalmente para a transmissão de dados, com taxas relativamente elevadas, entre computadores e outros dispositivos. Estas taxas variam dos 1Mbps aos 150Mbps, com alcances até aos 250m, através, maioritariamente, das bandas de frequência de 2.4 GHz e 5 GHz [17]. Estas características mostram-se interessantes do ponto de vista da implementação de redes de sensores

## Proposta de Gateway para Redes de Sensores Sem Fios

sem fios, no entanto, os elevados requisitos energéticos levam a que estes *standards* sejam preteridos em relação a outras normas [14].

### Bluetooth, IEEE 802.15.1

O Bluetooth é um *standard* para as redes WPAN que opera na banda não licenciada de 2.400-2.4835GHz [18] e foi originalmente pensado para servir aplicações, como a transferência de dados, entre computadores pessoais e periféricos, como telemóveis ou PDAs (*Personal Digital Assistants*) [14]. Embora possa ser visto como uma possibilidade para servir de base às redes de sensores sem fios, possui ainda limitações para se tornar um alternativa viável: consumo de energia relativamente elevado (embora menor que as normas IEEE 802.11), capacidade reduzida de número de nós por rede, alcance curto (poucas dezenas de metros), sincronização pouco eficiente, camada MAC complexa.

### IEEE 802.15.4

Desenvolvida pelo IEEE *Task Group 4*, esta norma pode ser vista como a versão de baixo consumo de energia do IEEE 802.11. Define a camada física e MAC do *stack* protocolar e providencia elevada flexibilidade para soluções de camadas superiores. Consiste de três bandas: uma global, com 16 canais, nas frequências 2.400-2.4835GHz; uma americana, com 10 canais, nas frequências 902-928 MHz; uma europeia, de um canal, nas frequências 868-868.6MHz. Estas bandas permitem velocidades na ordem dos 250kbps, 40kbps e 20kbps, respetivamente, com alcances dos 10 aos 100m [19]. Especificamente desenhado para os requisitos de aplicações de sensores sem fios é expectável que se torne o *standard* mais usado, principalmente a banda de 2.4GHz, globalmente livre de licença.

### ZigBee

A aliança ZigBee é uma associação de empresas e entidades que trabalham em conjunto para desenvolver produtos, ligados via *wireless*, de monitorização e controlo que sejam fiáveis, de baixo preço e consumo energético, baseados numa norma global [14]. Apoiado na camada física e MAC do *standard* IEEE 802.15.4, a aliança ZigBee procura criar normas para as camadas superiores de rede e aplicação. A camada de rede procura providenciar funcionalidades de ligação para diferentes topologias de rede e a camada de aplicação uma base de trabalho para desenvolvimento de aplicações distribuídas e comunicação entre estas [9].

### IEEE 1451

Esta norma reúne uma família de *standards* conhecidos como *Smart Transducer Interface Standards*, que definem um conjunto de interfaces de comunicação livres, comuns e independentes da rede para ligação de sensores/atuadores a microprocessadores, sistemas de instrumentação e redes de controlo. O objetivo é facilitar o desenvolvimento de dispositivos inteligentes e da sua ligação a redes sistemas e instrumentos através da incorporação de tecnologias de sensores e redes existentes e emergentes [9].

### 2.5. Desafios e trabalho futuro

As redes de sensores em geral, apresentam algumas dificuldades técnicas relacionadas com processamento de dados, comunicação e gestão de sensores. Devido a ambientes severos e dinâmicos e a restrições de energia e largura de banda, as redes de sensores sem fios apresentam ainda desafios adicionais relacionados com mapeamento da rede e compreensão das suas características, *routing* e controlo da rede, processamento de sinal e informação com cooperação, gestão de tarefas e de consultas à rede e segurança [8]. Ultrapassar algumas destas dificuldades é o próximo passo no desenvolvimento deste tipo de redes, por forma a maximizar as suas capacidades e, conseqüentemente, alcançar a sua sedimentação nas aplicações que foram referidas, ao mesmo tempo que se alarga o leque de aplicações em que estas podem trazer grandes benefícios.

O principal desafio, como foi várias vezes referido, prende-se com questões relacionadas com o consumo energético. Estão a ser desenvolvidos trabalhos em sistemas que exploram materiais piezoelétricos, entre outros, para recolher energia do ambiente e guardá-la em condensadores ou baterias recarregáveis. Combinando eletrónica inteligente e eficiente a nível energético, com novas tecnologias de baterias, estes sistemas podem providenciar a melhor solução para aplicações de monitorização apresentando as vantagens de uma comunicação sem fios, de longa autonomia e livre de manutenção [14].

## **Proposta de Gateway para Redes de Sensores Sem Fios**

## 3. Sistema

---

Após a introdução do projeto a desenvolver e do levantamento de informações e aprendizagem de conceitos relativos a vários aspetos das redes de sensores sem fios pode-se, agora, com mais detalhe, definir o sistema a implementar, o meio em que este se insere e o modo como deve interagir com os restantes intervenientes deste. Apresentam-se também as especificações do sistema, assim como *hardware* e *software*, necessários para o seu desenvolvimento.

### 3.1. Definição do sistema

Pretende-se que o *gateway* seja integrado numa rede em estrela em que este é a estação base, com a qual os nós remotos comunicam e único responsável pela gestão da rede, como foi referido no capítulo anterior. A capacidade da rede, em número de nós, deve ser a maior possível, sendo que tal estará dependente das capacidades dos dispositivos e das soluções protocolares propostas.

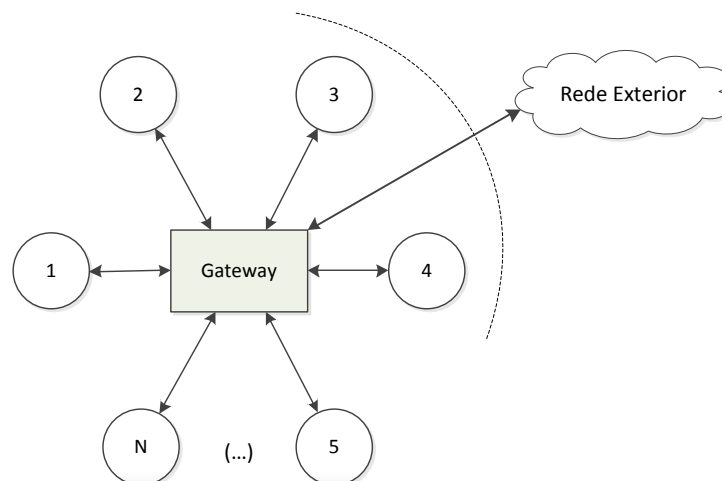


Figura 10 - Topologia da rede de sensores sem fios

Pelo uso desta topologia, como foi visto, obtém-se eficiência energética e reduzida latência. No caso da eficiência energética, no entanto, tal dificilmente se aplica para o *gateway*, visto que

## Proposta de Gateway para Redes de Sensores Sem Fios

este necessita de escutar todos os nós e de garantir a gestão da rede. Desta forma, e uma vez que se assumirá que este possui uma fonte de energia fixa e constante, não se exigem restrições a este nível. O *gateway* mantém assim fixa a sua posição ao longo do tempo, no entanto, tal não é requerido aos nós remotos que podem alterar a sua posição, desde que dentro do alcance dos *transceivers*. No caso dos nós, sem uma fonte de energia fixa, exigem-se restrições a nível de *firmware* e de protocolo de acesso ao meio para que a sua longevidade seja a maior possível.

Aquilo que se pretende do *gateway* encontra-se sumariamente representado na figura seguinte.

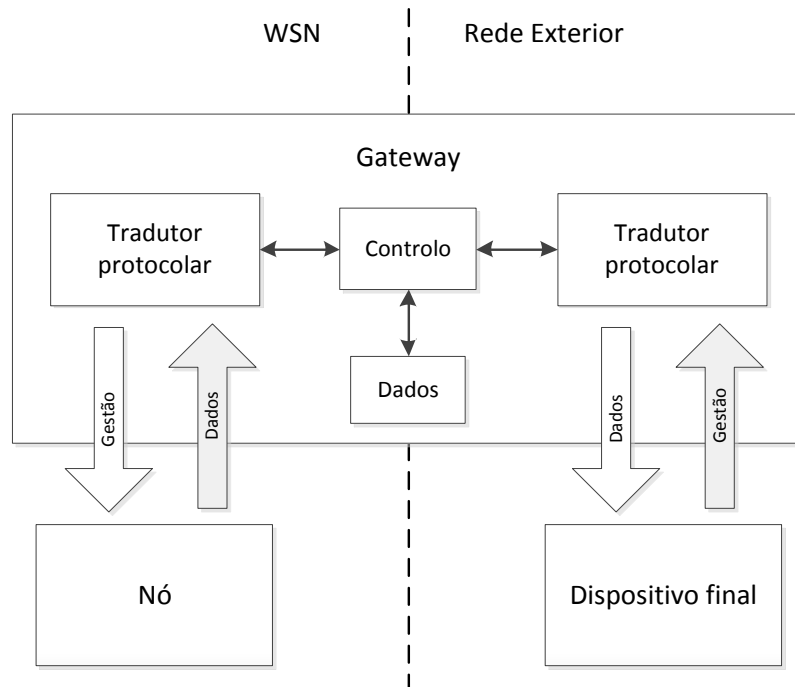


Figura 11 - Esquema geral do gateway

Como a imagem mostra, a plataforma irá fazer a ponte entre dois meios de características diferentes. Para tal, esta tem de ter integradas em si interfaces capazes de interagir com estes meios e nesta têm de ser implementados tradutores protocolares, tendo em conta os protocolos usados nas redes. No que diz respeito a comunicações, o *gateway* recebe dados dos nós da rede de sensores e partilha com estes mensagens relativas à gestão da rede. Do lado da rede exterior recebe mensagens ou pedidos de gestão da rede e envia os dados que forem exigidos.

Assim, resumidamente, aquilo que se exige ao dispositivo é:

- Permitir a escalabilidade e dinamismo da rede com a adição de novos nós ou desativação de outros. Dinamismo, também, do ponto de vista da localização física dos nós.
- Dependendo do modo de uso detetar e apresentar falhas da rede.
- Implementar um protocolo MAC que permita elevada eficiência energética dos nós remotos, maximizando a longevidade destes. Protocolo este também capaz de garantir as trocas de informação em tempo real evitando, ao mesmo tempo, os problemas analisados, como as colisões ou o *overhearing*.



### 3.2. Especificação de requisitos

É necessário agora ser um pouco mais concreto no que diz respeito às capacidades e características do dispositivo e do meio em que se deve integrar. Estas são condicionadas pelo material disponível e por aquilo que se crê serem as exigências atuais, tendo em conta a parceria existente com uma empresa presente e conhecedora do mercado.

Desta forma, do lado da rede de sensores sem fios admite-se a existência de uma rede baseada em *hardware* sob a alçada da norma ZigBee e do lado exterior uma ligação *Ethernet* (protocolo TCP/IP).

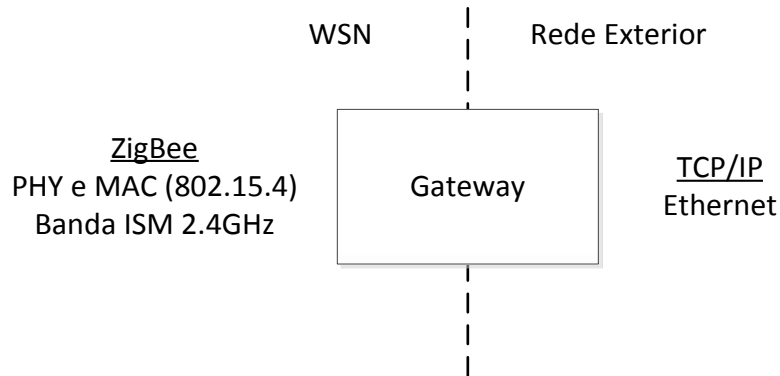


Figura 12 - Interações gateway

Do lado da WSN, a escolha prende-se essencialmente com:

- *Hardware (transceivers)* já disponível.
- Banda ISM livre de 2.4GHz.
- Programação e configurações simples (como vai ser analisado mais à frente).
- Liberdade de edição da camada protocolar de acesso ao meio, por forma a adaptar os *transceivers* à solução proposta.

No que diz respeito à rede exterior o motivo da escolha deve-se à simplicidade, visto que a placa a usar para desenvolvimento do *gateway* possui interface *Ethernet* (secção seguinte) e, assim sendo, para estabelecer a ligação física é necessário apenas um cabo.

No campo do controlo do processo e de interação entre nós e *gateway* admite-se que:

- A comunicação deve ser feita em tempo real, ou seja, se se pretende controlar um processo com um período de 10 segundos, por exemplo, o *gateway* deverá receber informações do nó a cada 10 segundos.
- Independentemente do protocolo escolhido um envio de dados para o *gateway* deverá ser sucedido de uma resposta deste para o nó emissor.

### 3.3. Hardware

Após os pontos anteriores, nos quais é introduzido aquilo que se pretende que seja o sistema a desenvolver, é importante definir qual o *hardware* que irá ser utilizado para esta implementação.

#### 3.3.1. FriendlyARM

Para a implementação do *gateway* a escolha, a nível de *hardware*, recaiu sobre uma placa de desenvolvimento da família FriendlyARM. As placas desta família são muito procuradas para o desenvolvimento de sistemas embutidos, devido às suas capacidades e características respeitantes a componentes e interfaces, como se verá no decorrer desta secção.

Em concreto, a placa de desenvolvimento usada é a Micro2440-SDK que tem como base o microprocessador S3C2440 ARM9 de 400MHz da Samsung, especialmente vocacionado para integração em sistemas embutidos.

ARM9 é uma família de microprocessadores da arquitetura ARM (*Advanced RISC Machines*, em que RISC significa *Reduced Instruction Set Computer*) de 32 bits. Os processadores desta arquitetura tem um uso extensivo na eletrónica de consumo, como telemóveis, câmaras digitais, equipamento de áudio, entre outros. Em 2009, do mercado de processadores RISC de 32 bits em sistemas embutidos, cerca de 90% destes diziam respeito aos processadores ARM [20].

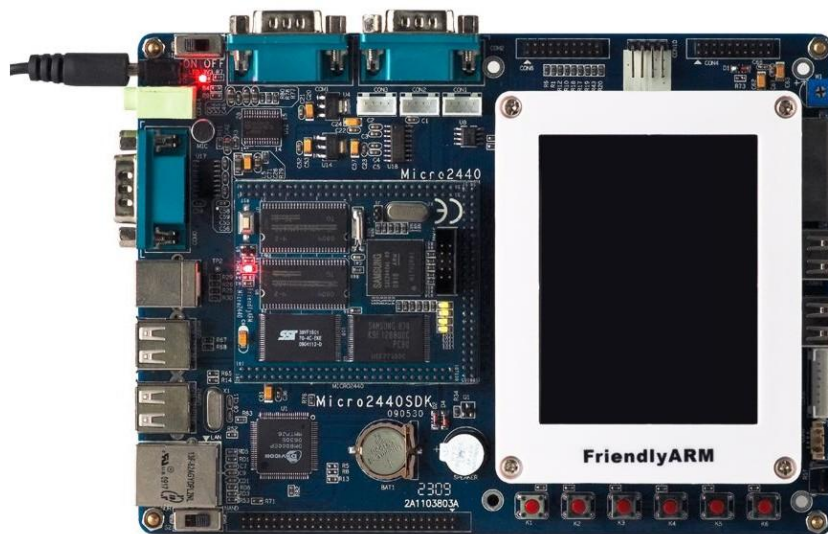


Figura 13 - Placa de desenvolvimento FriendlyARM

Assim, as principais características de *hardware* da placa em uso são [21]:

- CPU
  - Samsung S3C2440A, 400 MHz, máximo 533 MHz
- SDRAM
  - 64MB SDRAM
  - Data bus de 32 bits
  - Máxima frequência de clock SDRAM, 100 MHz
- Memória FLASH
  - 64MB Nand Flash

## Proposta de Gateway para Redes de Sensores Sem Fios

- 2MB Nor Flash com BIOS pré-instalada
- LCD

Outro dos pontos fortes desta placa de desenvolvimento diz respeito às interfaces e acessórios externos que esta possui. O número e variedade destes tornam-na muito apreciada para o desenvolvimento de sistemas embutidos. Nesta encontram-se, por exemplo [21]:

- Porta 100M Ethernet RJ-45 (chip de rede DM9000)
- 3 portas série
- Porta USB Host
- Porta USB Slave B
- Interface cartão SD
- Saída áudio Stereo e interface de microfone
- Porta GPIO de 34 pinos 2.0mm

No desenvolvimento do trabalho apenas uma parte das características da placa, tanto a nível de capacidades de processamento e armazenamento como de interfaces, são usadas. No entanto, quando se fala em redes de sensores sem fios, deve-se ter em conta a escalabilidade da rede e, assim, também as exigências que podem vir a ser feitas às capacidades do *hardware* ao longo do tempo. Desta forma se justifica o uso deste tipo de placas em detrimento de outras menos capazes.

### 3.3.2. XBee

Após a identificação da placa com a qual será implementado o núcleo do *gateway*, é necessário analisar como vai ser feita a ligação deste à rede de sensores sem fios. Como foi visto, a placa não possui capacidades de comunicação sem fios. Desta forma, terá de ser acoplado (através de uma das interfaces existentes) um módulo com as devidas capacidades para estabelecer uma comunicação com a rede.

A escolha para estabelecer a comunicação com a rede sem fios recaiu num módulo da família XBee, da Digi International (introduzidos no mercado inicialmente sob a marca MaxStream). A escolha de um módulo desta família deve-se ao facto de esta, possivelmente, representar a família com os módulos de rádio compatíveis mais usados neste tipo de aplicações.

Os módulos XBee foram desenhados para irem ao encontro do standard IEEE 802.15.4, suportando as necessidades de baixo custo e consumo energético das redes de sensores sem fios e proporcionando normas ZigBee para as camadas superiores. Esta família conta com duas principais variantes: XBee e XBee-PRO que operam na gama ISM de frequência 2.4GHz. O aspeto geral dos modelos é em tudo semelhante, com compatibilidade pino a pino um com o outro, podendo apresentar antena externa ou não.

## Proposta de Gateway para Redes de Sensores Sem Fios



Figura 14 - Módulos XBee

Entre as principais especificações destes módulos encontram-se [22]:

Especificação	XBee	XBee-PRO
Performance		
Alcance Indoor	Até 30m	Até 60m
Potência de transmissão	1mW	10mW
Ritmo de transmissão RF	250,000bps	250,000bps
Ritmo de dados interface série	1200bps - 250kpbs	1200bps - 250kpbs
Requerimentos energéticos		
Tensão de alimentação	2.8 - 3.4V	2.8 - 3.4V
Corrente de transmissão (típica)	45mA (@3.3V)	150mA (@3.3V)
Corrente idle/recepção (típica)	50mA(@3.3V)	55mA (@3.3V)
Corrente power-down	<10 $\mu$ A	<10 $\mu$ A

Tabela 1

No presente projeto, e para o resto da análise, assume-se o uso de módulos do modelo XBee. No entanto, as análises feitas são semelhantes para o modelo XBee-PRO, com as principais diferenças sendo referentes aos consumos energéticos.

Associados a estes módulos existem os *kits* de desenvolvimento XBee que constam de placas que, do ponto de vista do utilizador, têm como principais valias as interfaces RS-232 e USB. Estas permitem uma configuração simples dos módulos em conjunto com a ferramenta X-CTU, que é abordada no próximo capítulo.

## Proposta de Gateway para Redes de Sensores Sem Fios



Figura 15 - Placas de desenvolvimento para os módulos XBee

Importa, para uso posterior, fazer referência à arquitetura interna destes módulos em termos do fluxo de dados.

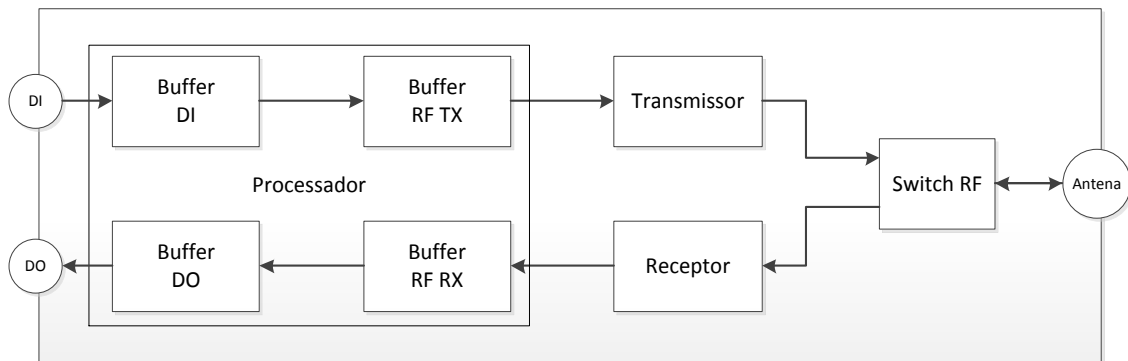


Figura 16 - Diagrama do fluxo interno de dados do módulo XBee

Como se pode ver pela Figura 16 [22], os dados são introduzidos pelo pino DI (*Data Input*) e conduzidos ao *buffer* DI. A partir desse ponto, caso o processador confirme a transmissão, estes são enviados para o *buffer* RF de transmissão, conduzidos para o transmissor e enviados pela antena. No caso de receção de dados o caminho é o inverso até ao pino DO (*Data Output*).

Também para análises futuras apresentam-se os modos de operação do módulo, que podem ser observados na figura seguinte [22].

## Proposta de Gateway para Redes de Sensores Sem Fios

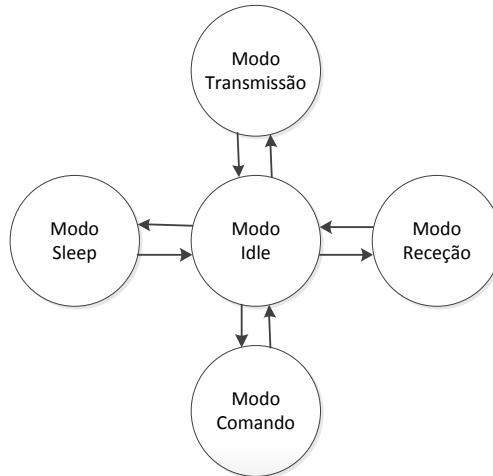


Figura 17 - Modos de operação do módulo XBee

Quando não está a receber ou a transmitir dados o módulo encontra-se no *Modo Idle* e altera o seu modo caso:

- Receba dados no *buffer* DI (*Modo Transmissão*)
- Receba dados RF válidos através da antena (*Modo Recepção*)
- Condições para 'adormecer' estejam cumpridas (*Modo Sleep*)
- Haja emissão de sequências de comando (*Modo Comando*)

### 3.4. Software

#### 3.4.1. Escolha do sistema operativo do gateway

Após a referência à placa de desenvolvimento a usar na implementação do *gateway*, a escolha seguinte diz respeito ao sistema operativo a carregar na mesma. Para tal, primeiro devem ser exploradas quais são as opções possíveis para desempenhar tal papel e tentar diferenciá-las, expondo vantagens e fragilidades, por forma a descortinar aquela que melhor se adequa ao projeto em mãos.

É também válida a possibilidade de manter o dispositivo livre de um sistema operativo. No entanto, tal implica um grande compromisso de desenvolvimento de código de mais baixo nível para gerir e executar as diferentes funções necessárias ao funcionamento do dispositivo, e de acordo com as necessidades do utilizador. As vantagens de usar um sistema operativo passam por:

- Permitir um agendamento determinístico em tempo real e priorização de tarefas.
- Abstrair o utilizador da complexidade do processador.
- Providenciar uma infraestrutura sólida baseada em regras e políticas.
- Integrar, gerir e otimizar recursos.

Assim sendo, a melhor opção passa mesmo por incluir um sistema operativo na placa de desenvolvimento. Existe uma grande variedade de sistemas operativos especificamente

## Proposta de Gateway para Redes de Sensores Sem Fios

vocacionados para a utilização em sistemas embutidos como: TinyOS, Embedded Linux, Windows CE, Symbian OS, iOS.

Da vasta lista de possibilidades, a placa de desenvolvimento FriendlyARM suporta sistemas operativos baseados em Linux e Windows, mais propriamente os sistemas Embedded Linux e o Microsoft Windows CE (cujo nome oficial é Windows Embedded Compact).

Como no caso dos sistemas operativos para sistemas *desktop* e outros, existe uma grande discussão sobre as vantagens e desvantagens da utilização de cada uma destas soluções. Torna-se, portanto, importante enumerar alguns dos argumentos usados para defender uma e outra solução, por forma a tornar o processo de escolha mais simples.

Em defesa do Embedded Linux os principais argumentos são:

- *Software open-source*, permitindo ao utilizador estudar ou alterar o código fonte do mesmo, com vista à adaptação aos objetivos e eventualmente partilhar estas alterações.
- Não possui nenhum custo de licença para uso ou redistribuição.

No caso do Windows CE:

- *Software* proprietário geralmente permite facilidade em obter suporte do fabricante como *updates* ou *drivers*.
- Mais estável no que diz respeito à resposta em tempo real [23].

Deve-se ter em conta que estes argumentos são, maioritariamente, baseados em testemunhos e opiniões de pessoas individuais e não têm por base estudos determinísticos às capacidades dos sistemas ou à apreciação geral dos mesmos. No entanto, existe a opinião que se um utilizador tiver pouca ou nenhuma experiência no desenvolvimento de sistemas embutidos a melhor escolha será o Windows CE, por ter à disposição ferramentas de construção mais simples que Linux, permitindo assim manter também o projeto o mais simples possível e com uma abordagem menos vasta que aquelas disponíveis via *software open-source*.

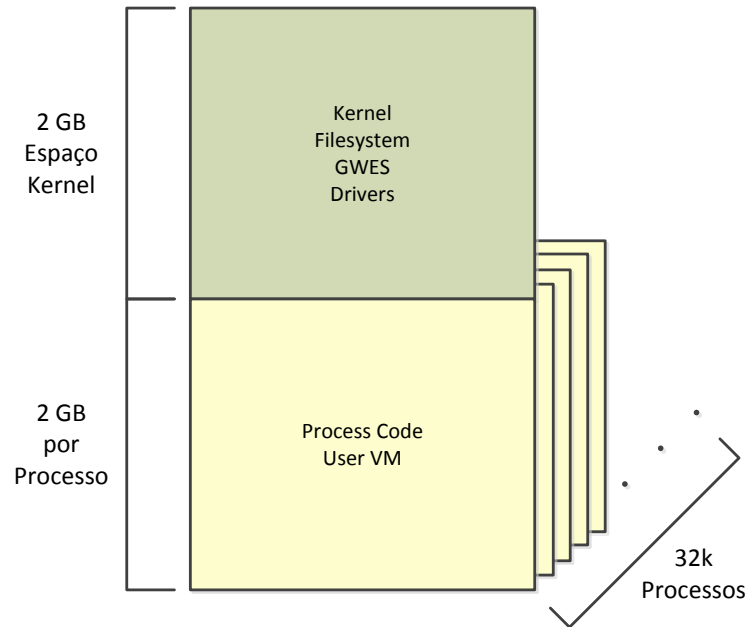
Assim sendo a escolha recai sobre o Windows CE como sistema operativo a carregar na placa de desenvolvimento.

### 3.4.2. Windows CE

O Microsoft WindowsCE é um sistema operativo desenvolvido para pequenos sistemas computacionais e sistemas embutidos, desenhados para desempenhar funções dedicadas com restrições, ou não, de tempo real e em *multithreading*. O Windows CE tem um conjunto de capacidades, com características de sistema operativo completo, e ferramentas de desenvolvimento necessárias para criar, testar e implementar dispositivos e sistemas embutidos com base nele.

O sistema operativo de 32 *bits* corre em múltiplas arquiteturas de processadores como MIPS, x86 e aquela que interessa para este projeto, ARM. Este opera num espaço de endereçamento virtual de 4GB, usando o *kernel* do sistema os 2GB superiores da memória virtual e os processos ativos do utilizador os 2GB de baixo, da forma apresentada pela figura [24].

## Proposta de Gateway para Redes de Sensores Sem Fios



**Figura 18 - Espaço de memória virtual**

O sistema suporta até 32000 processos de utilizador, dependendo das limitações dos recursos do sistema, o que mostra as grandes capacidades deste. Estes processos incluem processos especiais que tornam a API (*Application Programming Interface*) disponível para as aplicações do utilizador.

A Figura 19 [24] apresenta a arquitetura do sistema operativo Windows CE. No espaço do *kernel* está o núcleo do sistema operativo que é o processo Nk.exe, no qual bibliotecas dinâmicas responsáveis pelos vários tipos de funcionalidades do sistema são carregadas. É este o ficheiro que é necessário gerar e colocar em memória para a utilização do sistema operativo. Bibliotecas do sistema e *drivers* podem também ser carregados neste espaço do *kernel*, que tratam das interações com o *hardware*.

No espaço do utilizador estão as aplicações e a API do sistema, disponível para aplicação via biblioteca *coredll.dll*, que está ligada a todos os módulos executáveis do sistema operativo. O utilizador tem também acesso a funcionalidades aplicadas através de diferentes bibliotecas de aplicações. Em adição à API do sistema, o sistema operativo oferece uma API de aplicação que é similar à API Win32 dos sistemas *desktop*.



## Proposta de Gateway para Redes de Sensores Sem Fios

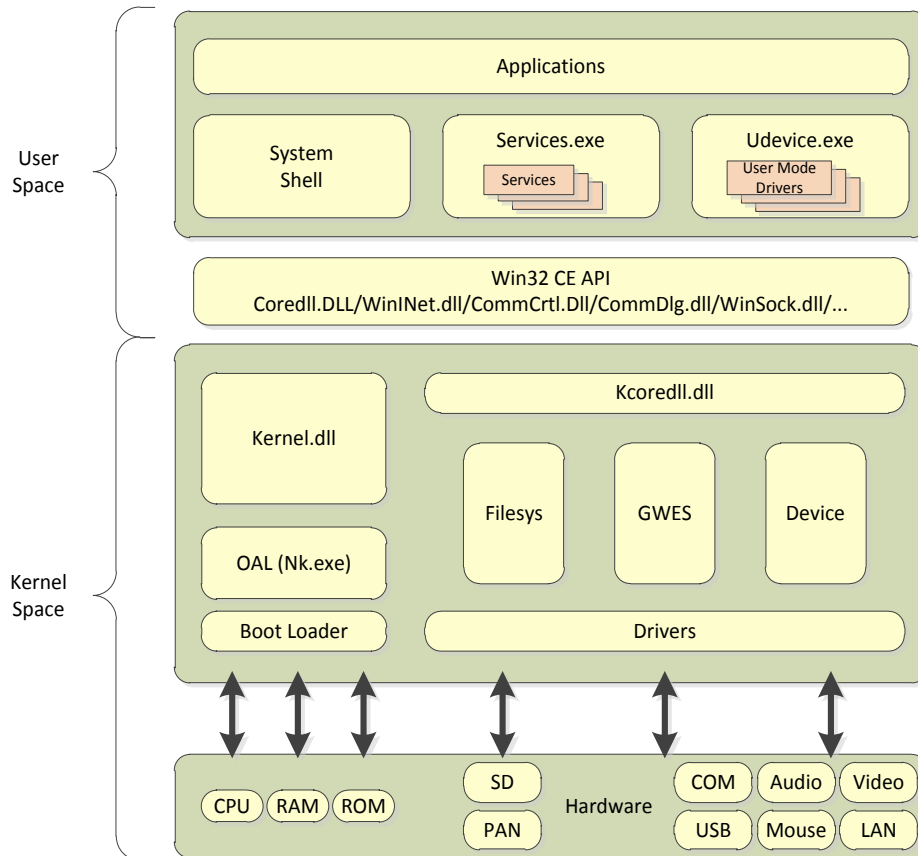


Figura 19 - Arquitetura Windows CE

Na versão 6.0 do Windows CE a ferramenta *Platform Builder* instala-se por cima do Visual Studio 2005 Pro que, como vai ser abordado, é o ambiente de desenvolvimento usado para a criação de aplicações do sistema.

Um BSP (*Board Support Package*) é necessário para adaptar o Windows CE a um dispositivo em particular. Os elementos contidos neste BSP e a forma como estes se ligam e relacionam entre si, o BSP e a SDB (*Standard Development Board*), ou plataforma de *hardware*, estão representados na figura seguinte.

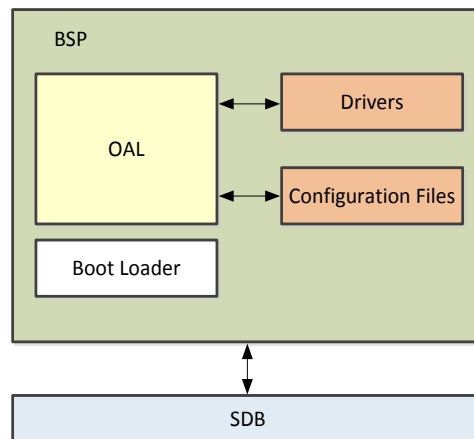


Figura 20 - Elementos BSP

## Proposta de Gateway para Redes de Sensores Sem Fios

Os elementos presentes no BSP são:

- Bootloader - Carrega a imagem do sistema operativo.
- OAL (*OEM Adaption Layer*) - Camada de adaptação que faz a ligação à imagem do *kernel* e suporta a inicialização e gestão do *hardware*.
- *Device Drivers* - Suportam os periféricos do SDB ou que se encontrem ligados a este.
- *Run-time image configuration files* - Ficheiros de configuração que permitem reconfigurações do BSP.

Estes dados são importantes para compreender as capacidades e limitações da integração destas ferramentas, em conjunto com o *hardware*, e também a forma como essa integração deve ser desenvolvida.

### 3.5. Conclusão

Neste capítulo deram-se a conhecer as características gerais do sistema em desenvolvimento assim como os requisitos de implementação. Deu-se a conhecer o *hardware* disponível e concluiu-se que este é capaz de se enquadrar com os requisitos exigidos. Também o sistema operativo analisado apresenta as capacidades necessárias para a implementação do dispositivo final. Arquiteturas de *hardware* e *software* foram apresentadas e analisadas, para compreensão das considerações a fazer ao longo do processo de integração de componentes, e no desenvolvimento de aplicações para o dispositivo final.

## 4. Implementação

A implementação de uma rede e a introdução de um dispositivo deste tipo nessa rede levam a um grande número de considerações. O objetivo deste capítulo é, portanto, analisar os principais problemas e dificuldades tendo em conta alguns pressupostos.

### 4.1. Arquitetura

Após a análise feita no capítulo anterior pode-se agora olhar para o conjunto final daquilo que se prevê que seja a constituição e organização do *gateway*, no que diz respeito à arquitetura física do mesmo.

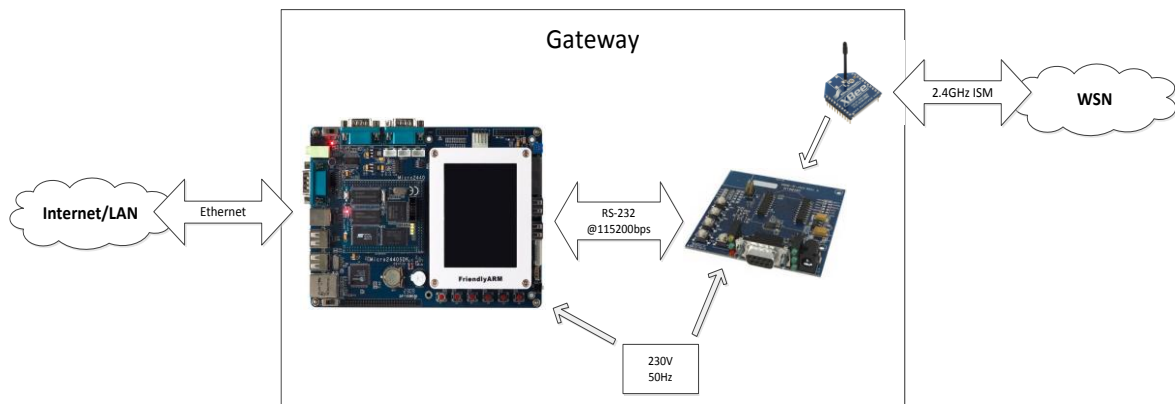


Figura 21 - Arquitetura física *gateway*

Como foi visto, na figura apresentam-se então os componentes de *hardware* necessários à constituição do dispositivo. São estes: a placa de desenvolvimento FriendlyARM e o módulo de comunicação XBee, apoiado pela sua placa de desenvolvimento. Como se referiu estas placas possuem interfaces RS-2323 e são estas as interfaces usadas para se estabelecer a comunicação entre estes dois componentes a um ritmo de transmissão de 115200bps. Ambas as placas de desenvolvimento são alimentadas a 230V disponíveis da rede elétrica.

Para alcançar este desenho é necessário estabelecer:

- Comunicação entre FriendlyARM e a placa de desenvolvimento do módulo XBee

## Proposta de Gateway para Redes de Sensores Sem Fios

- Comunicação entre módulos XBee
- Comunicação entre FriendlyARM e Internet ou uma rede de área local

### 4.2. Integração

#### 4.2.1. Comunicação FriendlyARM - XBee

Do lado do módulo XBee toda a informação propagada pelo cabo RS-232, chegando à porta série deste, é enviado para o *buffer* DI. Da mesma forma a informação que é enviada para o *buffer* DO propaga-se para a porta série e pelo cabo em direção ao FriendlyARM.

Por sua vez, do lado do FriendlyARM, foi discutido que a escolha no uso do Windows CE 6.0 permite uma maior simplicidade e abstração para o utilizador. No entanto, tal também se traduz em certas restrições nas capacidades de desenvolvimento, como é o caso de não permitir o acesso direto a registos sem o uso de *drivers* dedicados.

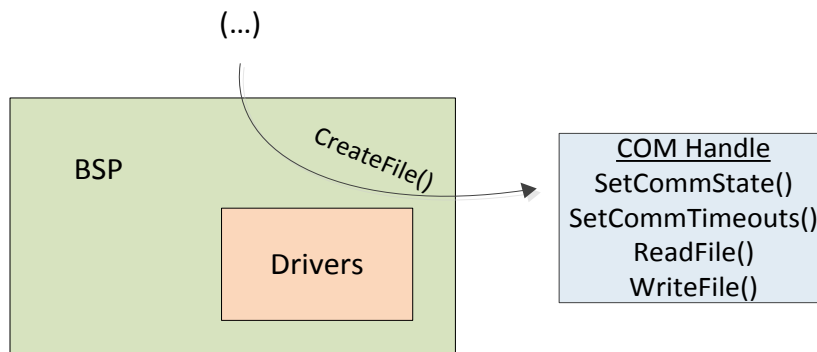


Figura 22 - Uso de *driver* para comunicação série

Assim, é necessário dar uso ao *driver*, existente no BSP da placa, para as portas série (COM), em conjunto com as funções existentes do Windows CE. O *driver* é aberto e um *handle* (referência abstrata ao recurso pretendido) é preparado pelo sistema operativo, o qual é passado para funções do sistema com as quais se podem, tendo em conta as exigências do projeto, definir parâmetros da comunicação série e ler ou escrever para a mesma.

Para teste destas capacidades um programa foi desenvolvido, cujo objetivo é precisamente testar parâmetros da ligação e permitir enviar e receber dados via RS-232. Através deste programa estabeleceu-se a ligação pretendida com um *baudrate* de 115200bps. O protótipo desta aplicação encontra-se no Anexo A.

#### 4.2.2. Comunicação entre módulos XBee

Os módulos XBee possuem dois modos de operação no que concerne a comunicação, são estes:

- Transparente - Este é o modo de operação dos módulos XBee por definição. Neste modo os módulos atuam como uma linha série em que todos os dados que chegam ao pino DI são enviados para o transmissor RF. Igualmente, toda a informação recebida pelo recetor RF é passada ao pino DO.

## Proposta de Gateway para Redes de Sensores Sem Fios

- API (*Application Programming Interface*) - Neste modo de operação todos os dados que entrem ou saiam do módulo são organizados em tramas pré-definidas.

O modo de operação escolhido para as comunicações sem fios pelos módulos XBee constituintes da rede (nós remotos e *gateway*) foi o API. Esta escolha deve-se às vantagens que a organização da informação em tramas introduz, assim como algumas funcionalidades que este modo apresenta. Entre estas, as que mais interessam para este sistema são:

- Simplificação dos preparativos para transmissão. Este modo segue um formato pré-definido de trama (particularmente importante para os nós remotos).
- Identificação do módulo emissor do pacote na trama.
- Introdução de CCA (*Clear Channel Assessment*).

As tramas do modo API dos módulos XBee regem-se por uma estrutura fixa que se apresenta na figura seguinte [22].

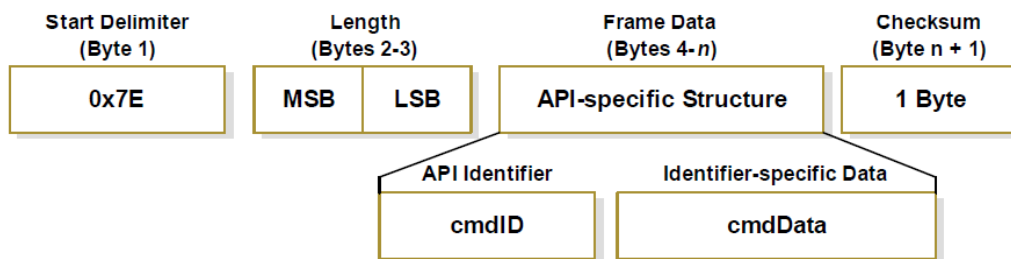


Figura 23 - Trama API

Esta estrutura encontra-se dividida em:

- *Start Delimiter* - Um *byte* que tem como objetivo sinalizar o início de uma nova trama.
- *Length* - Dois *bytes* relativos ao tamanho, em *bytes*, do campo *Frame Data*.
- *Frame Data* - Campo de tamanho variável onde está sinalizado o tipo de mensagem (*API Identifier*) e onde se encontram outros dados que variam de acordo com este tipo.
- *Checksum* - Um *byte* para teste da integridade dos dados.

Para os objetivos do projeto apenas dois tipos de mensagens API serão considerados. Um relativo ao envio de uma trama e outro à receção. São estes:

**TX Request: 16-bit address** Como o nome indica uma mensagem deste tipo traduz-se num pedido envio de dados e a geração da mensagem processa-se de acordo com o seguinte formato [22].

## Proposta de Gateway para Redes de Sensores Sem Fios

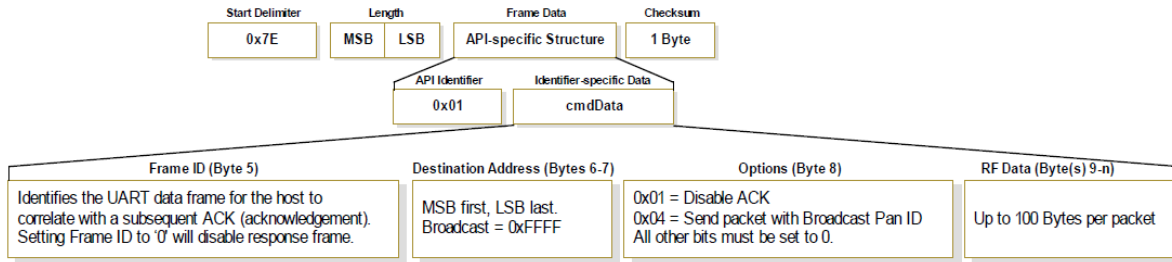


Figura 24 - Trama API TX Request

Constata-se que este tipo de mensagem é caracterizado pelo identificador API 0x01 e na mesma deve estar definido qual o endereço de destino do pacote. No campo de opções destaca-se a possibilidade de ligar/desligar o mecanismo de *acknowledge* (envio automático por parte do recetor da mensagem de uma mensagem de confirmação de receção). Neste projeto este mecanismo não estará em funcionamento. Por fim, o campo de dados permite um *payload* de até 100 *bytes* por pacote.

**RX Packet: 16-bit address** No caso da receção de uma mensagem esta será enviada para o exterior do módulo (pino DO) de acordo com a estrutura seguinte [22].

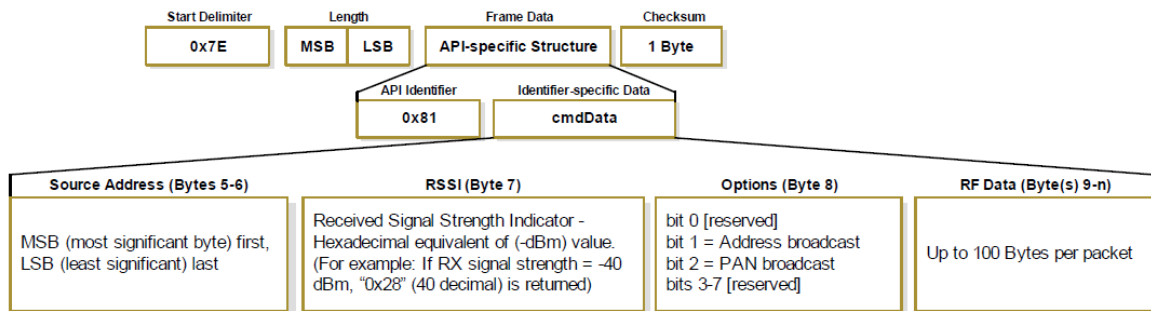


Figura 25 - Trama API RX Packet

Nesta mensagem, com o identificador 0x81, destaca-se o endereço do emissor da mensagem e ainda o RSSI (*Received Signal Strength Indicator*) do sinal recebido, que pode ser relevante para decisões relativas ao *layout* e construção da rede.

Existem outros dois tipos de mensagem cujo objetivo é exatamente o que acabou de ser visto para estes tipos. A diferença reside no facto de trabalharem com endereços físicos de 64 *bits* em lugar de 16 *bits*. Para o caso presente, considerando uma rede relativamente pequena que funcionará num ambiente mais ou menos restrito, um endereçamento a 16 *bits* é suficiente e permite reduzir a informação a transmitir, resultando numa poupança de energia. Considera-se também que todos os 100 *bytes* permitidos de *payload* são usados para transmissão de dados, resultando assim em tramas com um total de 109 *bytes*.

Ambos os tipos de comunicação foram testados através do uso da ferramenta X-CTU, da qual se fala numa secção mais à frente.

### 4.2.3. Comunicação FriendlyARM - Internet

Como foi visto, para estabelecer esta comunicação, é apenas necessário um cabo Ethernet. Após efetuada a ligação física, o Windows CE é responsável por garantir conectividade com a rede,

## Proposta de Gateway para Redes de Sensores Sem Fios

conetividade esta que é confirmada pela execução de comandos *ping*, quer de um computador da rede para a placa FriendlyARM, quer da placa para o computador. No entanto, se o objetivo é tornar dados acessíveis a computadores exteriores é necessário estabelecer algum tipo de acesso ao dispositivo através da Internet. Um dispositivo baseado em Windows CE pode funcionar como um Web Server (mediante a introdução de certos pacotes) permitindo a sua monitorização, configuração e controlo remoto.

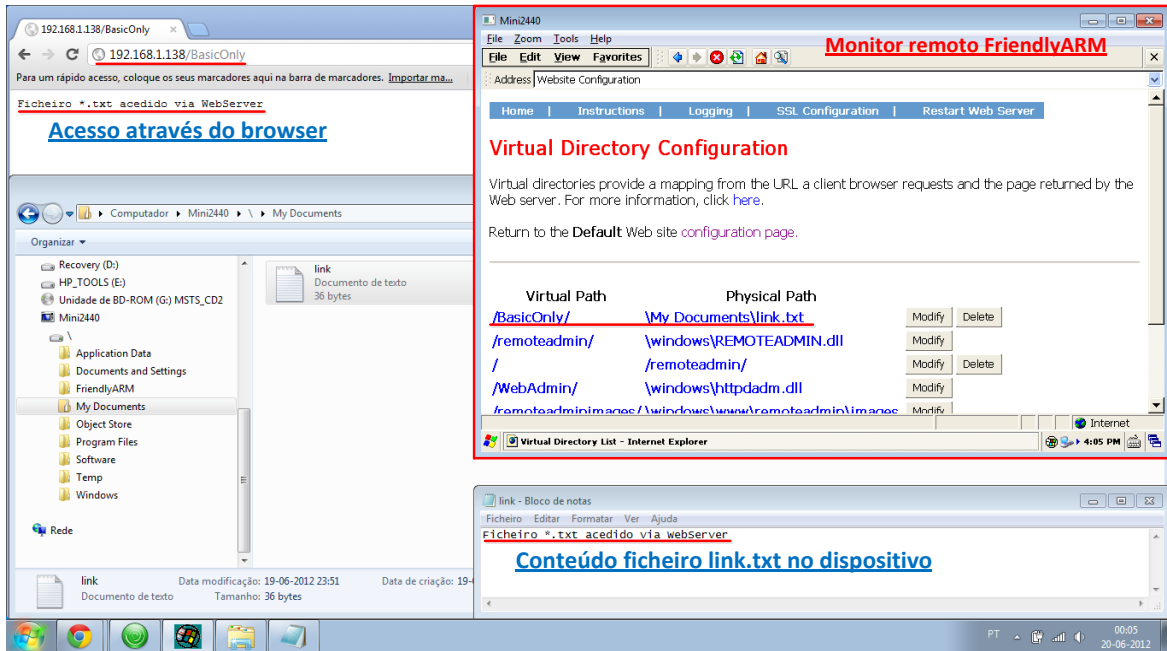


Figura 26 - Exemplo de leitura de ficheiro através da Internet

O Web Server permite, por exemplo, criar um caminho virtual que corresponde a um caminho físico do dispositivo. Como se vê na figura, o dispositivo apresenta o ficheiro 'link.txt' na pasta \\My Documentos\\ e tem a associado o caminho virtual /BasicOnly/. Este ficheiro pode ser acedido via *browser*, através do IP do dispositivo e do caminho virtual visto. Desta forma tornam-se acessíveis os dados pretendidos a um utilizador num terminal remoto, desde que com acesso à Internet.

### 4.2.4. Ambiente de desenvolvimento e outras ferramentas

O controlo do funcionamento e da utilização dos recursos do FriendlyARM, com vista à integração dos componentes, implica o desenvolvimento e carga de aplicações neste. Para tal recorre-se a um ambiente de desenvolvimento.

O ambiente de desenvolvimento é o Microsoft Visual Studio 2005 Pro, sobre o qual se instala o Windows CE 6.0, como já foi referido. Com recurso a este e ao SDK existente no BSP da placa podemos criar, compilar e carregar aplicações compatíveis com o sistema operativo e vocacionados para a placa em questão.

## Proposta de Gateway para Redes de Sensores Sem Fios

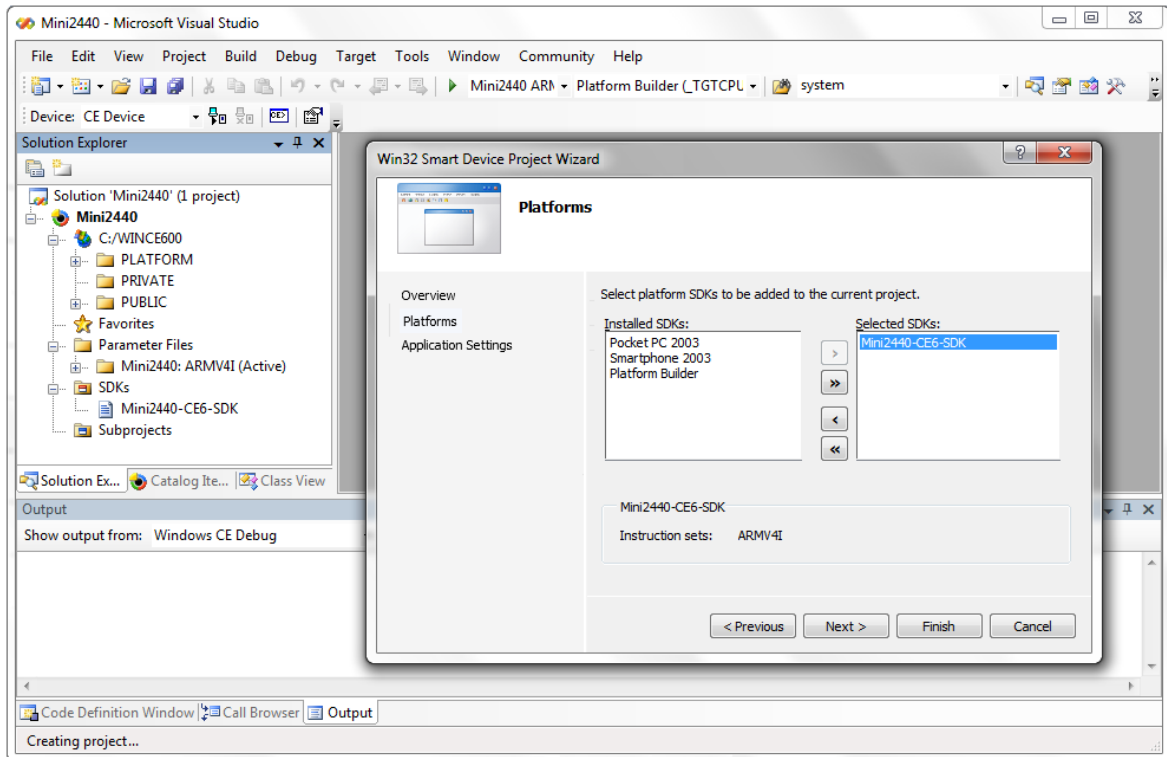


Figura 27 - Ambiente Microsoft Visual Studio 2005

A figura mostra o ambiente de desenvolvimento Visual Studio após a geração do SDK, pronto para iniciar projetos adaptados à placa Mini2440 (compatível com Micro2440) e ao sistema operativo Windows CE 6.0.

Para além do ambiente de desenvolvimento para o FriendlyARM outras ferramentas são necessárias para realizar e testar a integração dos componentes, como um terminal virtual ou *software* de apoio à placa, por exemplo. Entre estes destaca-se uma ferramenta de apoio aos módulos XBee fornecida pela Digi International. A ferramenta X-CTU, de ajuda à programação destes módulos, interage com o *firmware* presente nestes e, através de uma interface gráfica de fácil utilização, permite ao utilizador fazer *update* e *upgrade* de *firmware*, alterar parâmetros de configuração e testar a comunicação de uma forma simples.



## Proposta de Gateway para Redes de Sensores Sem Fios

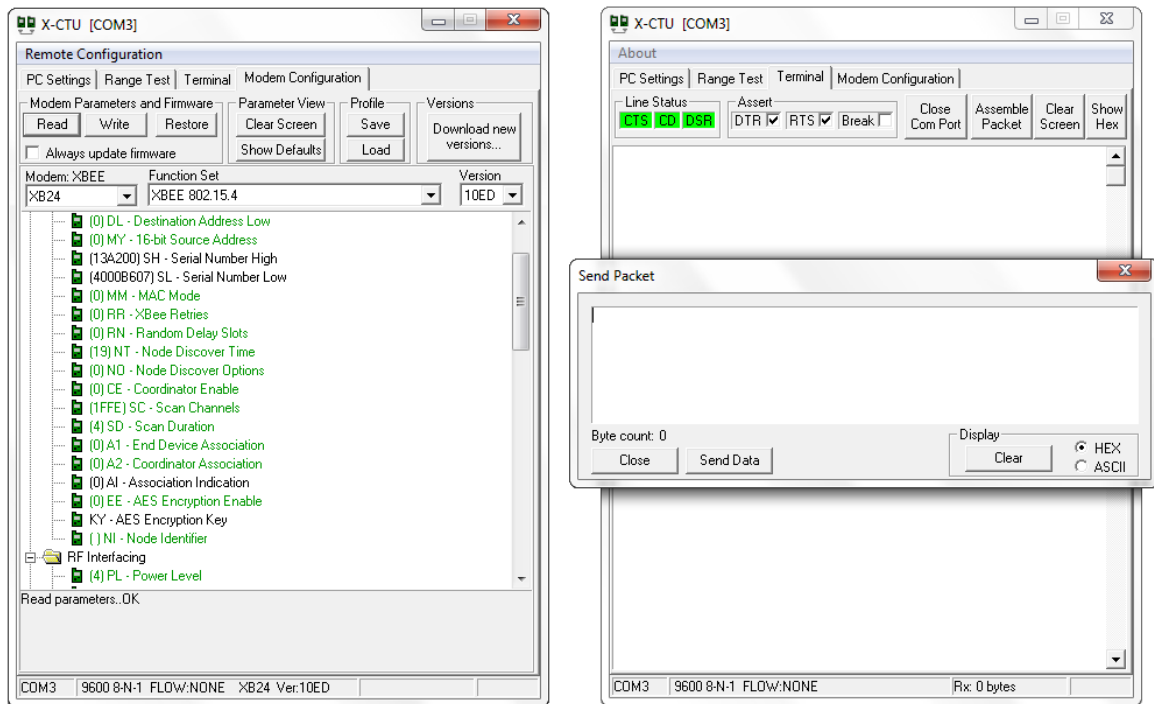


Figura 28 - Interface X-CTU

A figura apresenta a interface da aplicação relativa à alteração de parâmetros do módulo, através da qual facilmente se configura o dispositivo de acordo com o desejo do utilizador. Está presente também o separador onde se emula o terminal, que permite visualizar dados enviados e recebidos. Isto possibilita o teste das comunicações estabelecidas por configuração, inclusive no modo API, dado que permite construir pacotes.

### 4.3. Integração em rede

Após estas secções relativas à integração do *hardware*, para em conjunto constituir o *gateway*, é importante fazer algumas considerações relativas à sua introdução na rede de sensores sem fios.

#### 4.3.1. Etapas da troca de mensagens

Um dos pontos a ter em conta diz respeito às etapas de execução de uma troca de mensagens e, conseqüentemente, aos períodos temporais destes passos, mais concretamente os períodos de funcionamento dos módulos XBee, que deverão ser os principais responsáveis pelo consumo de corrente nos nós remotos. Isto é importante para estudar a ocupação do canal e os consumos energéticos relativos aos nós, por forma a avaliar a viabilidade do protocolo que será sugerido. A análise que se segue serve portanto esses dois propósitos.

Para fazer esta análise, parte-se do pressuposto que um ciclo normal de comunicação consistirá do envio de uma trama, por parte de um nó remoto para o *gateway*, seguida de uma resposta deste destinada ao nó emissor. Este ciclo deverá contar com várias etapas que se encontram discriminadas na figura seguinte e as quais se analisam.

## Proposta de Gateway para Redes de Sensores Sem Fios



Figura 29 - Etapas referentes a uma troca de mensagens

**Wake-up** Para iniciar uma nova comunicação, o módulo XBee tem de ser acordado (passagem do modo *Sleep* para o modo *Idle*). Este processo leva algum tempo a ser concluído até todas as funcionalidades do mesmo estarem disponíveis a serem usadas.

**Protocolo (1)** Após ser acordado pode-se tratar do envio da informação em si. Para concretizar este envio os seguintes passos são efetuados pelo protocolo do módulo XBee:

- Carregamento de dados no *buffer* DI do módulo.
- Verificação da trama que se pretende enviar - No modo API existe uma verificação da trama; se não for dada como válida, a trama não é passada ao *buffer* de transmissão e consequentemente enviada.
- CCA (*Clear Channel Assessment*) - Caso esta valência se encontre ativa o módulo faz uma leitura do canal de transmissão e verifica se este está ocupado. Caso tal se verifique, este recua na sua transmissão e tenta novamente passado um período de tempo aleatório. No caso de não estar ativa a transmissão é iniciada independentemente do estado do canal, o que pode resultar em colisões de tramas e perda de informação.

**Transmissão** Esta etapa diz respeito à transmissão dos dados em si. Após a etapa anterior os dados são passados ao transmissor e enviados pela antena.

**Protocolo (2)** Após o envio dos dados inicia-se uma nova fase protocolar mas desta feita da responsabilidade do *gateway*. Assim, temos os seguintes passos:

- Verificação da trama
- Descarregamento dos dados do *buffer* DO
- Processamento e geração da resposta
- Carregamento de dados no *buffer* DI
- Verificação da trama
- CCA

**Receção** O nó remoto recebe agora informações do *gateway* e passa-as para o *buffer* de receção.

**Protocolo (3)** Esta informação recebida passa pela mesma verificação de trama já vista e depois disso é transmitida para o *buffer Data Output*.

**Sleep** Para além das etapas vistas existe ainda o estado em que o nó remoto passa mais tempo. Neste estado toda a eletrónica do nó está desligada com o intuito de poupar bateria. Apenas o relógio do microcontrolador responsável pelo nó deverá estar em funcionamento, por forma a acordá-lo num instante previamente estabelecido para uma nova comunicação.

Após esta descrição é importante quantificar os períodos temporais associados a cada um destes passos e consequentes etapas da comunicação.

## Proposta de Gateway para Redes de Sensores Sem Fios

**Wake-up** O módulo XBee apresenta diferentes modos *sleep* e, com estes, diferentes períodos de *wake-up*. No caso presente, o objetivo é que o módulo seja acordado pelo microcontrolador associado a este no nó remoto. Ainda assim existem dois modos: um apresentando um tempo de *wake-up* de 13,2ms e um de 2ms. Apesar de parecer interessante escolher o modo que apresenta o menor tempo de *wake-up*, a verdade é que este tem um consumo mais elevado de corrente (50μA contra 10μA), o que leva à escolha do modo cujo tempo de *wake-up* é de 13,2ms.

**Protocolo (1)** O carregamento de dados no *buffer* depende do *baudrate* usado na comunicação do microcontrolador para o módulo. Considerando um *baudrate* de 115200bps e as tramas já vistas de 109 bytes temos:

$$\frac{109 \times 8}{115200} \approx 7,57ms$$

- Não existem dados para o tempo de verificação da trama, mas encontraram-se referências a processadores internos de módulos semelhantes com frequências de funcionamento da ordem dos MHz. Desta forma, este valor poder-se-á considerar desprezável tendo em conta os restantes.
- No que diz respeito à verificação do meio falamos de um tempo de 0,128ms [25].

**Transmissão** Na etapa de transmissão tem-se em consideração:

- Tempo para transmitir, tendo em conta, novamente, tramas de 109 bytes e um ritmo de transmissão de 250000bps:

$$\frac{109 \times 8}{250000} \approx 3,49ms$$

- Tempo de propagação da informação desde o nó remoto até ao *gateway*. Tendo em conta que as distâncias não excedem as poucas dezenas de metros este valor pode ser desprezado.

**Protocolo (2)** Como foi dito esta etapa protocolar diz respeito ao *gateway* e considera:

- Descarregar os dados do *buffer* depende do *baudrate* da comunicação entre módulo e o FriendlyARM. Aqui também são considerados 115200bps o que se traduz num tempo de 7,57ms.
- Processamento e geração de resposta pode também ser desprezado, tendo em conta os 400 MHz de frequência de funcionamento do processador ARM.
- Carregamento dos dados no *buffer* segue o princípio já visto, assim como a verificação da trama e CCA.

**Receção** Em tudo igual ao passo de transmissão

**Protocolo (3)** Verificação de trama e passagem para o *buffer* rege-se pelos valores também já apresentados.

## Proposta de Gateway para Redes de Sensores Sem Fios

**Sleep** Como foi dito este é o estado no qual o nó se encontrará a maior parte do tempo. Este tempo corresponde ao período de amostragem associado ao respetivo nó, subtraído dos tempos em que se encontra nas etapas analisadas.

Desta forma podemos chegar ao diagrama seguinte, que representa uma estimativa dos períodos temporais na sequência de uma comunicação e onde figuram os respetivos intervenientes.

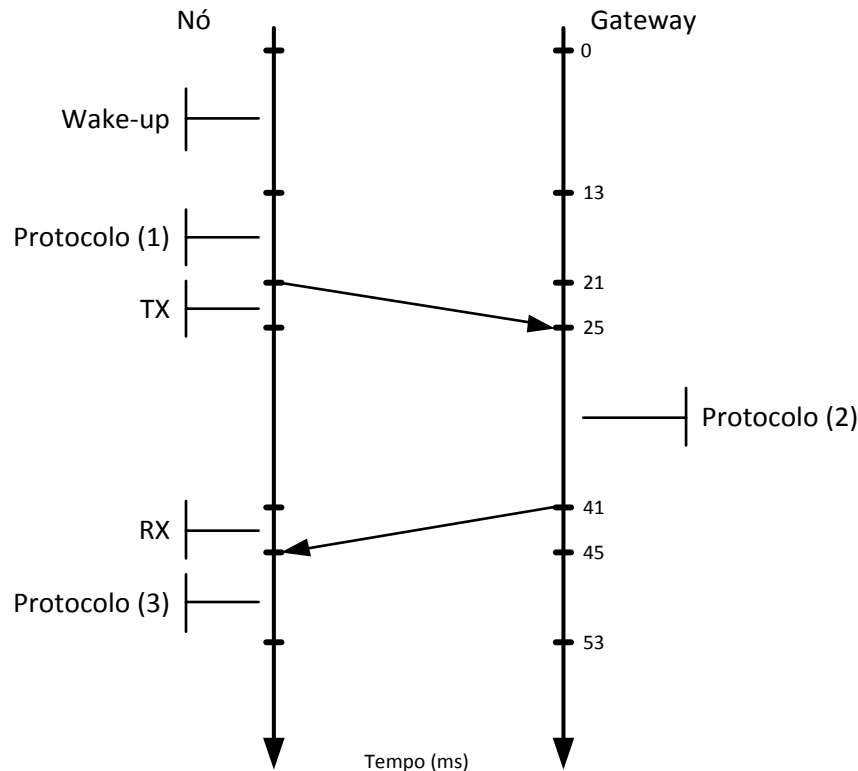


Figura 30 - Diagrama temporal de uma sequência de comunicação

### 4.3.2. Ocupação do canal e colisões

Com base na análise anterior, em que se estima a duração de um processo de troca de informação entre o nó e o *gateway*, faz-se agora um estudo quanto à ocupação do canal e mais propriamente à probabilidade de ocorrência de colisões, de acordo com certos parâmetros da rede.

Para tal recorre-se a um programa de simulação desenvolvido para este propósito. O Anexo B explica o funcionamento e utilização do mesmo. Em suma, o que se faz é simular o canal de transmissão de uma rede com um número variável de nós e com um período entre transmissões também variável, durante uma duração definida pelo número de mensagens a transmitir. O período visto anteriormente para a troca de mensagens entre nó e *gateway* é considerado como sendo o período efetivo de ocupação de canal, ou seja, o período em que o início de um novo processo de troca de mensagens, por outro nó, resulta numa colisão. Com esta simulação do canal, podem-se averiguar as mensagens que são entregues com sucesso e aquelas que colidem, e fazer o cálculo das probabilidades destes acontecimentos.

## Proposta de Gateway para Redes de Sensores Sem Fios

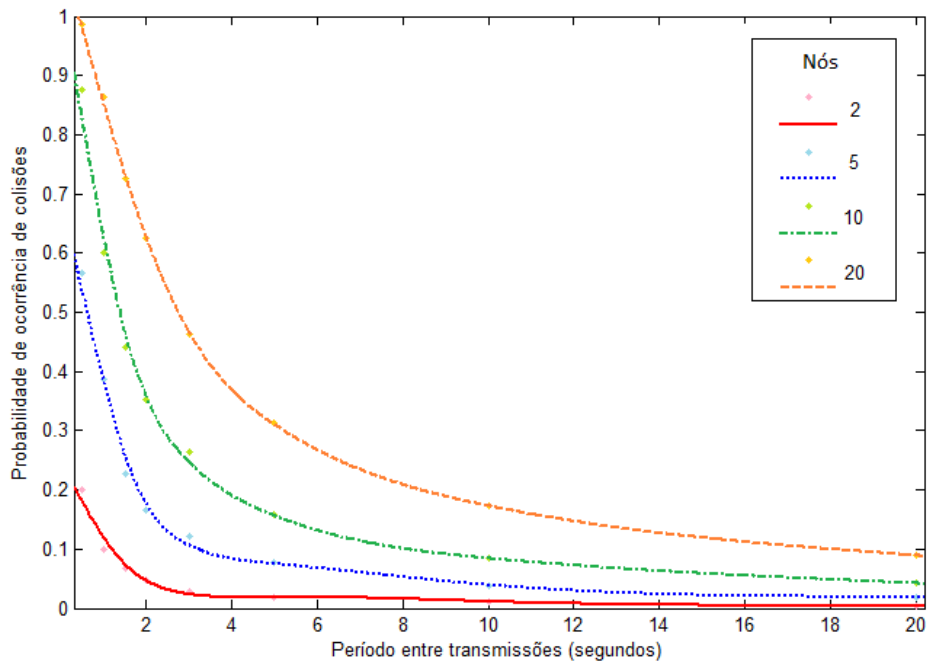


Figura 31 - Probabilidade de colisão em função do período entre transmissões

O gráfico traduz então a probabilidade de colisão em função do período de envio de mensagens e para diferentes tamanhos da rede (número de nós). O período de ocupação de canal utilizado é de  $50ms$ , que se aproxima da estimativa calculada de  $53ms$ . Por interpretação deste, pode-se afirmar que, com o aumento do período entre transmissões, a probabilidade da ocorrência de colisões decresce exponencialmente.

Importa também representar esta probabilidade em função do tamanho da rede para averiguar como esta se rege sobre tal influência.

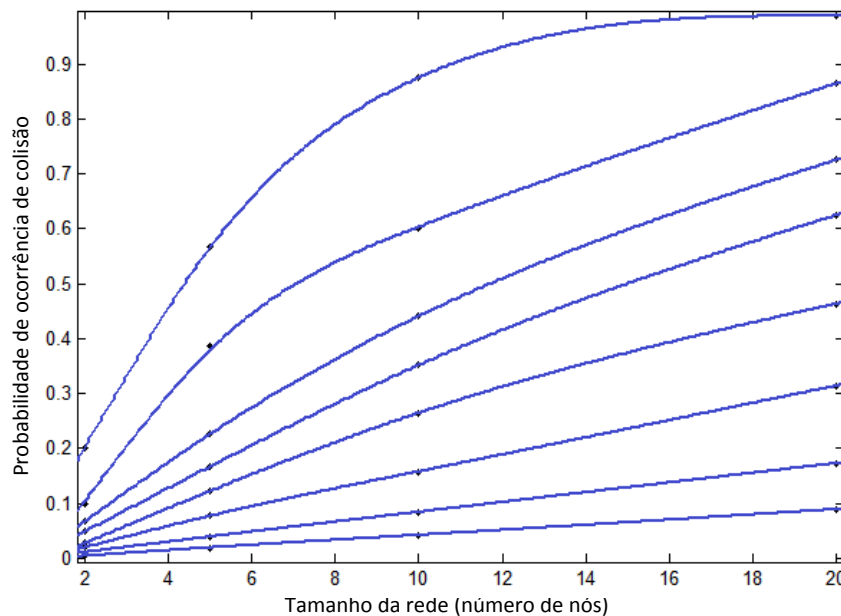


Figura 32 - Probabilidade de colisões em função do tamanho da rede, para diferentes períodos entre transmissões

## Proposta de Gateway para Redes de Sensores Sem Fios

O gráfico representa, portanto, a resposta da probabilidade de colisão com o aumento do número de nós da rede, para diferentes períodos entre transmissões. Observa-se que o crescimento desta probabilidade é logarítmico, com tendência para a probabilidade total de colisões com o crescimento do tamanho da rede. Por uma questão de escala e pela dificuldade em simular redes com tamanhos superiores, devido à duração da simulação, as curvas inferiores não aparentam o crescimento logarítmico.

Esta secção permite justificar a necessidade de introduzir um protocolo de acesso ao meio, que permita aos nós remotos manter a transmissão de informação em tempo real, mas que evite a ocorrência de colisões. Com os resultados vistos, neste esquema de transmissão, se se pretender montar uma rede com 10 nós remotos a transmitir informação a uma frequência de 1Hz, por exemplo, deve-se contar com uma perda de cerca de 60% de todos os pacotes enviados para o meio, como se pode verificar pela análise da Figura 31. Tal é inadmissível do ponto de vista de um controlo eficaz de um processo e do ponto de vista energético, visto que se gasta energia para transmitir informação que será corrompida.

### 4.3.3. Consumo energético

Como foi visto, ao longo do Capítulo 2, uma das dificuldades a ultrapassar neste tipo de redes é a minimização dos consumos energéticos. Tal preocupação pretende maximizar a longevidade energética dos nós remotos que geralmente, e no caso deste projeto, não contam com uma fonte de alimentação fixa, dependendo de baterias.

Nesta secção analisa-se o consumo energético associado ao módulo XBee do nó remoto, que à partida será o principal responsável pelo consumo total da eletrónica do nó. Para tal, tem-se em conta o estudo feito anteriormente relativo aos períodos das etapas relativas à troca de mensagens. A figura seguinte mostra o consumo de corrente durante os períodos calculados anteriormente. Os valores considerados de corrente são retirados do *datasheet* dos módulos XBee e podem ser vistos na Tabela 1. O fabricante não apresenta, no entanto, considerações acerca do comportamento do consumo de corrente nos período de *wake-up*. Para a análise considerou-se que o seu aumento é linear. Entre a transmissão da sua mensagem e a receção da resposta do *gateway* o nó coloca o módulo XBee em Modo *Sleep*. Este modo consome uma corrente superior àquela que consome o modo de *sleep* principal, visto que aqui se pretende um *wake-up* mais rápido.

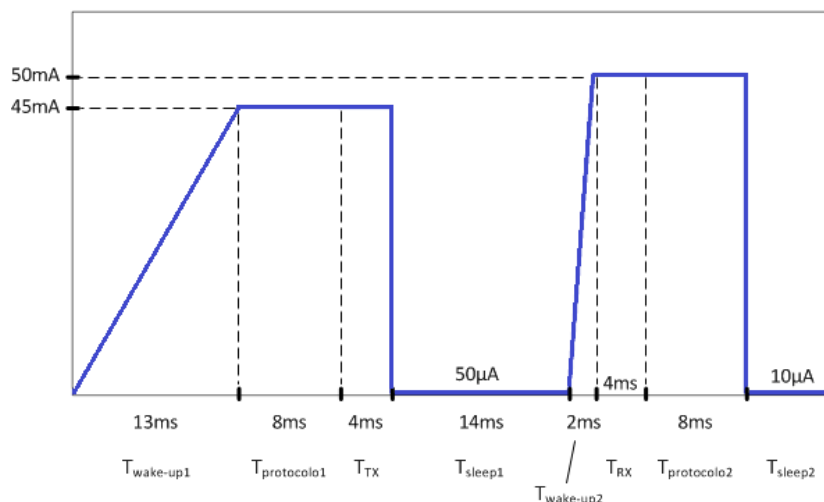


Figura 33 - Esquema do consumo de corrente do módulo XBee

## Proposta de Gateway para Redes de Sensores Sem Fios

Desta forma pode-se fazer o cálculo de uma estimativa do consumo de corrente e prever o mesmo para diferentes períodos entre amostras/comunicações através da Equação 1.

$$\text{Consumo} = \frac{50\mu\text{A} \times T_{\text{sleep}_1} + 25\text{mA} \times T1 + 45\text{mA} \times T2 + 50\text{mA} \times T3 + 10\mu\text{A} \times T_{\text{sleep}_2}}{\text{Periodo}}$$

Equação 1

Com T1, T2 e T3 dados por:

$$T1 = T_{\text{wake-up}_1} + T_{\text{wake-up}_2}$$

Equação 2

$$T2 = T_{\text{protocol}_q} + T_{TX}$$

Equação 3

$$T3 = T_{\text{protocol}_q} + T_{RX}$$

Equação 4

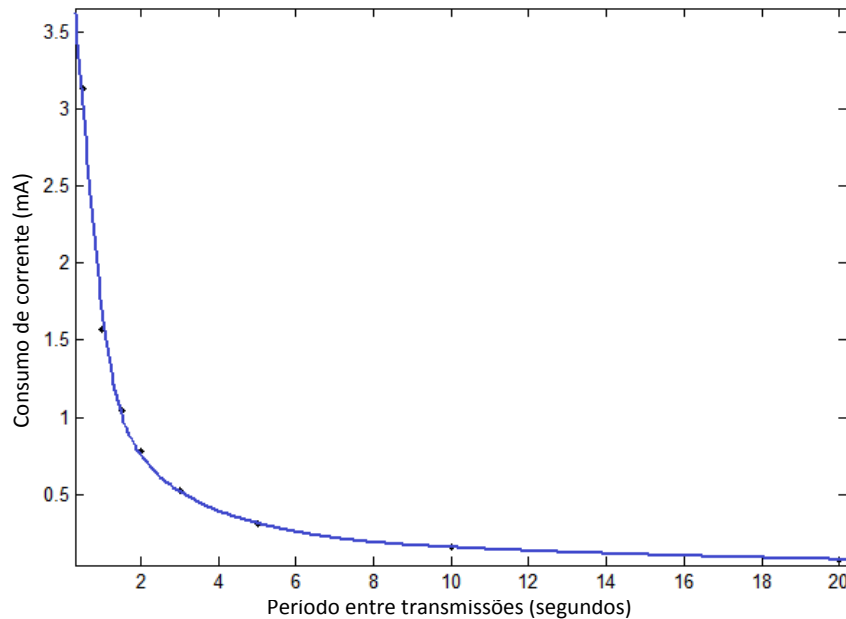


Figura 34 - Consumo de corrente em função do período entre transmissões

Como se pode ver, obviamente, o consumo energético é menor com a diminuição da frequência de amostragem. Com estes dados é possível fazer um estudo prévio da viabilidade da implementação de uma rede, de acordo com os requisitos exigidos a nível energético.

### 4.4. Conclusão

Este capítulo apresentou a forma como deve ser feita a integração dos componentes e estabelecidas as comunicações entre estes. No que diz respeito a estas comunicações os resultados são o seu estabelecimento com sucesso.

Analisaram-se certos aspetos da integração do dispositivo em rede, com principal ênfase na ocupação do canal de transmissão e colisão de mensagens. Os resultados são uma análise através de simulação do comportamento desta, em termos de ocupação de canal e colisão de mensagens, por forma a mostrar a relevância da utilização de um esquema de acesso ao meio e procurar a melhor solução. Realiza-se também uma análise à influência do módulo XBee no consumo energético dos nós remotos, com proposta de modelo de previsão em dependência do período entre transmissões.



## 5. Soluções Propostas

---

Na sequência dos estudos efetuados até este ponto, sugerem-se agora soluções para a implementação da rede tendo em conta os objetivos pretendidos nos campos do esquema de acesso ao meio, *firmware* dos intervenientes e da gestão da mesma. Estas soluções são cruciais tendo em vista a garantia da qualidade da informação, os requisitos temporais exigidos e a alta eficiência energética.

### 5.1 Controlo do acesso ao meio

#### 5.1.1. Protocolo de acesso ao meio

Uma das grandes dificuldades de uma rede deste tipo prende-se com o controlo do acesso ao meio, como foi visto previamente. Tal agrava-se com um crescente número de nós constituintes da rede, do aumento da sua frequência de amostragem ou com as exigências do ponto de vista do processo a controlar. O que se apresenta nesta secção é, então, uma proposta de um esquema de controlo de acesso ao meio pelo qual a rede se deve reger.

Tendo em conta tudo aquilo que foi visto até aqui, pretende-se que este esquema permita o envio de informação, por parte dos nós, em tempo real, evitando colisões e com a maior eficiência energética possível. Para tal, uma das soluções possíveis passa por tornar o *gateway* responsável por iniciar as comunicações, enviando um pedido de informação a cada um dos nós, como a seguinte figura apresenta.

## Proposta de Gateway para Redes de Sensores Sem Fios

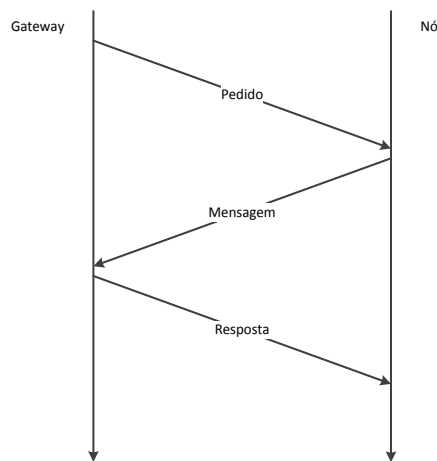


Figura 35 - Esquema por pedido do gateway

Esta solução permite evitar colisões pois um nó só ocupa o canal se tal lhe for requerido. Permite também cumprir requisitos temporais na medida em que o gateway pode requerer a informação por forma a satisfazer as frequências de amostragem de cada um dos nós (dentro de algumas restrições). No entanto, esta implica que o nós têm de estar 'à escuta' do meio para receber o pedido do gateway, assim como rejeitar pedidos que não são a eles destinados. Obviamente, esta solução não é apropriada para os requisitos.

Uma solução poderia passar por acordar o nó remoto instantes antes do momento de chegada de um novo pedido de informação. Ainda assim, existe um período em que não há troca efetiva de informação e a cada troca de mensagens exige-se a receção de uma mensagem de pedido. Esta receção, mesmo num esquema de *wake-up* precisamente prévio ao instante do pedido, representa um aumento do consumo de corrente do módulo XBee de 40% (valor obtido através de análise idêntica àquela feita no capítulo anterior), em relação ao caso estudado previamente para o envio de uma mensagem e a receção de uma resposta. É preciso ter em atenção, pela Tabela 1 e como foi analisado, que 'escutar' o canal consome mais corrente que enviar informação.

A alternativa mais lógica passa, portanto, por fazer com que a comunicação seja iniciada pelos nós remotos. Desta forma, cada um deles pode estar a dormir e acordar exclusivamente para enviar a sua trama e receber a resposta do gateway.

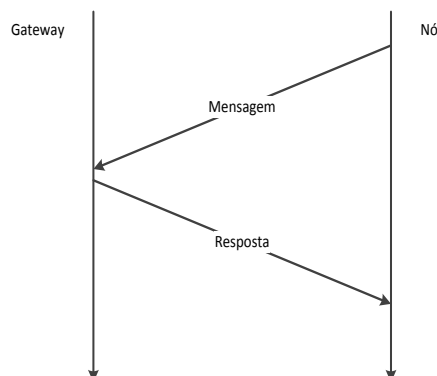


Figura 36 - Esquema por iniciativa dos nós remotos

## Proposta de Gateway para Redes de Sensores Sem Fios

Desta forma maximizamos o tempo em que o consumo energético dos nós é mínimo. No entanto, a rede está agora vulnerável a colisões. Para resolver este problema podia-se recorrer ao mecanismo de CCA. Escutar o canal antes de enviar uma trama permite evitar a grande maioria das colisões. No entanto, implementando este mecanismo não se têm garantias no cumprimento das frequências de amostragem visto que o envio de uma trama é adiado enquanto não existirem condições de transmissão. Tal implica a perda de um controlo em tempo real e, mesmo que tal não seja exigido, pode-se gerar uma outra complicação por aparecimento de nova informação que substituí a que se encontrava em espera para ser enviada.

Conclui-se portanto que, pretendendo cumprir requisitos temporais e evitar a perda de informação, terá de se introduzir um agendamento e reserva de períodos no acesso ao meio. Assim, a solução encontrada passa por reservar previamente um período de tempo para cada um dos nós fazer a troca de informação com o *gateway*, ou seja, adotar uma abordagem de TDMA.

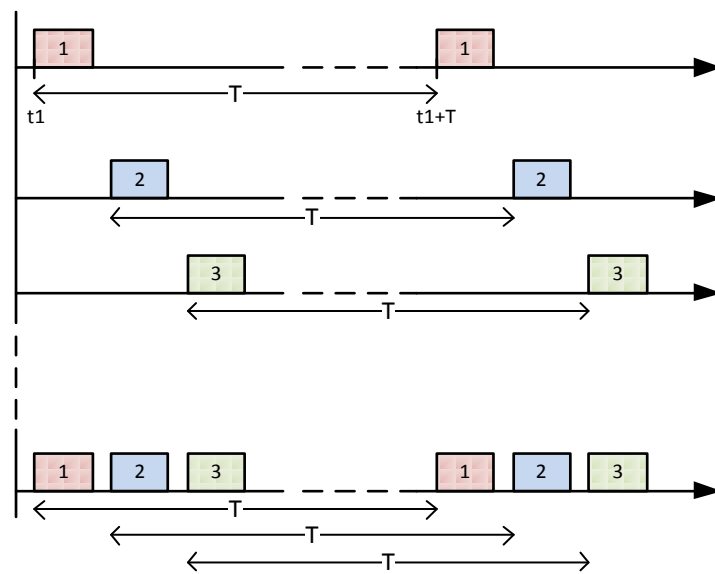


Figura 37 - Esquema de reserva temporal

A figura mostra como o canal é reservado, para as trocas de informação com cada um dos nós, por um determinado período de tempo. Admite-se, por exemplo, que o nó referido como 1 inicia a sua comunicação em  $t1$  e, por forma a cumprir com os requisitos do sistema, repetirá esta comunicação a cada  $T$ . O mesmo se verifica para todos os outros nós. Desta forma evitam-se as colisões e permite-se o controlo de um processo em tempo real e com uma cadência constante para cada um dos nós.

Este esquema possui contudo três limitações óbvias:

- Requer um período entre transmissões comum a todos os nós ou múltiplos inteiros do menor destes períodos. Se tal não acontecer perde-se controlo e eventualmente existirão colisões como se pretende retratar na figura seguinte.

## Proposta de Gateway para Redes de Sensores Sem Fios

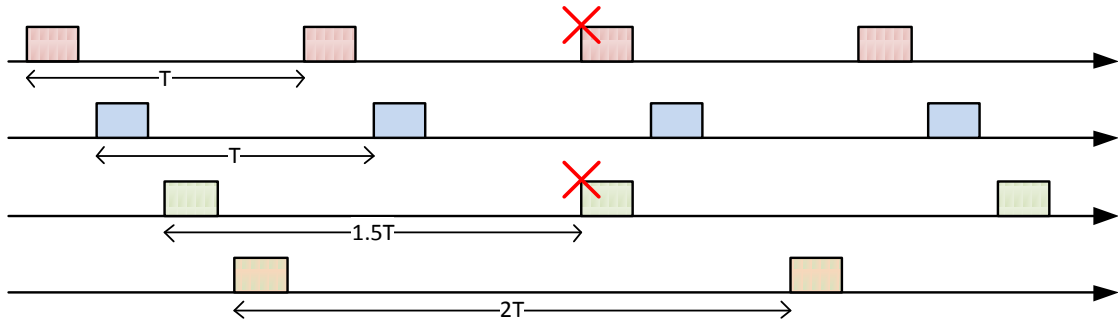


Figura 38 - Limitações do período entre transmissões

- A capacidade da rede em número de nós é, preferencialmente, limitada pela maior frequência de amostragem. Para tal verificam-se quantas trocas de informação podem ser realizadas no menor período entre transmissões. Se tal não for cumprido poderão aparecer colisões ao longo do funcionamento da rede.

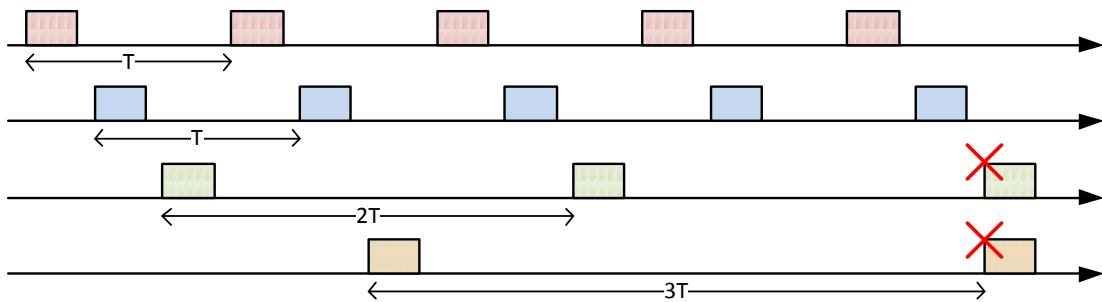


Figura 39 - Limitações à capacidade da rede

Assume-se então, que a capacidade máxima da rede é dada pela parte inteira da seguinte divisão:

$$\text{Capacidade máxima} = \frac{\text{Menor período}}{\text{Intervalo entre comunicações}}$$

### Equação 5

- Admite que os instantes de comunicação são cumpridos. A principal causa para não haver cumprimento destes instantes será a deriva do relógio dos microcontroladores dos nós remotos, responsável por iniciar a comunicação, que leva à dessincronização entre os intervenientes e pode resultar em colisões se não houver correção.

## Proposta de Gateway para Redes de Sensores Sem Fios

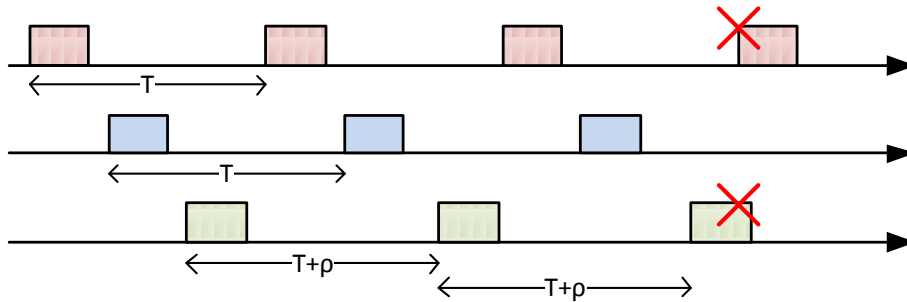


Figura 40 - Influência da deriva do relógio

Para além destas limitações é preciso ainda ter em conta outro pormenor. Como agendar a primeira transmissão de cada nó? Para tal deve-se assumir que após ser ligado, o nó remoto espera por uma primeira mensagem do *gateway*, a qual servirá de referência para as transmissões futuras. Basicamente, a primeira comunicação apresentará um esquema semelhante àquele visto anteriormente.

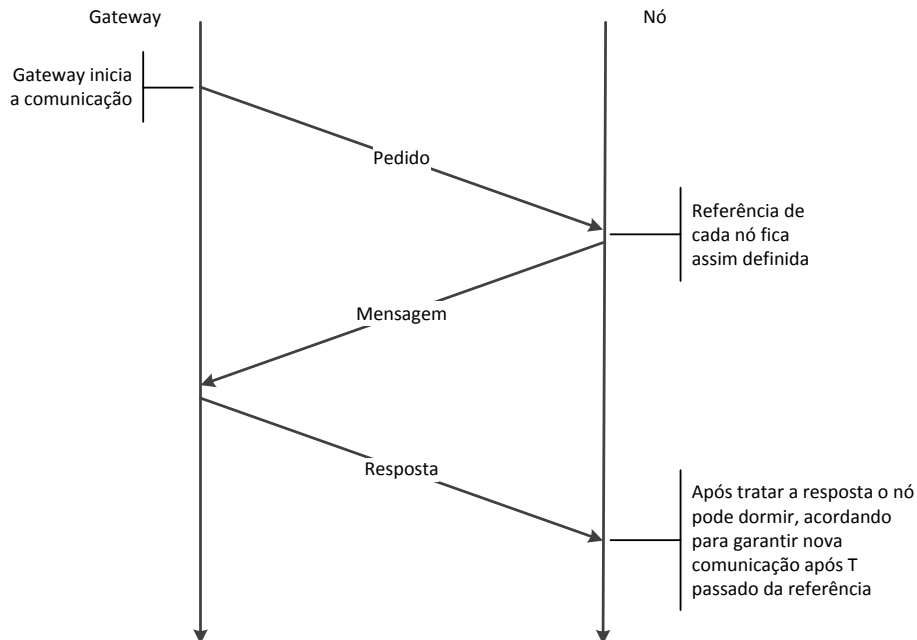


Figura 41 - Primeira comunicação

De acordo com o agendamento feito pelo *gateway*, em pormenor mais à frente, este envia uma mensagem a cada um dos nós para fornecer uma referência para a próxima comunicação. Cada um dos nós tem agora uma referência temporal para iniciar por si as comunicações futuras, que deverão ser feitas após um período  $T$ , desta referência, e pode portanto entrar num estado de adormecimento e acordar para efetuar estas transmissões.

### 5.1.2. Sincronização

Após a escolha do protocolo MAC ter recaído sobre um agendamento dos instantes de comunicação, o maior desafio agora é garantir o sincronismo entre todos os membros da rede por forma a cumprir com os instantes estabelecidos.

## Proposta de Gateway para Redes de Sensores Sem Fios

Como foi dito, se não se mantiver o sincronismo dos membros da rede, em ultimo caso iremos presenciar o aparecimento de colisões entre tramas.

Com base na simulação já apresentada analise-se o caso de uma rede com 10 nós, cada um com uma frequência de amostragem de 1 segundo. Desta feita, a primeira mensagem de cada nó é agendada de acordo com o protocolo apresentado, sendo enviadas mensagens com intervalos de 100ms entre o início de uma transmissão e outra. No entanto, não existe correção da deriva do relógio de cada um destes nós que se encontra no intervalo de 1 a 10ms por segundo.

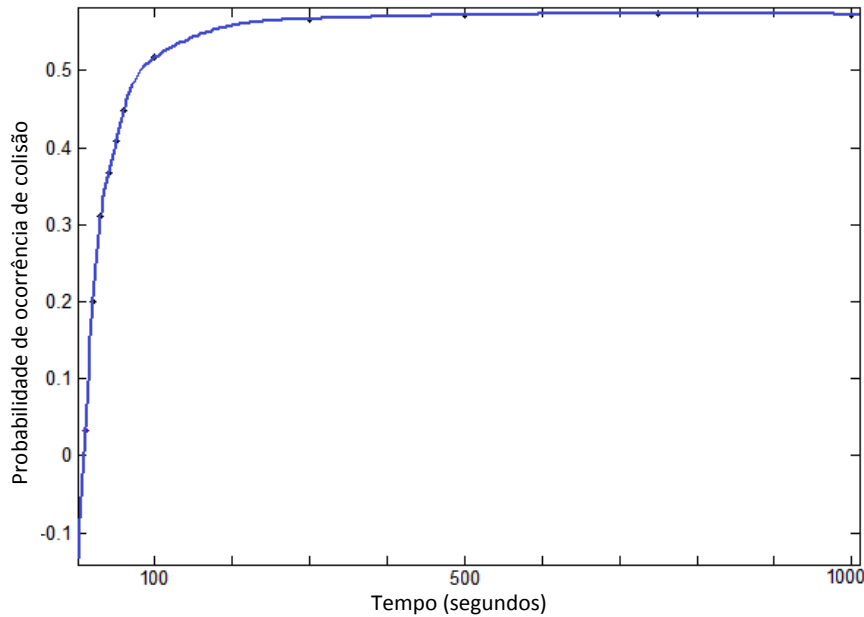


Figura 42 - Probabilidade de erro ao longo do tempo sem mecanismo de sincronização

Como a figura mostra, assiste-se a um aumento da probabilidade de ocorrência de colisões ao longo do funcionamento da rede. Esta probabilidade tende para a que se verifica na Figura 31, de acordo com as mesmas condições. Ou seja, sem correção da dessincronização a rede caminha para um comportamento semelhante ao caso analisado nessa secção.

Para corrigir esta situação implementa-se uma técnica de sincronização, que tira partido das trocas de tramas entre os nós sensores e o *gateway* para aproximar o relógio dos nós ao relógio do *gateway*.

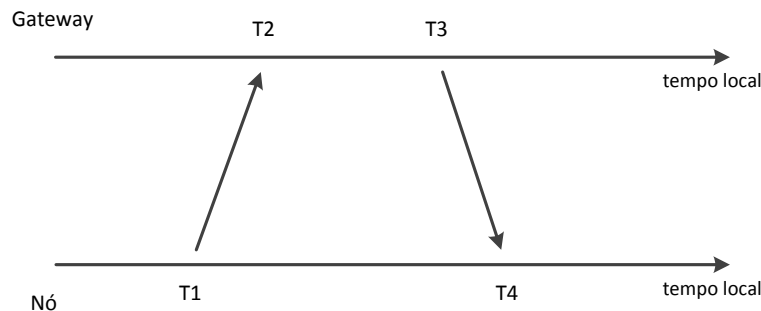


Figura 43 - Esquema de sincronização

## Proposta de Gateway para Redes de Sensores Sem Fios

Consideremos esta troca de mensagens e os tempos apresentados na figura. O nó inicia a sincronização e guarda o valor  $T1$ . O *gateway* recebe esta trama em  $T2=T1+\theta+d$ , em que  $\theta$  é a deriva relativa dos relógios dos intervenientes e  $d$  o atraso de propagação da trama (em relação ao seu próprio relógio). O *gateway* responde, em  $T3$ , com uma trama que contém os valores  $T2$  e  $T3$ . Desta forma o nó remoto pode calcular a deriva do relógio e sincronizar-se com o *gateway*.

$$\theta = \frac{(T2 - T1) - (T4 - T3)}{2}$$

Equação 6

Com esta deriva e o seu relógio sincronizado ( $RS$ ), a transmissão seguinte ( $TS$ ) do nó é agendada para:

$$TS = RS - (T4 - T1) + T - 2\theta$$

Equação 7

Desta forma, compensa-se a deriva existente, o que aproxima a comunicação dos instantes ideais em  $t=t_{inicial}+NT$ .

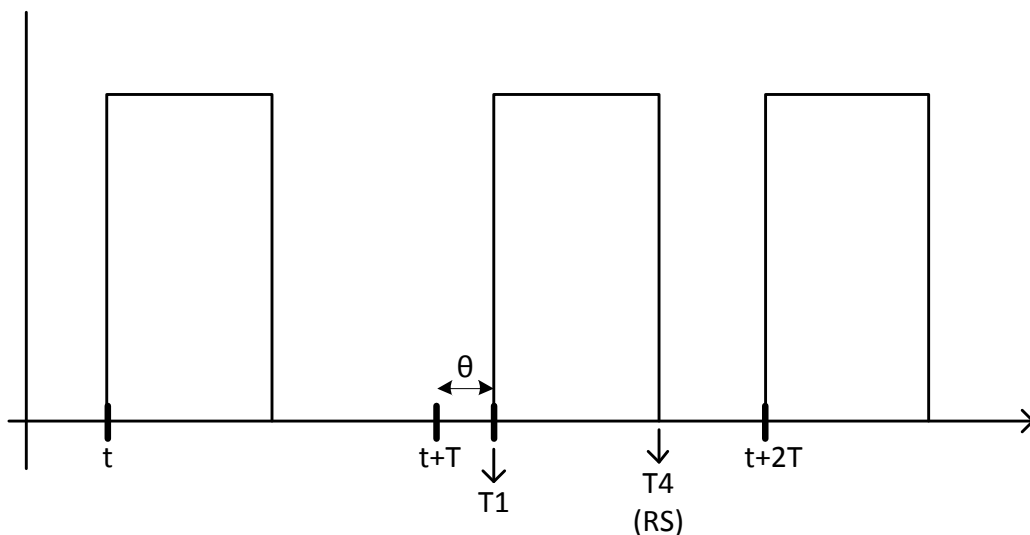


Figura 44 - Compensação da deriva

No entanto, apenas a partir da segunda comunicação é possível retirar o valor da deriva referente a um período, e fazer esta compensação. Assim, pode acontecer, caso as derivas existentes o proporcionem, a ocorrência de colisões nesta segunda comunicação, visto que apenas neste ponto se corrige esta deriva que se cria a cada período, como foi referido. Para contornar esta situação o método CCA é ativado nos nós remotos. Caso o meio esteja ocupado o nó mantém-se acordado até receber uma mensagem do *gateway*. Esta mensagem que o *gateway* envia naquele que é o instante correto na próxima ronda de comunicações, de acordo com o seu agendamento que segue  $t=t_{inicial}+NT$ . Assim, garante-se que não existirá nova colisão e o nó fica agora a conhecer a deriva correspondente a dois períodos, que pode usar para adaptar as comunicações futuras.

## 5.2. Firmware

### 5.2.1. Gateway

Como foi visto o objetivo do *gateway* passa por recolher informação da rede e, ao mesmo tempo, tratar de certos aspetos da sua gestão e organização. Tal implica uma série de decisões e uma corrente de acontecimentos que deverão ser implementados pelo seu *firmware*. Para sua representação recorre-se ao diagrama de *software* seguinte.

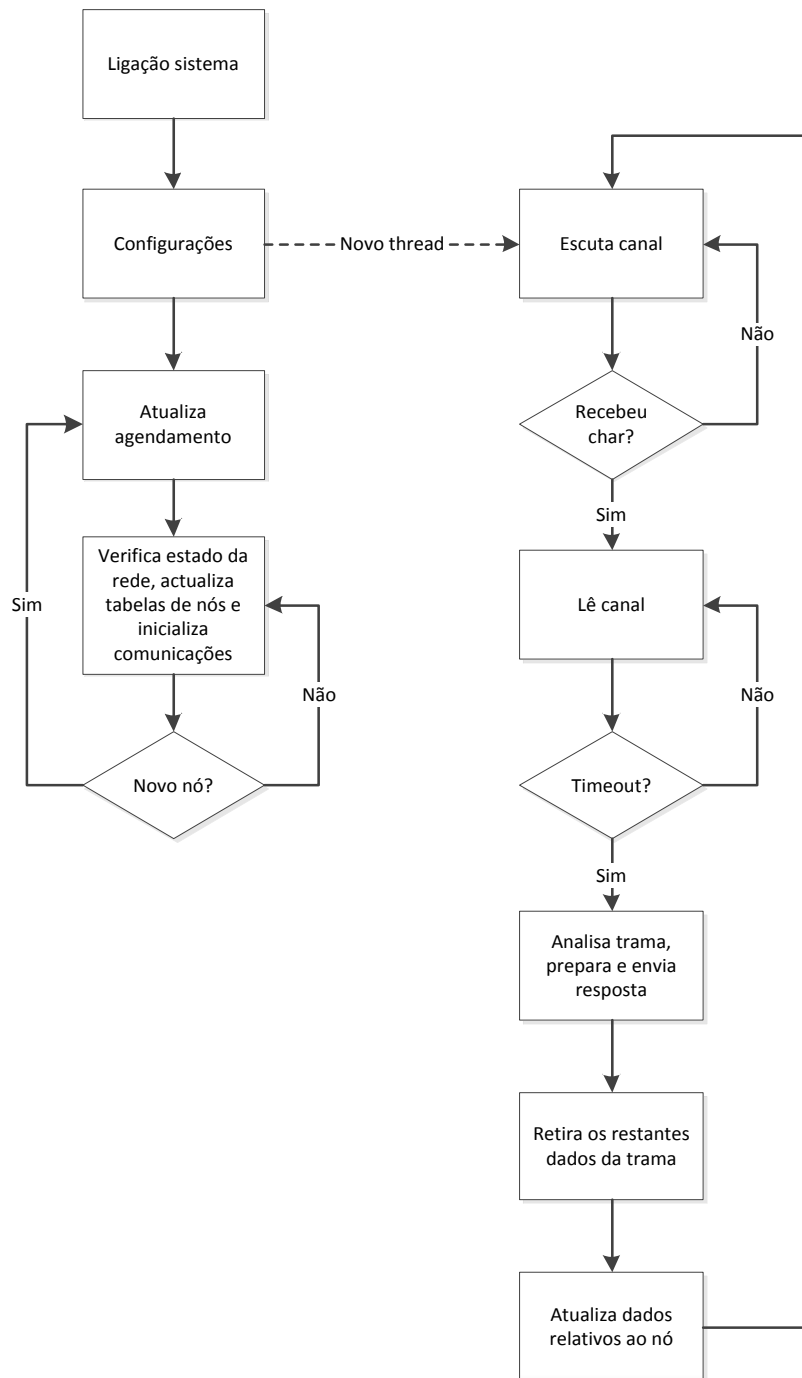


Figura 45 - Diagrama de *software* do gateway



## Proposta de Gateway para Redes de Sensores Sem Fios

No que concerne ao *firmware* do *gateway* é crucial tirar partido das capacidades *multithreading* oferecidas pelo sistema operativo Windows CE. Tal deve-se à necessidade de controlar o canal de forma constante, por forma a garantir que não é perdida informação. Para isso um *thread* é criado paralelo à rotina principal e com maior prioridade para lidar com a chegada de uma trama de forma imediata. Quando a rotina deste *thread* é finalizada, e enquanto não é recebida nova informação, a rotina principal trata de verificar o estado da rede, atualizar as tabelas de nós em relação a falhas e à adição de novos nós e inicializar as comunicações com estes.

### 5.2.2. Nó remoto

No que diz respeito à adaptação dos nós às soluções protocolares encontradas e aqui explicadas, existem certos passos que devem ser cumpridos pelos nós remotos.

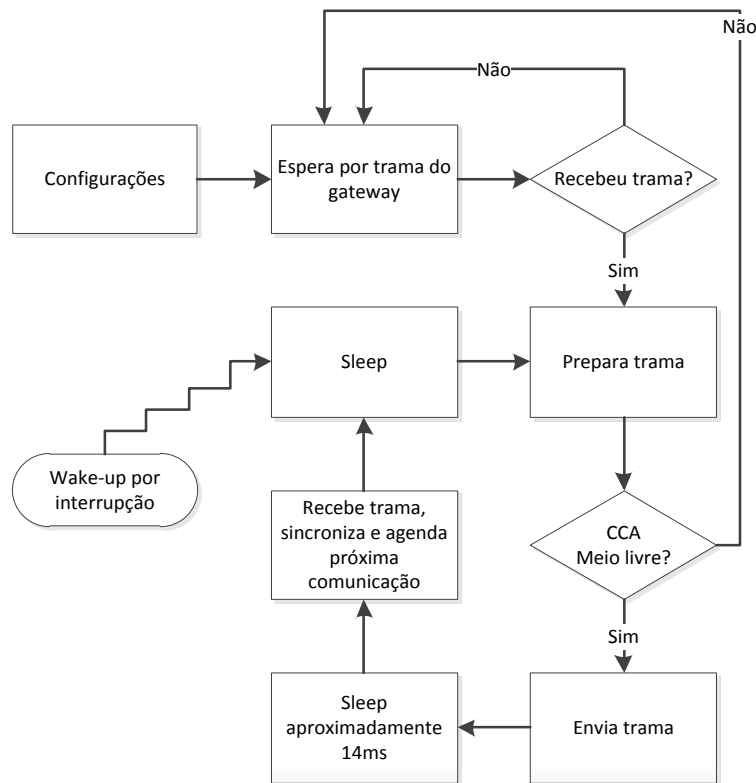


Figura 46 - Diagrama de *software* dos nós remotos

Desta forma, para além das restantes capacidades exigidas ao nó de acordo com o objetivo da implementação, e não olhando aos passos necessários a nível de *software* para implementar estas capacidades, este deverá reger-se pelo diagrama apresentado, por forma a cumprir com os requisitos exigidos no que diz respeito a protocolo de acesso ao meio e de gestão de rede.

Como a figura mostra, após ser ligado, o nó deverá esperar pela trama do *gateway* responsável por criar a referência temporal dos nós remotos. Após a receber, prepara a trama com os dados referentes ao processo e verifica o meio. Caso esteja ocupado espera por nova trama do *gateway*. Caso esteja livre a trama é enviada e, após o módulo XBee ser adormecido durante aproximadamente 14ms, é novamente ligado para receber a resposta do *gateway*, com a qual sincroniza o relógio e agenda a próxima comunicação. O nó pode agora ser adormecido para se

## Proposta de Gateway para Redes de Sensores Sem Fios

atingir o menor consumo energético possível, até ao momento exigido para garantir o instante de envio de novos dados.

### 5.3. Gestão da rede

A gestão da rede cabe ao *gateway* que lida com o escalonamento desta, atualização do agendamento de comunicação, deteção de falhas e manutenção de sincronismo.

Para efetuar o agendamento das comunicações o *gateway* necessita de manter uma tabela atualizada com informações relativas a cada um dos nós. Para tal, para cada nó que se pretenda introduzir na rede, tem de se passar ao *gateway* o seu endereço físico e a sua frequência de amostragem.



Figura 47 - Associação de nós

Mesmo após a entrada em funcionamento da rede, deve ser permitido ao utilizador adicionar novos nós, desde que dentro das especificações e capacidade desta (tendo em conta a análise feita às restrições do protocolo, daí as diferentes indicações apresentadas na Figura 47).

O *gateway* deve também ser capaz de detetar falhas como a não receção de mensagens agendadas. Como foi visto, estas podem estar associadas a falhas na comunicação (que são posteriormente corrigidas) ou perda do nó por avaria ou fim de vida da bateria. Estas informações fazem parte da tabela de nós que deve estar presente no *gateway*.

## Proposta de Gateway para Redes de Sensores Sem Fios

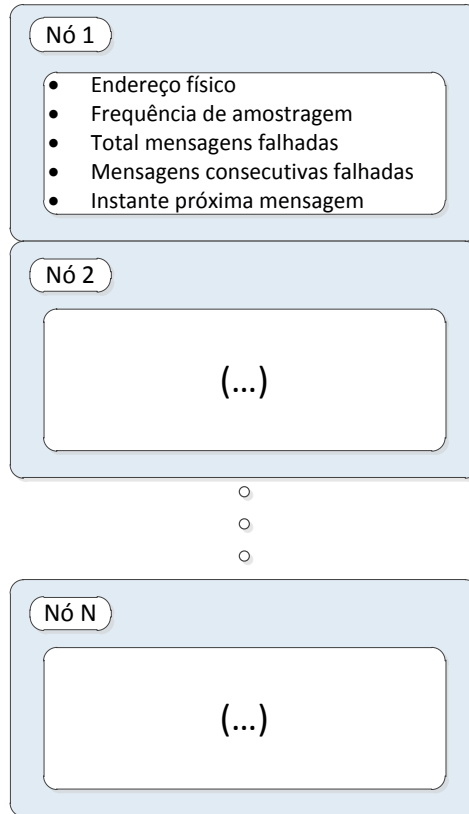


Figura 48 - Tabela de nós

Com a informação presente na tabela, especificamente os endereços e as frequências de amostragem, deve ser realizado o agendamento das comunicações. Para tal primeiro deve ser encontrado o nó com o menor período. Como foi visto este limita a capacidade da rede. É este o primeiro nó a ser instado a comunicar. A partir daí, por ordem crescente de período vai-se construindo a tabela de agendamento.

Tomemos como exemplo uma rede com as seguintes características:

- 6 nós
- Período mais baixo entre transmissões de 1 segundo

Admitindo que se exige um intervalo de  $100ms$  entre início de uma comunicação até ao início de outra, o *gateway* verifica a se estes parâmetros podem ser concretizados.

$$\frac{1s}{6} \geq 100ms$$

Equação 8

Dado que a verificação passa com sucesso este envia uma trama a pedir informação ao nó remoto com o menor período, e passados  $100ms$  envia para outro e assim sucessivamente.

A razão para utilizar um tempo fixo entre inícios de comunicação e não maximizar este valor o mais possível, de acordo com o número de nós presentes, tem duas razões principais. Por

## **Proposta de Gateway para Redes de Sensores Sem Fios**

um lado, a introdução de novos nós torna-se mais simples uma vez que não exige alterações no agendamento dos restantes nós. Simplesmente coloca-se no agendamento uma nova comunicação neste espaço temporal livre. A outra razão prende-se com a 'reserva' de um período para o *gateway* tratar dados, verificar o estado da rede e lidar com os próprios pedidos de atualização da rede.

## 6. Conclusões e Trabalho Futuro

---

Nesta dissertação foi apresentado o estudo para o desenvolvimento e implementação de uma solução para um *gateway*, a ser introduzido numa rede com uma topologia em estrela e baseada na tecnologia Zigbee. Como foi visto, a escolha deste *standard* permite a utilização de uma banda ISM livre e caracteriza-se por equipamentos de baixo consumo energético, fácil programação e com liberdade de edição da camada MAC da pilha protocolar. Foi necessário fazer uma análise acerca do *hardware* disponível para averiguar as dificuldades a ultrapassar, e encontrar as melhores soluções para a integração e a o desenvolvimento de capacidades do dispositivo a implementar.

Este *hardware* inclui a placa de desenvolvimento FriendlyARM, desenhada para a implementação de sistemas embutidos, e cujas capacidades computacionais e características físicas, mais propriamente as interfaces exteriores que possui, a tornam uma boa solução para os objetivos propostos. O outro componente de *hardware* que faz parte do dispositivo *gateway*, e dos próprios nós sensores constituintes da rede, é o módulo XBee. Estes módulos baseiam-se no protocolo IEEE 802.15.4, e permitem a implementação de normas ZigBee para as camadas protocolares superiores.

Após uma análise geral destes dispositivo, e também da sua arquitetura interna, testou-se a integração estabelecendo-se as vias de comunicação necessárias entre os componentes. Para tal foram desenvolvidas rotinas e efetuados testes, recorrendo ao ambiente de desenvolvimento Visual Studio 2005 da Microsoft e outras ferramentas. Tanto a comunicação entre FriendlyARM e módulo XBee, como a comunicação entre módulos XBee foi estabelecida com sucesso. Em relação à comunicação entre módulos XBee os dois modos de funcionamento suportados, transparente e API, foram estudados e testados. Também a comunicação FriendlyARM com a Internet foi estabelecida e testada, assim como a solução Web Server que permite a um utilizador remoto aceder a dados do dispositivo.

Tendo em conta os requisitos exigidos analisou-se o esquema de uma troca de mensagens entre nó e *gateway*. Isto permitiu simular o canal de comunicações e estudar a probabilidade de ocorrência de colisões, tendo-se concluído que, para redes relativamente densas, a perda de pacotes torna-se incontrolável, caso não seja aplicado nenhum esquema de controlo de acesso ao meio. Permitiu, também, obter uma equação, com a qual se pode prever o consumo de corrente do módulo XBee, de acordo com os requisitos da rede, em função do período entre transmissões.

## Proposta de Gateway para Redes de Sensores Sem Fios

Sugere-se, então, um esquema de controlo de acesso ao meio baseado em agendamento e reserva de períodos temporais para cada um dos nós transmitir as suas informações. Este esquema permite as comunicações em tempo real, com uma cadência constante, e evita colisões. No entanto, como foi visto por simulação, se não se mantiver a sincronização dos relógios dos intervenientes, a rede adota um comportamento, em termos de probabilidade de colisões, que tende, ao longo do tempo, para uma rede sem controlo no acesso ao meio. Para corrigir esta situação implementa-se uma técnica de sincronização dos relógios dos nós remotos, que tira partido da troca de mensagens existente entre estes e o *gateway*. Acertam-se, assim, os relógios dos nós a cada comunicação, compensando ao mesmo tempo as derivas existentes, para que as comunicações sejam efetuadas na janela temporal esperada.

Por fim, faz-se referência ao *firmware* que deverá ser implementado, tanto nos nós remotos como no *gateway*, através da apresentação de esqueletos de diagramas de *software*. São ainda referidos certos aspetos relativos à gestão da rede.

Através do trabalho efetuado espera ter-se reunido um conjunto de conhecimentos e propostas relevantes de soluções para problemas, que possam ser utilizados tanto por futuros colegas como pela empresa com a qual existiu a parceria, criando não só valor intelectual como eventual valor comercial. Criaram-se ferramentas de simulação e protótipos de rotinas que podem ser usados e melhorados em trabalho futuro. Para complementar o trabalho podem também ser realizadas as seguintes tarefas:

- Implementação de rotinas com base no trabalho realizado
- Construção de rede e implementação e teste do protocolo sugerido, assim como da técnica de sincronização
- Realização de testes e comparação com resultados obtidos por simulação
- Criação de interface gráfica para gestão e verificação do estado do *gateway* e da rede
- Explorar outras formas de enviar dados e receber comandos através da Internet
- Exploração das capacidades oferecidas pelo *standard* ZigBee nas camadas protocolares superiores

# Bibliografia

---

- [1] A. Goldsmith, *Wireless Communications*, Cambridge University Press, 2005.
- [2] "Semaphore line," [Online]. Available: [http://en.wikipedia.org/wiki/Semaphore\\_line](http://en.wikipedia.org/wiki/Semaphore_line). [Acedido em Outubro 2011].
- [3] C. f. B. History, "Ericsson: The History of Wireless Communication," 2001. [Online]. Available: <http://www.youtube.com/watch?v=X5jPoQzEh-M>. [Acedido em Novembro 2011].
- [4] J. M. Shea, "History of Wireless Communication," Janeiro 2000. [Online]. Available: <http://wireless.ece.ufl.edu/jshea/HistoryOfWirelessCommunication.html>. [Acedido em Novembro 2011].
- [5] T. Seymor e A. Shaheen, "History of Wireless Communication," *Review of Business Systems*, vol. 15, Novembro 2011.
- [6] N. S. Networks, *The building block of Mobile broadband - Celebrating GSM 20 years*, 2011.
- [7] "Wireless LAN," [Online]. Available: [http://en.wikipedia.org/wiki/Wireless\\_LAN](http://en.wikipedia.org/wiki/Wireless_LAN). [Acedido em Novembro 2011].
- [8] C.-y. Chong e S. P. Kumar, "Sensor Networks: Evolution, Opportunities and Challenges," *Proceedings of the IEEE*, vol. 91, pp. 1247 - 1256, August 2003.
- [9] J. Zheng e A. Jamalipour, *Wireless Sensor Networks, A Networking Perspective*, Wiley, 2009.
- [10] H. Karl e A. Willig, *Protocols And Architectures for Wireless Sensor Networks*, Chichester: Wiley, 2005.
- [11] I. F. A. a. M. C. Vuran, *Wireless Sensor Networks*, Wiley, 2010.
- [12] C. Lynch e F. O' Reilly, "Processor Choise For Wireless Sensor Networks," *Workshop on Real-World Wireless Sensor Networks*, pp. 1-5, 2005.
- [13] T. Instrument's, *MSP430 Family User's Guide*, 2004.
- [14] J. S. Wilson, *Sensor Technology Handbook*, Oxford : Elsevier Science, 2004.
- [15] G. J. Pottie e W. J. Kaiser, "Wireless Integrated Network Sensors," *Communications of the ACM*, vol. 43, pp. 51-58, May 2000.
- [16] I. C. Society, *IEEE Std 802.11*, New York, 2007.
- [17] B. S. I. Group. [Online]. Available: <https://www.bluetooth.org/Building/HowTechnologyWorks/Architecture/Radio.htm>.

[Acedido em Novembro 2011].

- [18] I. 8. S. Committee, Janeiro 2010. [Online]. Available: <http://www.ieee802.org/15/pub/TG4.html>. [Acedido em Novembro 2011].
- [19] "ARM," [Online]. Available: [http://en.wikipedia.org/wiki/ARM\\_architecture#cite\\_note-4](http://en.wikipedia.org/wiki/ARM_architecture#cite_note-4). [Acedido em Maio 2012].
- [20] M. Systems, *MINI2440 User's Manual*, 2009.
- [21] D. International, *XBee/XBee-PRO RF Modules Product Manual*, 2009.
- [22] R. V. Aroca e G. Caurin, "A Real Time Operating Systems (RTOS) Comparison," EESC - USP, São Paulo, 2009.
- [23] S. Pavlov e P. Belevsky, *Windows Embedded CE 6.0 Fundamentals*, Microsoft, 2008.
- [24] D. I. - K. B. Article, "Sending data through an 802.15.4 network latency timing," [Online]. Available: <http://www.digi.com/support/kbase/kbaseresultdetl?id=3065>. [Acedido em Abril 2012].
- [25] H. Zimmermann, "OSI Reference Model - The ISO Model of Architecture for Open Systems Interconnection," *IEEE Transactions on Communications*, vol. 28, 1980.



# Anexo A

---

Neste anexo apresenta-se o protótipo da aplicação criada para testar a comunicação, via porta série, entre a placa FriendlyARM e o módulo XBee.

```
// Protótipo de aplicação para estabelecer a comunicação entre a placa FriendlyARM e o módulo XBee. Abre o driver COM3 e cria um handle (hCOM3) para ser usado pelas funções setupComm, sendBytes e getBytes.
// setupComm-Configura os parâmetros da ligação série
// sendBytes-Envia um número de bytes pela porta série do buffer de transmissão
// getBytes-Recebe um número de bytes pela porta série e colocando buffer de transmissão
```

```
#include "stdafx.h"
```

```
#include <windows.h>
```

```
#include <commctrl.h>
```

```
HANDLE hCOM3;
```

```
DWORD dwError, dwSize, dwRead, dwWritten, dwCommEvent;
```

```
BYTE RX_buffer[109]; // Buffer de recepção
```

```
BYTE TX_buffer[109]; // Buffer de transmissão
```

```
// Funções //
```

```
bool setupComm() //Configura parâmetros da ligação série
```

```
{
```

```
    DCB PortDCB; //Define a estrutura DCB (Device Control Block)
```

```
    PortDCB.DCBlength = sizeof(DCB);
```

```
    GetCommState(hCOM3,&PortDCB);
```

```
    PortDCB.BaudRate=115200; //Baudrate
```

```
    PortDCB.fParity=PARITY_NONE; //Verificação de paridade
```

```
    PortDCB.ByteSize=8; //Número de bits por byte
```

```
    PortDCB.StopBits=ONESTOPBIT; //Stop bits
```

```
    //Configura a comunicação de acordo com os parâmetros
```

```
    if(!SetCommState(hCOM3,&PortDCB))
```

```
    {
```

```
        dwError=GetLastError ();
```

```
        MessageBox(NULL,TEXT("Erro ao configurar a comunicação série"),TEXT("Error!"),MB_OK);
```

```
        return FALSE;
```

```
    }
```

```
    COMMTIMEOUTS commTimeOut;
```

```
    memset(&commTimeOut,0,sizeof(COMMTIMEOUTS));
```

```
    memset(&RX_buffer,0,10);
```

```
    commTimeOut.ReadIntervalTimeout=1000; //Tempo (ms) para timeout entre a recepção de um byte e outro
```

```
    commTimeOut.ReadTotalTimeoutConstant=0;
```

```
    commTimeOut.ReadTotalTimeoutMultiplier=0;
```

```
    //Configura timeouts da comunicação
```

```

        SetCommTimeouts(hCOM3,&commTimeOut);
    }

int sendBytes(int nBytes) //Envia nBytes da porta série
{
    dwSize=nBytes;
    dwWriten=0;

    if (!WriteFile(hCOM3,&TX_buffer,dwSize,&dwWriten,NULL))
        return 0;

    return 1;
}

int getBytes(int nBytes) //Lê nBytes da porta série
{
    dwSize=nBytes;
    dwRead=0;

    if (!ReadFile(hCOM3,&RX_buffer,dwSize,&dwRead,NULL))
        return 0;

    return 1;
}

void CloseApplication() //Fecha os handles
{
    CloseHandle(hCOM3);
}

int _tmain(int argc, _TCHAR* argv[])
{
    //Abre o driver da porta série e cria o handle
    hCOM3=CreateFile(L"COM3:", GENERIC_READ | GENERIC_WRITE, 0, NULL,
    OPEN_EXISTING, FILE_ATTRIBUTE_NORMAL, NULL);
    if (hCOM3==INVALID_HANDLE_VALUE)
    {
        MessageBox(NULL, L"Erro ao abrir o driver da porta série\n", L"ERROR",
        MB_OK);
        return -1;
    }

    setupComm();

    //
    // (...)
    //

    CloseApplication();

    return 0;
}

```

# Anexo B

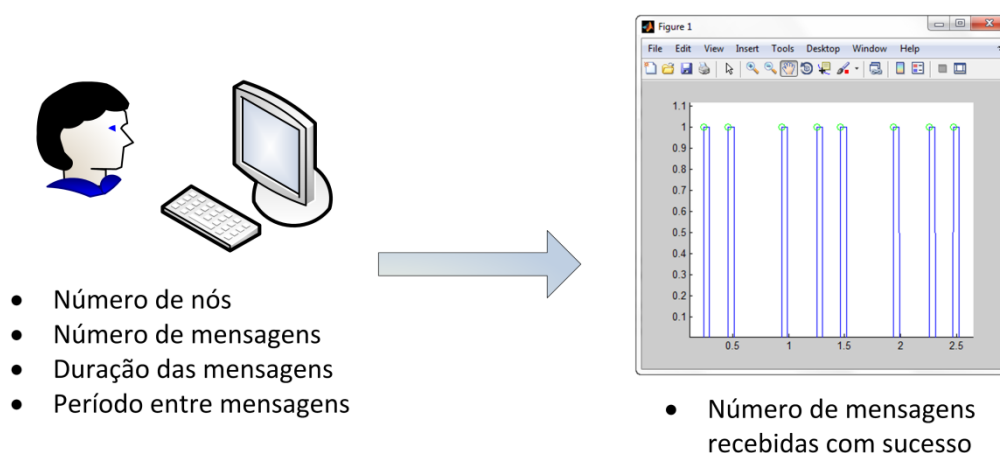
---

Foi criada uma ferramenta em MATLAB® para simular o estado do canal de transmissão tendo em conta certos parâmetros da rede e dos nós constituintes por forma a permitir explorar as capacidades e limites de uma dada rede.

Assim, a ferramenta criada permite ao utilizador definir:

- Tamanho da rede
- Número de mensagens a ser enviada por cada nó durante a simulação
- Duração das tramas
- Período entre mensagens, ou seja, a frequência de amostragem dos nós da rede

Tendo em conta estes parâmetros a ferramenta calcula o estado do canal de transmissão ao longo do tempo e conta o número de mensagens recebidas com sucesso (aquelas que não sofrem colisões com outras durante a sua transmissão).

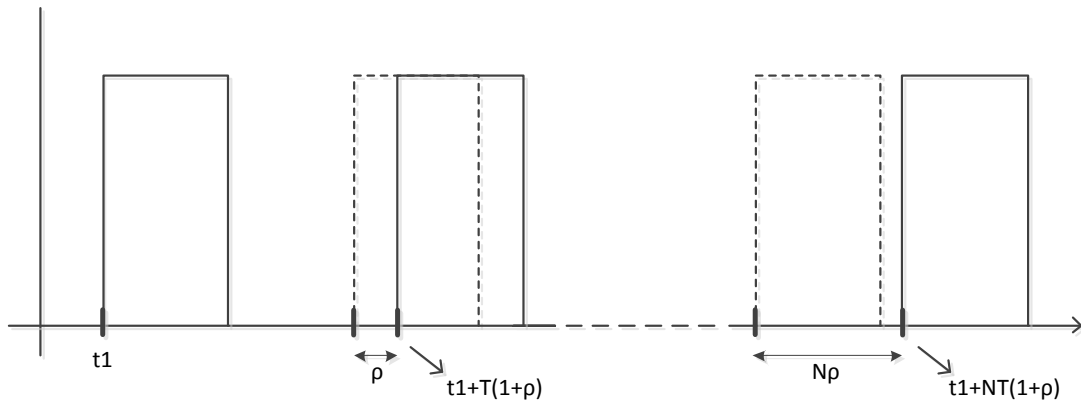


**Figura B1 - Esquema do funcionamento geral do simulador**

## **Funcionamento**

O canal de transmissão é simulado com uma escala temporal de milissegundos e não admite uma ordem particular no envio das mensagens dos nós, ou seja, durante o primeiro período qualquer nó pode enviar a sua primeira mensagem em qualquer altura. Tal pretende simular uma rede deixada a si própria durante um período de tempo definido pelo número de mensagens a enviar.

Uma outra variável deve ser levada em conta nesta simulação para melhor traduzir o comportamento expectável da rede real. Essa variável é o *drift* do relógio dos microcontroladores responsáveis pelos nós remotos e consequentemente pelo agendamento da comunicação.

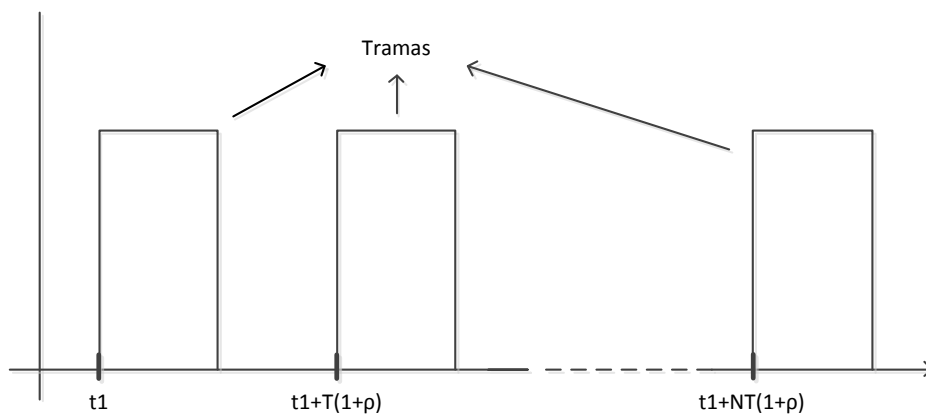


**Figura B2 - Influência da deriva do relógio**

A figura mostra como este *drift*, representado por  $\rho$  e que corresponde ao *drift* por período, influencia os instantes de transmissão das tramas ao longo do tempo. À medida que nos deslocamos no tempo as transmissões de tramas afastam-se cada vez mais dos instantes previstos para a sua ocorrência em  $t1+NT$ , sendo  $N$  o número da mensagem.

Os *drifts* associados a cada um dos nós é aleatório e está compreendido entre 0 e 10 milissegundos por segundo.

Assim, com o instante de envio da primeira trama, o período, o *drift* e a duração das tramas é possível simular a ocupação do canal de transmissão com respeito a cada um dos nós.



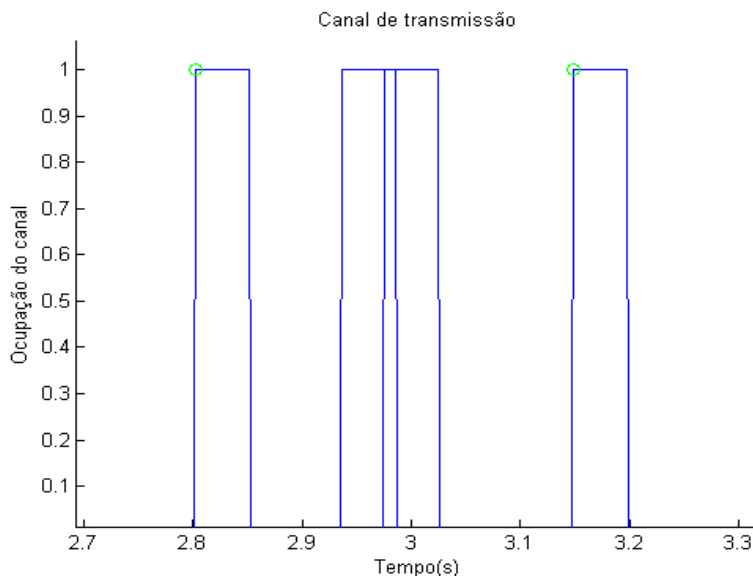
**Figura B3 - Simulação do canal**

Após estes passos resta identificar quais as mensagens que não sofrem colisões durante a sua transmissão. Para tal, cada um dos vetores correspondentes ao estado do canal, para cada um dos nós, é comparado com a soma de todos esses vetores para verificar se existem outros nós a ocupar o canal, quando um deles está a enviar a sua trama.

### **Outputs da simulação**

Do ponto de vista de *outputs* da simulação, aquilo que mais interessa ao utilizador é visualizar o estado do canal de comunicação ao longo do tempo e contabilizar o número de tramas que não sofrem colisões durante a sua transmissão.

Desta forma, após a simulação, é apresentada uma figura onde se pode ver, ao longo do tempo, a utilização do canal. Nesta são visíveis as tramas correspondentes aos diferentes nós assim como uma sinalização que indica que a trama não sofreu colisões. Na figura exemplificativa seguinte pode ver-se uma parcela do canal de transmissão. Neste intervalo constata-se que existem duas tramas que não sofrem qualquer interferência e são por isso sinalizadas como sendo recebidas com sucesso e duas tramas que colidem e portanto são descartadas.



**Figura B4 - Resultado da simulação**

As tramas sinalizadas são assim contabilizadas e apresentadas ao utilizador para com estas poder fazer o cálculo de probabilidades, por exemplo.

### Utilização

Para usar este simulador é necessário o script MATLAB ChannelSimulation (.m) e as funções NodeResponse (.m) e NormalizeStatus (.m).

A descrição do objetivo destas funções, que também se encontra no cabeçalho dos respetivos ficheiros é:

- NodeResponse - Esta função computa a resposta do nó ao longo do tempo tendo em conta os parâmetros definidos. Para tal são lidos os parâmetros: número de mensagens, período das mensagens, *drift* a cada período, instante da primeira mensagem e duração das mensagens. Recebe ainda uma *flag* para fazer ou não o *plot* desta resposta
- NormalizeStatus - Esta função 'estica' cada vetor que contém a resposta correspondente a cada nó com zeros até ao tamanho do maior vetor deste tipo. Isto é feito para que estes vetores tenham as mesmas dimensões e possam ser somados. Para tal a função recebe o vetor com a resposta e o tamanho do maior destes vetores.

Trabalhar com o script criado implica interagir apenas com seis variáveis presentes no cabeçalho do script sob o nome 'Variables and Control'. Estas variáveis são:

- N - Número de nós da rede
- NMsgs - Número de mensagens a serem enviadas por cada nó
- MsgLength - Duração das mensagens em milissegundos
- Period - Período entre mensagens até à décima de segundo
- Plot - *Flag* que sinaliza o *plot* ou não do estado do canal de transmissão. Plot=1 corresponde a sim e Plot=0 corresponde a não
- NoR - Número de vezes que a simulação é feita

Deve-se ter em conta que a simulação de redes muito grandes ou onde se pretende uma troca de um grande número de mensagens levam a tempos de simulação elevados. Também o desenho do canal de comunicação aumenta o tempo de simulação, especialmente nas condições anteriormente descritas.

# Anexo C

---

Neste anexo apresentam-se algumas informações relativas a estudos realizados, em torno do sistema operativo, nomeadamente no que diz respeito a *threads* e funções associadas ao relógio, com vista a implementação do dispositivo.

## *Threads*

Como foi visto, as capacidades *multitasking* do sistema operativo terão de ser utilizadas na implementação do dispositivo. Tal deve-se, principalmente, à necessidade de controlar, de forma permanente, o canal de comunicação, verificando se surge informação para que não haja perda da mesma. Para tal, apresentam-se as seguintes funções que dizem respeito à criação de um *thread* e associação de uma rotina a este.

```
// Protótipo de aplicação para criação de thread e associação de rotina de processamento

#include "stdafx.h"
#include <windows.h>
#include <commctrl.h>

HANDLE hThread;

// Rotina do thread
void ThreadProc(void)
{
    (...)
}

void CloseApplication() // Fecha os handles
{
    CloseHandle(hThread);
}

int _tmain(int argc, _TCHAR* argv[])
{
    // Cria uma nova thread
    hThread = CreateThread(NULL, 0, (LPTHREAD_START_ROUTINE)ThreadProc, 0, 0, NULL);
    if (NULL == hThread)
    {
        MessageBox(NULL, L"Erro ao criar thread", L"ERROR", MB_OK);
        return -1;
    }

    //
    //(...)
    //

    CloseApplication();

    return 0;
}
```

No seguimento deste pensamento, poderá ser necessário considerar as prioridades dos diferentes *threads* criados. Por exemplo, a rotina de escuta do canal deverá ser prioritária às rotinas de gestão da rede. O sistemas de suporte *multitasking* tem 256 níveis de prioridades sendo que o nível zero é aquele com maior prioridade. Para alterar a prioridade dos diferentes *threads* pode-se usar a função `SetThreadPriority`.

Outra função importante para a implementação é a função `WaitCommEvent`. Como o nome indica, esta função espera por um evento de comunicação associado a um dado dispositivo de comunicação. Desta forma, para escutar o canal, por exemplo, cria-se um *thread* cuja rotina de processamento espera por uma comunicação, referente ao *handle* criado para a porta série, através desta função.

// Rotina que espera por comunicação

```
void ThreadProc(void)
{
    DWORD dwCommEvent;

    while(1)
    {
        dwCommEvent = EV_RXCHAR; // O evento é a receção de um carater
        WaitCommEvent(hCOM3, &dwCommEvent, NULL);
        //
        //(...)
        //
    }
}
```

## Relógio

Outra das questões diz respeito à implementação do relógio do *gateway*. Para este fim analisaram-se qualitativamente diferentes soluções baseadas em funções disponibilizadas pelo sistema operativo.

	Precisão	Resolução
<code>GetSystemTime</code>	√	×
<code>SetTimer</code>	×	×
<code>GetTickCount</code>	×	√

Tabela C1 - Soluções implementação do relógio

Em algum pormenor pode-se fazer uma análise a estas funções:

- A função `GetSystemTime` usa o RTC (Real-Time Clock) da placa e graças a isso é muito precisa na manutenção do tempo do sistema. No entanto, não permite uma resolução superior a um segundo.
- A função `SetTimer` cria um *timer* com um período definido pelo utilizador. Esta função mostrou-se pouco precisa na medida em que acaba por gerar um atraso ao longo do tempo. Também a implementação de um *timer* com vista a resoluções da ordem das unidades de milissegundos se traduziu em comportamentos erróneos do sistema operativo.



- A função `GetTickCount` informa do tempo passado, em milissegundos, desde que o sistema é iniciado. Também esta função cria uma deriva temporal em relação ao tempo real.

Do ponto de vista de implementação, com esta análise, pode-se concluir que:

- Se o utilizador admitir que a deriva temporal existente não é determinante no funcionamento do sistema, a solução indicada passa pela utilização da função `GetTickCount`.
- Caso se pretenda manter o relógio acertado uma solução é a utilização conjunta da função `GetSystemTime` e da função `GetTickCount`, aliando a precisão do RTC de uma com a resolução em milissegundos de outra. Isto implica sincronizar periodicamente o momento da passagem de um segundo da função `GetSystemTime` com a contagem de milissegundos, mantendo a deriva controlada.
- O utilizador pode ainda optar por criar os seus próprios *drivers* para implementar o relógio do *gateway*.