FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Comunicação WiFi para monitorização móvel de sinais fisiológicos

João Tiago Teixeira Mesquita

Mestrado Integrado em Engenharia Eletrotécnica e de Computadores

Orientador: Luís Miguel Pinho de Almeida (Prof. Dr.)

24 de Julho de 2018

Resumo

Na Europa, a esperança média de vida à nascença cresceu nos últimos 20 anos cerca de 14%, esse incremento e o consequente envelhecimento generalizado da população mundial traz consigo um conjunto de novos desafios. Nomeadamente acarreta a necessidade de encontrar soluções inovadoras e de baixo custo que facilitem a monitorização contínua de doentes dentro e fora do meio hospitalar, reduzindo custos aos sistemas de saúde e aumentando a qualidade de vida desses doentes. Simultaneamente, a *Internet of Things* (IoT) tem registado um rápido crescimento e tem vindo a ser adotada em múltiplos domínios, mostrando ser um dos grandes pilares para a implementação dos referidos sistemas de monitorização. Ainda assim, as tecnologias de comunicação que prevalecem em pequenos dispositivos (*wearables*) que habilitam esses sistemas, continuam a ser o Bluetooth e o IEEE 802.15.4. Essas tecnologias obrigam a utilização de *gateways* (GWs), como por exemplo *smartphones*, para efetuar a ligação desses dispositivos à Internet. Por outro lado, a utilização da tecnologia IEEE 802.11 (WiFi) e de toda a infraestrutura já existente, habilita a conexão direta à Internet, mostrando grande potencial para redução de atrasos nas comunicações e nos custos na implementação desses sistemas. Apesar disso, o WiFi apresenta geralmente consumos energéticos mais elevados que as restantes tecnologias, o que pode representar uma redução significativa na autonomia desses dispositivos.

Nesta dissertação é analisada a utilização da norma IEEE 802.11 (WiFi) em aplicações de IoT e de comunicação *machine-to-machine* (M2M) com restrições no consumo energético. Em particular, é realizada a caracterização detalhada de um novo módulo, o ESP8266, com suporte à tecnologia WiFi e que possui mecanismos de poupança de energia embebidos, mas cuja performance em cenários de IoT permanece por documentar. São dessa forma explorados os diferentes modos de poupança de energia disponíveis (*Modem-sleep*, *Light-sleep* e *Deep-sleep*) e é quantificado o impacto de diferentes valores dos parâmetros de configuração Beacon *interval* e DTIM *period* no *Access Point* (AP). São também avaliados outros parâmetros de performance relevantes, como o *Packet Delivery Ratio* (PDR) ou o *Received Signal Strenght* (RSSI) em função da distância e da orientação da antena embutida no módulo.

São também desenvolvidos os componentes de *software* necessários à integração do referido módulo WiFi numa *framework*, baseada no *standard* oneM2M, que pretende habilitar o seguimento e monitorização em contínuo de pacientes em ambiente hospitalar. Foi considerado um modelo de comunicação do tipo *publish-subscribe* e utilizado o MQTT como protocolo da camada de aplicação. De forma a avaliar a performance do módulo ESP8266 neste cenário, foram realizadas medições de consumo energético e de avaliação da latência ponto-a-ponto, com transmissões de dados a diferentes frequências e utilizando os diferentes modos de poupança de energia. Os resultados mostram a adequação deste módulo a cenários como este, ao apresentar latências médias na comunicação entre 30-60ms e uma autonomia de 2-4 dias considerando transmissão continua de dados com frequências entre 1s e 10s e a alimentação direta do mesmo com uma bateria de 3.3V/1000mAh de capacidade, até esgotar 90% da sua carga.

Abstract

In Europe, average life expectancy at birth has increased 14% in the last 20 years, this increment and the consequent aging of the world population brings with it a huge number of new challenges. Especially, it entails the need to find innovative and low-cost solutions that facilitate the continuous monitoring of patients inside and outside of the hospital environment, reducing costs to health systems and increasing the quality of life of these patients. At the same time, Internet of Things (IoT) has been growing rapidly and is one of the major pillars for the implementation of such monitoring systems. Nevertheless, the communication technologies that prevail in wearables (e.g. smartband) that enable such systems, remain Bluetooth and IEEE 802.15.4. These technologies impose the use of gateways (GWs), such as smartphones, to connect the wearables to the Internet. On the other hand, the use of IEEE 802.11 technology (WiFi) and all existing infrastructure, enables direct connection of that diapositives to the Internet, showing great potential for reducing communications delays and the costs of implementing these systems. Despite this, WiFi generally has higher power consumption than other technologies, which can represent a significant reduction on the autonomy of these devices.

In this dissertation is analysed the use of the IEEE 802.11 (WiFi) standard for machine-to-machine (M2M) communication in applications with energy consumption constraints. In particular, we make a detailed characterization of the ESP8266 modules. These new modules have a set of characteristics that make them attractive for that type of applications, like the support of WiFi standard, a small size and a set of energy saving mechanisms. Regardless of that, the performance of this module remains to be documented. The different energy-saving modes (Modem-sleep, Light-sleep and Deep-sleep) and the impact of different values of the Beacon Interval and DTIM Period configuration parameters in the Access Point (AP) are thus evaluated. Other relevant performance parameters such as the Packet Delivery Ratio (PDR) or the Received Signal Strength (RSSI) are also analysed as a function of the distance and orientation of the built-in antenna of the module.

Is also developed the required software components to integrate the ESP8266 into a framework, based on standard oneM2M, intended to enable continuous monitoring of patients in a hospital environment. For that a publish-subscribe communication model was considered and MQTT was used as application layer protocol. To evaluate the performance of the ESP8266 module in this scenario, energy consumption and point-to-point latency measurements were performed, with data transmissions at different frequencies and using the different energy saving modes. The results show the suitableness of this module to scenarios like this, with average end-to-end latencies between 30-60 ms and autonomy of 2-4 days, considering continuous transmission of data every 1s or 10s and the power of the module directly by a battery of 3.3V/1000mAh of capacity, until reach 90% of its capacity.

Agradecimentos

Em primeiro lugar, gostaria de agradecer ao meu orientador Professor Luís Almeida, pela constante rapidez de resposta, pelo seu apoio sempre presente e por todas as oportunidades de desenvolvimento pessoal que me proporcionou ao longo do período de realização desta dissertação.

A todos os colegas do laboratório DarTES, e em particular à Diana Guimarães e ao Carlos Pereira, pelos conselhos, trocas de ideias e todo o apoio prestado na realização desta dissertação.

A toda a minha família, e em particular aos meus pais, por todo o apoio, ajuda e acima de tudo por me terem proporcionado desde sempre todas as condições para poder singrar na vida e me tornar quem sou hoje.

À minha namorada, Daniela, por ser um exemplo de dedicação e garra, e por toda a indispensável força e motivação que sempre me deu.

Por último, mas não menos importante, gostaria de agradecer a todos os meus amigos pelos bons momentos que me proporcionaram.

A todos o meu muito obrigado,

João Mesquita

"Everybody is a genius. But if you judge a fish by its ability to climb a tree, it will live its whole life believing that it is stupid."

Albert Einstein

Conteúdo

Lista de Figuras

Lista de Tabelas

Abreviaturas e Símbolos

ADN	Application Dedicated Nodes
AE	Application Entity
AP	Access Point
BSS	Basic Service Set
CSE	Common Service Entity
CoAP	Constrained Application Protocol
DTIM	Delivery traffic indication message
EHR	Electronic health record
ESS	Extended service set
GW	Gateway
HR	Heart rate
HTTP	Hypertext Transfer Protocol
IN	Infrastructure Node
IoT	Internet of Things
MAC	Medium access control
MQTT	Message Queuing Telemetry Transport
PDR	Packet Delivery Ratio
QoS	Quality of Service
REST	Representational State Transfer
RSSI	Received Signal Strength Indicator
RTD	Round-trip delay
SDK	Software development kit
STP	Spanning tree protocol
TCP	Transmission Control Protocol
TIM	Traffic indication map
TLS	Transport Layer Security
UDP	User Datagram Protocol
URI	Uniform Resource Identifier
WBAN	Wireless body area network
WSN	Wireless sensor network

Capítulo 1

Introdução

1.1 Motivação e contexto

O aumento da esperança média de vida e o consequente envelhecimento da população mundial traz consigo um conjunto de novos desafios. Na Europa, a esperança média de vida à nascença cresceu cerca 14%, nos últimos 20 anos, ao aumentar dos 69 anos em 1980 para os 82 anos em 2015 [12]. A comparação da pirâmide demográfica em 2016 com a respetiva projeção para 2080, figura 1.1, indica uma continuação da tendência de envelhecimento da população, sendo expectável que o grupo de pessoas com idade superior a 85 anos represente o maior grupo populacional de toda a pirâmide.

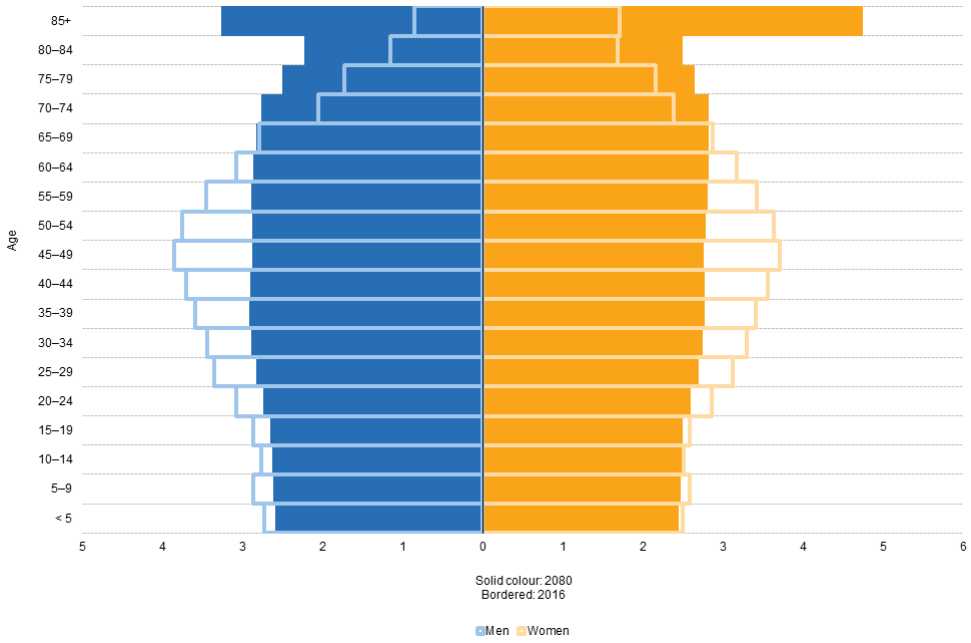


Figura 1.1: Comparação das pirâmides demográficas na Europa (2016-2080) [1].

Por outro lado, o número de doenças crónicas como doenças do foro cardiovascular, diabetes ou obesidade tende a aumentar todos os anos. Segundo a Organização Mundial de Saúde, a nível

mundial, cerca de 17.5 milhões de pessoas morrem de ataque cardíaco todos os anos e cerca de 246 milhões de pessoas sofrem de diabetes. Valores que tendem a aumentar para 380 milhões em 2025 [4]. Estes dados antecipam uma sobrecarga dos sistemas de saúde dos vários países e o consequente aumento de custos, ilustrando por isso a necessidade de encontrar soluções inovadoras e de baixo custo que facilitem a monitorização contínua de doentes dentro e fora do meio hospitalar.

Os mais recentes avanços nos sistemas de comunicação, sistemas eletrónicos e redes *wireless* permitem que estas soluções sejam uma realidade num futuro próximo a um custo relativamente reduzido. Particularmente, o aparecimento de um novo paradigma tecnológico intitulado de *Internet of Things* (IoT), tende a banalizar a conexão de qualquer objeto à Internet. Com isto as *Wireless Body Area Networks* (WBANs) são tidas como a solução mais óbvia e eficiente para obedecer aos requisitos das referidas aplicações de telemedicina. Apesar disso, a maior parte das soluções de monitorização no mercado que utilizam WBANs tem a condicionante de depender de tecnologias de comunicação *wireless* (Bluetooth ou ZigBee) que necessitam de uma *gateway* para estabelecer uma ligação à Internet e dessa forma proceder ao envio da informação recolhida para um centro médico. A par disso, a tecnologia Wi-Fi (IEEE 802.11) tem visto uma adesão crescente com a expansão para domínios onde não era tipicamente usada. Torna-se, por isso, cada vez mais interessante o estudo da convergência e aplicabilidade, em aplicações de WBANs, do *standard* WiFi integrando assim este tipo de redes no paradigma da IoT.

1.2 Objetivos do projeto

As aplicações médicas de WBANs apresentam como principais requisitos um baixo consumo energético e uma baixa taxa de manutenção associado a um tamanho do produto final que se pretende que seja o mais reduzido e leve possível.

Pretende-se com esta dissertação o estudo aprofundado da aplicabilidade a este tipo de redes de pequenos sistemas baseados em microcontroladores de 32 bits, com tecnologia WiFi (IEEE 802.11). Em particular, pretende-se realizar o estudo e caracterização de um recém introduzido dispositivo com essas propriedades: o módulo WiFi ESP8266. Este dispositivo apresenta um custo muito reduzido e é anunciado como tendo mecanismos que permitem vários níveis e modos de poupança de energia, mas cuja verdadeira performance em cenários de IoT é ainda desconhecida.

É importante por isso, o estudo da norma IEEE 802.11 dando particular atenção aos mecanismos já previstos na mesma que permitem a implementação dos referidos ciclos de muito baixa atividade no módulo WiFi, quantificando o impacto que esse tipo de abordagem tem no consumo energético.

Serão também estudados aspetos arquiteturais que permitem a ligação à Internet deste tipo de dispositivos, possibilitando a sua integração em aplicações de monitorização remota de pacientes.

1.3 Estrutura do documento

Os capítulos do presente documento foram organizados segundo a estrutura descrita nos pontos seguintes:

- No **capítulo 1** é apresentada a contextualização e a motivação do projeto, bem como são definidos os objetivos do mesmo.
- No **capítulo 2** é realizada uma revisão bibliográfica dos requisitos e de algumas das aplicações de WSNs e WBANs e são apresentados protocolos e tecnologias de comunicação *wireless* relevantes.
- No **capítulo 3** é realizada uma introdução à norma IEEE 802.11, explicando com detalhe o tipo de tramas e mecanismos que permitem a operação de estações WiFi em modos de poupança de energia.
- No **capítulo 4** são discutidas as principais características do módulo WiFi ESP8266, dando particular destaque aos diferentes modos de poupança de energia disponíveis.
- No **capítulo 5** é descrita a metodologia adotada nas experiências de caracterização do módulo ESP8266 e são discutidos os resultados obtidos nas diferentes experiências.
- No **capítulo 6** é apresentada a arquitetura proposta para integração do módulo ESP8266 numa *framework* para monitorização de pacientes em ambiente hospitalar.
- No **capítulo 7** são discutidos os resultados de performance relativos ao consumo de corrente e latência na comunicação, com transmissões periódicas de informação pelo módulo ESP8266 integrado na arquitetura proposta no capítulo anterior.
- No **capítulo 8** são discutidas as conclusões finais do projeto e são apresentadas tópicos para o desenvolvimento do trabalho no futuro.

Capítulo 2

Revisão Bibliográfica

2.1 *Wireless Sensor Networks* (WSNs)

2.1.1 Visão geral

Têm-se assistido na última década a uma redução acentuada do tamanho e custo associado aos mais variados componentes eletrónicos. Esta redução de tamanho, associada ao aumento do poder computacional dos mesmos, tem feito emergir o aparecimento de novas classes de computadores e de dispositivos inteligentes. Em particular, os avanços na comunicação sem fios, no design de sensores e na eficiência e armazenamento de energia elétrica, fizeram elevar o conceito de *Wireless Sensor Network* (WSN) [13]. Esta classe de redes, apresenta normalmente pequenos sensores integrados, com um custo reduzido e uma capacidade limitada de processamento, mas capazes de medir características mecânicas, térmicas, biológicas, químicas ou magnéticas do meio envolvente e de as transmitir ao utilizador. A tecnologia atual presente nestes sensores permite a sua aplicação em WSN num grande número de contextos, alguns deles descritos com detalhe na secção 2.1.3.

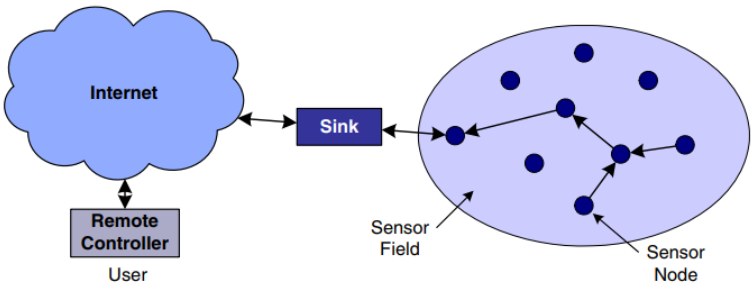


Figura 2.1: Arquitetura típica de uma WSN [2].

É comum enquadrar as características dos nodos presentes numa WSN em duas categorias: nós genéricos e nós *gateway* [3]. Uma combinação destas duas categorias é normalmente usada para formar uma WSN, tal como representado na figura 2.1:

- **Nós genéricos**: A principal função dos sensores presentes neste tipo de nó é fazer medições do meio a ser monitorizado. Por exemplo, é comum realizarem medições dos atributos

físicos desse mesmo meio: temperatura, humidade, pressão atmosférica, velocidade, aceleração, etc.

- **Nós *gateway***: Estes nós devem agregar a informação adquirida pelos nós genéricos e transmiti-la para uma estação base. Devem, por isso, possuir maior capacidade de processamento e bateria, aliados a um maior alcance de transmissão de dados *wireless*.

Pode-se ainda dividir em três grupos as tarefas necessárias à utilização destes sensores em aplicações de redes WSN, tal como representado na figura 2.2 [3]:

- **Sistema**: Cada um dos nós pode ser considerado um sistema individual que possuí características especificas que dependem da plataforma utilizada, do sistema operativo, do microcontrolador, da capacidade de memória, entre outros fatores.
- **Protocolos de comunicação**: Permitem a conexão e transmissão de informação entre os diferentes nós. Estes protocolos podem ser divididos em 5 camadas distintas *standard* para a troca de pacotes: aplicação, transporte, rede, ligação de dados e física. A implementação dos protocolos de rede em diferentes camadas pode afetar significativamente o consumo de energia e a dinâmica da WSN.
- **Serviços**: Este grupo deve ser desenvolvido de forma a enriquecer a aplicação e a melhorar a performance do sistema e a eficiência da rede. Funções como localização, sincronização, segurança, agregação e compressão de informação, são normalmente características desta categoria.

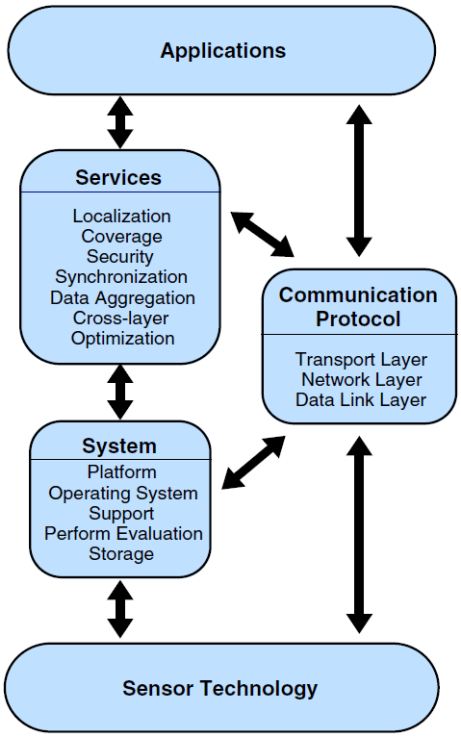


Figura 2.2: Classificação em grupos de aspectos relevantes em redes WSN [3].

2.1.2 Consumo energético

Antes de discutir algumas das técnicas possíveis para implementar mecanismos que conduzam a um reduzido consumo de energia neste tipo de redes, é importante referir quais os subsistemas presentes em cada um dos nós e desses quais os que representam uma maior percentagem no consumo de energia total. Resultados experimentais indicam que a transmissão de informação apresenta um consumo energético bastante mais elevado que o processamento [14]. Já o consumo do subsistema de *sensorização* depende do sensor em causa, mas em algumas situações pode até ser negligenciado quando comparado com o gasto verificado pelos restantes subsistemas. Noutros casos, a energia necessária neste subsistema é comparável, ou ainda maior, à necessária para a transmissão de informação [2]. Assim, é comum as técnicas de poupança de energia nestes sistemas focarem-se em dois subsistemas: comunicação e *sensorização*.

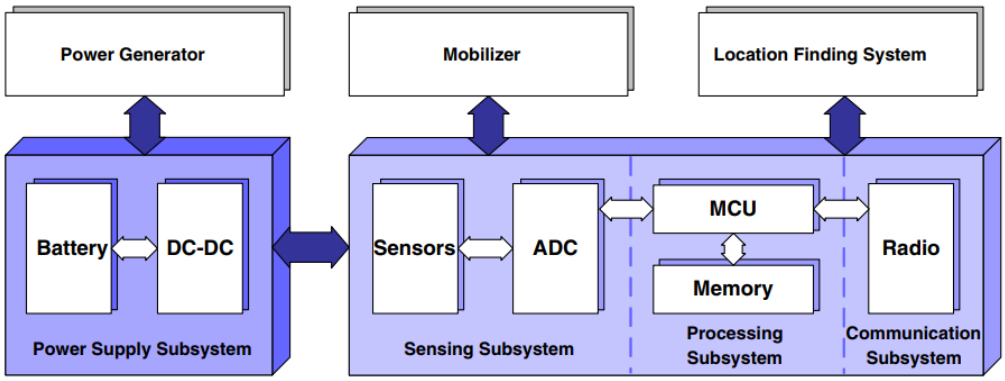


Figura 2.3: Arquitetura típica de um nó numa rede WSN [2].

Tal como representado na figura 2.3, a arquitetura de um nó pode ser dividida em quatro componentes principais [14]:

1. Fonte de energia: Constituído por uma bateria e um conversor DC-DC, fornece energia aos restantes subsistemas do nó. Este modulo poderá ainda possuir um elemento de captação de energia, como uma célula solar.

2. Subsistema de *sensorização*: Consiste num grupo de sensores, em alguns casos complementando com atuadores, que permitem estabelecer a ligação entre o meio físico e o restante sistema.

3. Subsistema de processamento: Inclui um microprocessador ou um microcontrolador e uma memória, responsáveis pelo processamento de informação ao nível local e pela interligação entre o subsistema de *sensorização* e o de comunicação.

4. Subsistema de comunicação: Consiste num *transceiver* rádio que permite efetuar uma comunicação *wireless* bidirecional de baixo alcance.

Embora o consumo de energia esteja altamente relacionado com as especificações de cada nó, pode-se retirar algumas ilações que se aplicam à maior parte dos sistemas. Como já referido, o

subsistema de comunicação apresenta um consumo que ultrapassa geralmente o registado pelo subsistema de processamento. O custo de transmitir um único bit é aproximadamente o mesmo de processar um conjunto de milhares de instruções [2]. É possível também verificar que o consumo de energia é da mesma ordem de magnitude na transmissão, receção e nos estados em que as comunicações se encontram ativas mas não existe comunicação efetiva, descendo consideravelmente num estado de *sleep*. A ativação de modos de *sleep* deve por isso ser maximizada. Outro aspeto a ter em conta e que depende da especificação de cada aplicação, é o consumo do subsistema de *sensorização* que pode em algumas situações ser significativo [2].

2.1.2.1 Mecanismos de poupança de energia

De uma forma geral, pode-se identificar duas estratégias principais para aplicação de mecanismos de poupança de energia: *duty cycling* e abordagens *data-driven* [2].

O *duty cycling* foca-se essencialmente no subsistema de comunicação. Conforme anteriormente discutido, a forma mais efetiva de poupança de energia é manter o *transceiver* rádio num estado de *sleep* sempre que não é necessário efetuar uma comunicação. Idealmente, o sistema deveria apenas sair desse estado de poupança energética quando fosse necessário transmitir ou receber alguma informação, regressando ao mesmo imediatamente após esse evento terminar. Desta forma, os nós devem alterar entre períodos de atividade e de *sleep* dependendo da atividade da rede, sendo o *duty cycle* definido como a percentagem de tempo que os mesmos permanecem ativos. Como cada nó realiza habitualmente tarefas cooperativas dentro da WSN, devem existir mecanismos de sincronização que permitam uma coordenação da transmissão e receção de informação e conduzam a uma comunicação efetiva entre todos os nós da rede, mesmo mantendo um *duty cycle* muito reduzido em todos os nós.

Aliar as estratégias de *duty cycling* a abordagens de controlo da informação adquirida torna possível reduzir ainda mais o consumo energético nestes sistemas. Deve-se por isso considerar a eliminação de amostras desnecessárias ou redundantes. No caso em que o consumo energético do subsistema de *sensorização* é negligenciável, cada amostra em excesso irá provocar comunicações desnecessárias. Os mecanismos de controlo de informação devem por isso restringir o número de amostras ao estritamente necessário para cada aplicação.

2.1.3 Aplicações

O surgimento do paradigma das WSN despertou o interesse na investigação dos diversos aspetos relacionados com as mesmas, tendo surgido inúmeras possibilidades de aplicação de carácter académico e comercial. Uma WSN pode consistir num conjunto de diferentes tipos de sensores em que se incluem os seguintes: magnéticos, térmicos, visuais, infravermelhos, acústicos ou radar. Estes sensores permitem medir um grande número de parâmetros do meio envolvente, entre os quais: temperatura, humidade, pressão atmosférica, velocidade, movimento, intensidade luminosa, níveis de ruído, presença ou ausência de determinados objetos, entre outros [15]. Como resultado, tem-se um potencial crescimento para a utilização das WSNs em inúmeros cenários. O

espectro de utilização inclui aplicações para o seguimento de pessoas, objetos ou animais, para criação de sistemas de segurança em habitações, análise e previsão das condições meteorológicas, monitorização da condição de saúde de um paciente, análise dos consumos energéticos de uma instalação elétrica ou a automação de um processo industrial. Tal como representado na figura 2.4 pode-se classificar essas aplicações em, essencialmente, duas categorias: seguimento e monitorização [3].

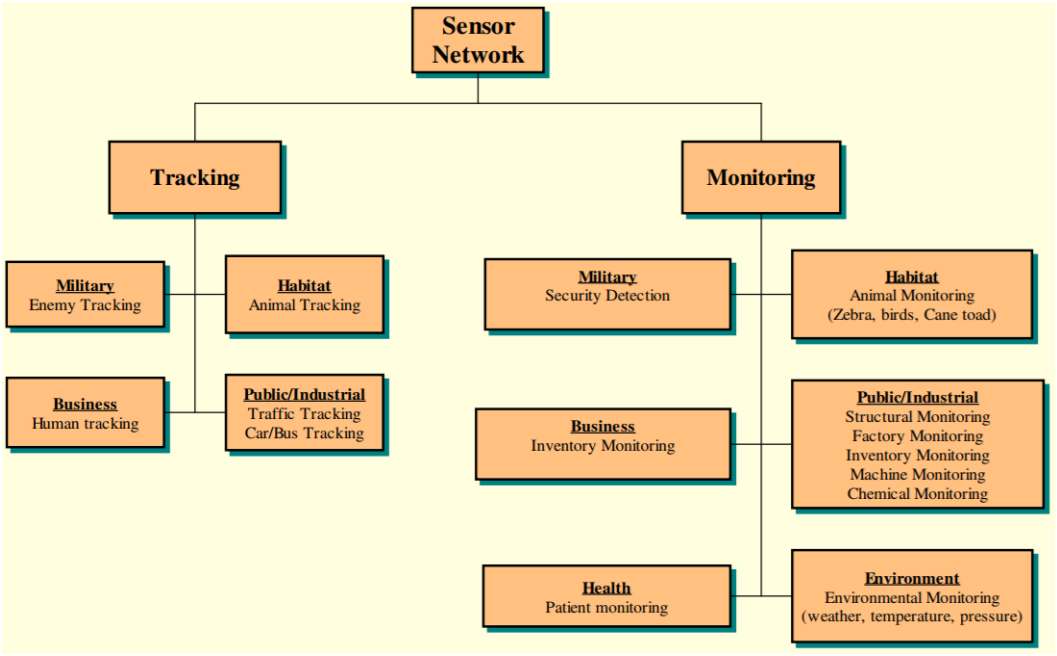


Figura 2.4: Exemplos de aplicação de WSNs [3].

2.1.3.1 Aplicações militares

A relativa facilidade de implementação e as características de tolerância a falhas inerentes a este tipo de redes, faz com que possam ser integradas em tarefas militares de comando, controlo, comunicação, computação, inteligência, vigilância ou reconhecimento com elevado sucesso. Ao serem baseadas numa elevada densidade de pequenos sensores de baixo custo, a destruição de alguns nós da rede, num ataque de um adversário, tende a não comprometer a missão militar [15]. São exemplos de projetos nesta área: Boomerang III [16], para deteção e localização de um atirador; e VigilNet [17], para vigilância e seguimento remoto.

2.1.3.2 Aplicações ambientais

A capacidade de coordenação autónoma das WSNs permite que sejam utilizadas num vasto número de aplicações deste carácter. Algumas destas incluem seguimento de movimento de pequenos animais e insetos, monitorização das condições atmosféricas, sistemas de agricultura e rega automática ou a deteção de fogos florestais [15]. São exemplos de projetos nesta área: Milsar

[18], para monitorização de animais à distância; e AVTEC [19], para monitorização e gestão das condições ambientais no interior de edifícios.

2.1.3.3 Aplicações em habitações

Com os avanços nesta tecnologia, sensores inteligentes e atuadores podem construir aplicações para controlo e monitorização dos mais variados sistemas domésticos, onde se podem incluir: controlo da iluminação, sistemas de vídeovigilância, fornos, frigoríficos ou gestão dos sistemas de energia elétrica. A conexão à Internet destes sensores e atuadores permite que um utilizador faça a gestão dos seus equipamentos domésticos à distância com elevada facilidade. São exemplos de projetos nesta área: InovGrid [20], para automatização da gestão do sistema de energia elétrica; e MEO SmartHome [21], para efetuar controlo e monitorização dos mais variados dispositivos domésticos.

2.2 *Wireless Body Area Networks* (WBANs)

O envelhecimento da população mundial e o consequente aumento dos custos inerentes aos sistemas de saúde dos vários países, tem conduzido a um grande esforço, por parte de toda a comunidade cientifica, na introdução e desenvolvimento de novos sensores do tipo *wearable* (que se podem inserir no corpo ou no vestuário) que permitam a monitorização de parâmetros fisiológicos de um utilizador. Esses desenvolvimentos levaram à introdução do conceito de *Wireless Body Area Network* (WBAN).

2.2.1 Requisitos

Uma *Wireless Body Area Network* (WBAN), tal como representado na figura 2.5, consiste num conjunto de dispositivos inteligentes e pequenos, implantados no ou juntos ao corpo humano, que possuem a capacidade de estabelecer uma comunicação *wireless* entre si e uma entidade externa. Os dispositivos presentes numa WBAN podem ser divididos em duas categorias:

- **Sensor**: Dispositivo que recolhe, processa e envia informação de um determinado parâmetro fisiológico do utilizador (batimento cardíaco, temperatura do corpo, nível de oxigénio no sangue, entre outros). São constituídos pelo hardware do sensor, um microcontrolador, memoria e um *transceiver* radio que habilita as comunicações *wireless*.

- **Atuador**: Dispositivo que atua de acordo com a informação recolhida pelos sensores ou consoante uma ordem fornecida pelo utilizador. Por exemplo, este pode administrar a dose correta de insulina a um paciente com diabetes, tendo em conta os níveis de glicose medidos pelo respetivo sensor. São constituídos por componentes semelhantes aos sensores, podendo ainda conter um reservatório onde é armazenada a medicação a administrar.

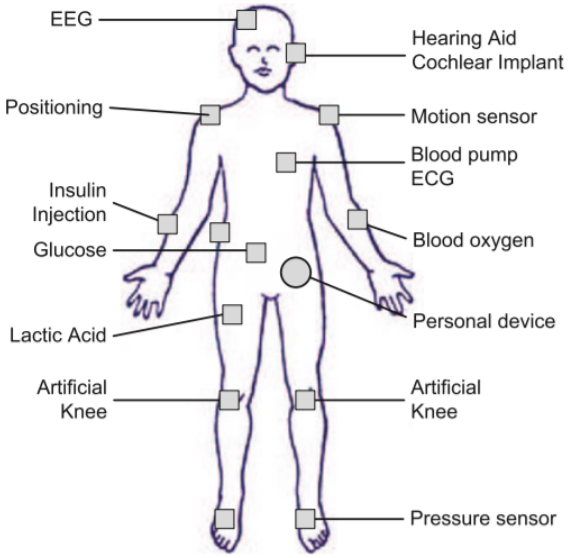


Figura 2.5: Exemplo do conjunto de sensores e atuadores presentes numa WBAN [4]

Aplicação	**Data rate**	**Exatidão (bits)**	**Tipo de sensor**	**Descrição da informação medida**
Eletrocardiograma (ECG) - 12 pontos	288 kbps	12	Elétrodos na pele	Análise da atividade elétrica do coração.
Eletromiografia (EMG)	320 kbps	16	Elétrodos na pele	Análise da atividade elétrica das fibras musculares.
Eletroencefalografia (EEG)	43.2 kbps	12	Elétrodos no couro cabeludo	Análise da atividade elétrica do cérebro.
Níveis de oxigénio no sangue	16 bps	8	Oximetro de pulso	Registo da saturação periférica de oxigénio (SpO2)
Monitorização da glicose	1600 bps	16	Sensor de glicose inserido na pele	Registo dos níveis de glicose no sangue.
Temperatura	120 bps	8	Adesivo corporal	Registo da temperatura corporal.
Sensor de movimento	32 kbps	12	Acelerómetro	Registo dos movimentos corporais.
Áudio	1 Mbps	-	Microfone	Registo de som.
Voz	50-100 kbps	-	Microfone	Registo de som.

Tabela 2.1: Taxa de transmissão de dados e exatidão necessária para um conjunto de aplicações em WBANs. Adaptado de [4].

A maior parte das implementações necessita ainda de um dispositivo pessoal, como por exemplo um *smartphone*, para servir como *gateway* para a Internet e desta forma efetuar uma sincronização da informação recolhida com um servidor externo [4]. Pretende-se assim tirar partido destes

sensores e atuadores, para a criação de sistemas que efetuem recolha de dados de sinais fisiológicos de um utilizador e procedam ao envio dessa informação para um servidor externo, onde a mesma pode ser consultada e analisada por uma equipa médica ou pelo próprio utilizador. Devido à elevada heterogeneidade de sensores passíveis de integrar uma WBAN, a taxa de transmissão dos dados e a exatidão necessária para cada aplicação, pode variar desde as dezenas de bps às centenas de de kbps [4]. Encontra-se sintetizado na tabela 2.1 essa informação para um conjunto de aplicações e sensores típicos numa WBAN.

2.2.2 Aplicações

As aplicações de WBANs abrangem um largo espectro de atividades. É possível encontrar utilidade na aplicação destes serviços em áreas do foro militar, aplicações de monitorização em situações médicas, apoio a atletas de alto rendimento, entretenimento, entre outras. De forma geral, o principal objetivo destas aplicações é possibilitar uma melhoria da qualidade de vida do utilizador nos mais variados cenários. Pode-se classificar as mesmas em duas grandes áreas: aplicações médicas e aplicações não médicas.

2.2.2.1 Aplicações médicas

As aplicações de foro médico concentram-se na medição continua de um ou mais parâmetros fisiológicos de um utilizador (pressão arterial, temperatura corporal, níveis de oxigénio ou glicose no sangue, entre outros). Desta forma, o principal objetivo destas aplicações é providenciar uma monitorização em continuo do utilizador sem interferir com as suas atividades do dia-a-dia [22]. Na figura 2.6 encontra-se representado um cenário da utilização de WBANs nestes contextos.

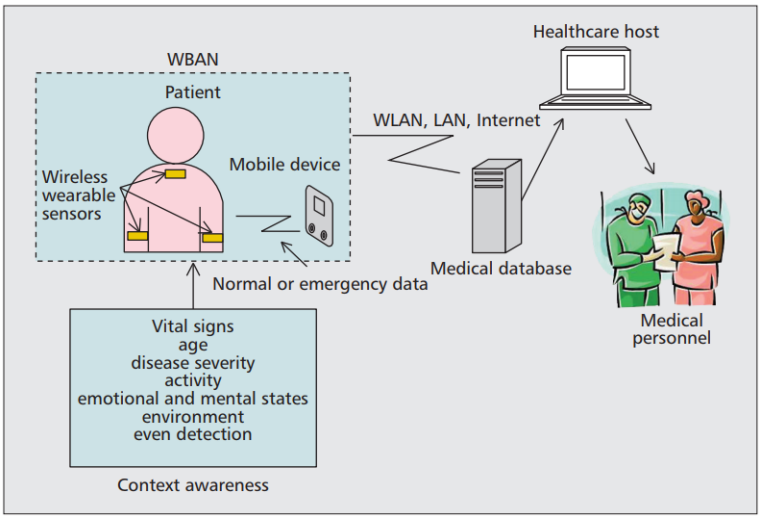


Figura 2.6: Cenário de utilização de WBANs em contextos médicos [5]

Pode-se subdividir nos seguintes 3 grupos as aplicações médicas de WBANs:

- **Monitorização remota de pacientes**: Este tipo de aplicações consiste na monitorização dos sinais vitais de um paciente, fora do ambiente hospitalar, providenciando desta forma informação sobre o estado de saúde do mesmo ao pessoal médico que o segue. Incluem, normalmente, sensores que permitem realizar eletrocardiogramas (ECG) e medir a pressão sanguínea para controlar a condição cardíaca, realizar eletroencefalogramas (EEG) de forma a monitorizar as habilidades cognitivas ou controlar a respiração e medir a temperatura corporal de forma a detetar infeções. Na tentativa racionalizar o consumo energético, algumas aplicações utilizam este tipo de monitorização num contexto baseado em eventos, enviando a informação recolhida apenas no caso de alguma condição anormal ser detetada. Estas aplicações pretendem assim conduzir a uma redução das necessidades de hospitalização, em caso de doenças menos graves, retirando uma carga significativa aos hospitais e levando a uma redução de custos com estes doentes. É possível, ainda introduzir melhorias na qualidade de vida do paciente como a possibilidade de conciliar uma vida independente com uma monitorização, em continuo, do seu estado de saúde fora do ambiente hospitalar. São exemplos de projetos nesta área: BIOTRONIK *Home Monitoring* [23] e LATITUDE NXT *Patient Management System* [24], como sistemas que suportam a monitorização remota de parâmetros fisiológicos de um paciente na sua habitação.

- **Reabilitação de pacientes**: Este tipo de aplicações pretende ajudar na reabilitação de pacientes com problemas de mobilidade, ao permitir que os exercícios e atividades realizadas nesses contexto sejam monitorizados. Desta forma, pretende-se analisar a execução correta dos movimentos assim como fornecer informações que permitam reajustar as necessidades de cada paciente ao mesmo tempo que se monitoriza a recuperação do paciente. O objetivo da WBAN neste contexto é a utilização de acelerómetros, giroscópios e magnetómetros para capturar os movimentos e a postura do utilizador.

- **Vida assistida**: Estas aplicações pretendem ser uma alternativa à colocação em lares de pessoas desabilitadas ou idosas que já não conseguem ser completamente independentes, embora não necessitem de um acompanhamento médico permanente. As WBANs conseguem acompanhar os movimentos e estado de saúde dessas pessoas informando um responsável no caso de uma queda ou problema crítico de saúde. Em [25] é apresentado um sistema destinado a pessoas idosas que permite o acompanhamento remoto dos movimentos e posição do corpo do utilizador.

2.2.2.2 Aplicações não médicas

As aplicações neste âmbito têm como público alvo utilizadores habituais de tecnologia e que têm interesse em usar informação sobre os diversos parâmetros fisiológicos para fins de entretenimento, segurança ou *fitness*. Na figura 2.7 encontra-se um cenário representativo da utilização de WBANs nestes contextos.

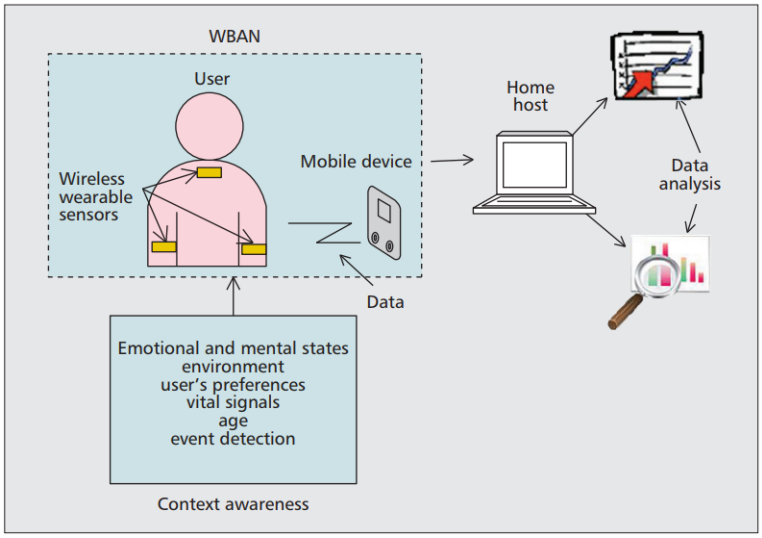


Figura 2.7: Cenário de utilização de WBANs em contextos não médicos [5]

As aplicações não médicas de WBANs são habitualmente utilizadas para monitorizar os movimentos corporais do utilizador, alguns sinais fisiológicos (pulsação cardíaca e nível de oxigénio no sangue), luz e temperatura ambiente. Possibilitam assim ao utilizador um acompanhamento da atividade física diária ou a análise dos seus ciclos de sono. A partir da análise da informação recolhida o utilizador pode alterar hábitos e consequentemente melhorar a sua qualidade de vida. Em [26] é apresentado o projeto *Smart Vest*, um sistema de monitorização de parâmetros fisiológicos utilizando sensores associados à roupa do utilizador.

2.3 *Internet of Things* (IoT)

Como discutido nas secções 2.1 e 2.2 a proliferação das WSNs e das WBANs permite a criação de redes de objetos inteligentes com capacidades de sensorização, processamento e comunicação. Neste contexto a *Internet of Things* (IoT) permite a conexão desses objetos à Internet, promovendo a interligação entre os mesmos e as diferentes aplicações. Assim, enquanto a Internet *comum* é formada por um conjunto de dispositivos, com capacidades distintas mas com propósitos e propriedades muito semelhantes entre si, é expectável que a IoT exiba um muito maior nível de heterogeneidade, ao integrar a conexão de um grande número de objetos com funcionalidades distintas [27]. Desta forma, a IoT introduz um conjunto de novos desafios, com vista a responder às restrições de capacidade de processamento, memória e consumo energético, características dos dispositivos integrados nesse tipo de redes. Esses desafios incluem a exploração de novos modelos e protocolos de comunicação que suportam as aplicações baseadas em IoT, bem como o desenvolvimento de novas plataformas de hardware que respondam aos desafios impostos por este paradigma. Esta secção explora alguns dos protocolos e plataformas considerados capazes de responder a esses desafios.

2.3.1 Protocolos de comunicação

Os protocolos de comunicação definem a forma como a informação é trocada entre qualquer dispositivo conectado a uma rede, permitindo que dispositivos com diferentes recursos ou com *software* desenvolvido em linguagens de programação distintas consigam trocar dados entre si. Assim, estes protocolos podem estar presentes desde o momento de recolha de dados por um sensor, ao envio e entrega desses a uma determinada aplicação. Em particular, uma vez que em IoT os dispositivos conectados tem recursos limitados, a escolha do protocolo a utilizar pode ter um grande impacto na eficiência das comunicações e correspondente autonomia dos mesmos. No decorrer desta secção serão revistos os aspetos mais relevantes de três protocolos adequados [6] a esse tipo de redes: MQTT, HTTP e CoAP.

2.3.1.1 MQTT

O *Message Queuing Telemetry Transport* (MQTT) [28] é um protocolo de transmissão leve, assente na camada de transporte TCP e especialmente desenvolvido para operar em dispositivos com recursos limitados e redes de comunicação com fiabilidade reduzida. É baseado no modelo *publisher-subscriber*, estando por isso cada uma das mensagens transmitidas associadas a um tópico especifico. Tal como apresenta a figura 2.8, essas mensagens são enviadas para um *broker* que atua como mediador central das comunicações, procedendo ao encaminhamento das mesmas para as entidades que subscreveram o tópico associado a cada mensagem.

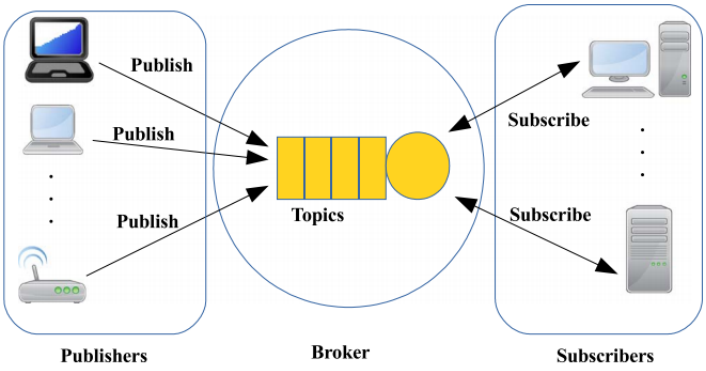


Figura 2.8: Arquitetura do protocolo MQTT [6]

Cada uma das mensagens pode ser enviada com diferentes níveis de *Quality of Service* (QoS). O parâmetro de QoS, define o nível de garantia que determinada mensagem tem de ser entregue ao seu destinatário. Assim, quando determinado cliente subscreve ou publica num tópico, acorda com o *broker* MQTT o nível de QoS que deseja para as mensagens que subscreveu ou vai publicar no mesmo. O nível de QoS é assim utilizado para controlar a forma como as mensagens são transmitidas entre o *broker* e determinado cliente, estando definidos três níveis distintos [11]:

- **QoS 0**: Este nível de QoS oferece as garantias de entrega inerentes ao protocolo TCP. As mensagens são transmitidas do cliente para o servidor, ou vice-versa, sem a existência de

nenhuma confirmação na direção oposta. O servidor e o cliente podem desta forma eliminar estas mensagens após logo após o seu envio.

- **QoS 1**: Este nível de QoS oferece garantias que a mensagem vai ser entregue pelo menos uma vez ao seu destinatário. Quando um servidor ou cliente recebe mensagens deste tipo, deve enviar uma confirmação de volta ao emissor da respetiva mensagem. Assim, o remetente original da mensagem deve manter uma copia da mesma até receber a referida confirmação, devendo reenviar a mensagem no caso de não receber nenhuma confirmação após um *timeout* definido.

- **QoS 2**: Este nível de QoS oferece garantias que a mensagem vai ser entregue uma única vez ao seu destinatário. As mensagens são enviadas com um mecanismos de confirmação de duas etapas, onde cada um dos passos pode ser repetido, no caso de perda de conexão, sem ocorrer a duplicação da mensagem original. O cliente e o servidor devem manter uma copia da mensagem durante todo o processo.

Fica claro que o incremento do nível de QoS associado a uma mensagem está associado a uma maior fiabilidade na comunicação. Apesar disso, a escolha desse parâmetro deve ter em conta aspetos como a maior latência ou largura de banda necessária associada aos níveis de QoS superiores. Adicionalmente, a disponibilização dos níveis de QoS1 e QoS2 pelo *broker*, obriga a que o mesmo consiga destingir cada um dos clientes individualmente. Esse procedimento pode ser realizado através da atribuição de um identificador único (Client ID) para cada cliente, permitindo que mensagens pendentes sejam entregues mesmo após a perda de conexão temporária com um cliente.

2.3.1.2 HTTP

O *Hypertext Transfer Protocol* (HTTP) é nos dias de hoje um dos protocolos mais utilizados na Internet. As paginas *web* são normalmente transferidas utilizando o HTTP, ou a sua versão segura o HTTPS, que adiciona *Transport Layer Security* (TLS) para encapsular os pacotes HTTP.

Um dos princípios da arquitetura deste protocolo é o *Representational State Transfer* (REST) [29], uma das abordagens dominantes em comunicações do tipo cliente-servidor. Sendo um protocolo RESTful, o HTTP utiliza *Uniform Resource Identifiers* (URI) para identificar os recursos *web* no servidor, no qual os clientes podem executar um conjunto de operações (criação, leitura, atualização e eliminação). Essas operações podem ser executadas por meio da troca de pedidos efetuados pelos clientes e obtenção das respetivas respostas pelo servidor. Cada pedido ao servidor HTTP pode ser de quatro tipos distintos:

- **GET**: Permite obter o conteúdo de um recurso ou conjuntos de recursos já existente.

- **POST**: Permite a criação de um novo recurso.

- **PUT**: Permite a atualização do conteúdo de um recurso já existente.

- **DELETE**: Permite remover um recurso existente.

Desta forma, o cliente possui uma abstração de como os seus pedidos são processados no servidor, mas ao mesmo tempo possui uma forma bem definida de os realizar. Adicionalmente, permite manter uma abordagem *stateless*, uma vez que o servidor não necessita de manter o estado de cada cliente após a resposta a um pedido [29]. Na camada de transporte é comum o uso do protocolo TCP, pela fiabilidade que consegue fornecer. Desta forma, o protocolo HTTP pode ser utilizado num grande número de aplicações, incluído as comunicações em IoT, apesar do elevado *overhead* provocado pelo grande cabeçalho inerente ao protocolo.

2.3.1.3 CoAP

O *Constrained Application Protocol* (CoAP) [30] é um protocolo leve desenhado especialmente para dispositivos com recursos limitados e redes de comunicação constrangidas. Tal como o HTTP, o CoAP é um protocolo RESTful, mas que utiliza algumas adaptações que tornam mais adequadas a sua utilização nesse tipo de dispositivos e redes, como a redução significativa do tamanho dos cabeçalhos. As mensagens CoAP podem ser de quatro tipos distintos: confirmáveis, não confirmáveis, *reset* ou de confirmação. Uma vez que na camada de transporte é utilizado o protocolo UDP, a fiabilidade e garantia de entrega em CoAP é conseguida pelo intermédio das mensagens confirmáveis.

Uma das funcionalidades mais importantes disponibilizadas pelo protocolo CoAP é a obtenção de informação de um recurso de forma assíncrona [6]. Com o envio de um pedido GET com a opção de OBSERVE, o cliente indica ao servidor que pretende obter atualizações de determinado recurso sempre que o mesmo seja alterado, evitando o envio de um novo pedido GET sempre que pretenda obter atualizações desse mesmo recurso. Esta funcionalidade pretende atingir um modelo de comunicação do tipo *publisher-subscriber*, ideal para dispositivos com recursos limitados.

2.3.2 Tecnologias de comunicação *wireless*

Ao longo dos últimos anos, surgiram um grande número de protocolos para comunicação sem fios. Algumas tecnologias permitem a comunicação a longas distâncias e com taxas de envio de dados relativamente altas, enquanto outras são direcionadas a redes de pequeno alcance e com reduzidas taxas de transmissão. Esta secção pretende fazer um *overview* sobre as tecnologias mais relevantes no contexto das redes e sistemas para monitorização corporal.

2.3.2.1 Bluetooth

O *standard* IEEE 802.15.1 (Bluetooth) foi desenhado inicialmente para comunicações de baixo alcance com intuito de substituir as ligações cabladas entre um computador e os diversos periféricos. Os dispositivos Bluetooth operam na banda ISM 2.4 GHz e usam 79 canais na mesma. Este *standard* especifica três classes de dispositivos com diferentes potências e alcances de transmissão e permite que cada dispositivo se conecte simultaneamente a até 7 outros dispositivos dentro de uma única rede chamada *piconet*. Apesar de existirem varias implementações

que utilizam este *standard* em aplicações de telemedicina, o facto de apenas permitir a criação de redes de tamanho reduzido, os elevados tempos de inicialização na ligação de novos dispositivos e o elevado consumo energético durante a operação, torna o mesmo não adequado a aplicações deste tipo. A introdução da variante de baixo consumo deste *standard*: Bluetooth Low Energy (BLE) [31], tornou mais adequada a sua utilização em WBANs que requeiram um reduzido consumo energético. Apesar disto, a sua aplicação, tal como nas restantes versões do *standard*, requer sempre a utilização de um dispositivo como *gateway* (ex: *Smartphone*) para ligação da WBAN à Internet. A falta de suporte desta tecnologia por parte de muitos dos dispositivos atualmente no mercado, torna a sua utilização ainda pouco apetecível.

2.3.2.2 ZigBee

O protocolo ZigBee é um dos vários protocolos desenvolvidos sobre a camada de ligação de dados IEEE 802.15.4 e que é já usado largamente em implementações de WSNs. Ao suportar comunicação *multi-hop* consegue oferecer uma grande área de cobertura e uma melhor performance quando comparado com Bluetooth. Além disso, o consumo de energia é de cerca de um terço do verificado na tecnologia Bluetooth. No entanto, este *standard* apresenta uma reduzida velocidade de comunicação, não consegue suportar uma qualidade de serviço (QoS) adequada a aplicações médicas de WBANs [22] e não permite a ligação direta de uma rede corporal à Internet.

2.3.2.3 WiFi

O protocolo IEEE 802.11, denominado WiFi, estabelece uma serie de *standards* para a criação de redes locais sem fios (WLAN). Baseado neste *standard*, a tecnologia Wi-Fi é capaz de fornecer uma conectividade *wireless* segura, confiável e rápida entre os mais variados dispositivos eletrónicos. Permite ainda a conexão direta desses dispositivos à Internet, a velocidades de banda larga, quando ligados a um ponto de acesso (AP). Este *standard* apresenta ainda a vantagem de já ser largamente utilizado nos mais variados cenários (habitações, hospitais, escolas, ...) e de estar presente na maior parte dos dispositivos eletrónicos atuais (*smartphone*, computador, ...). Apesar das referidas vantagens, este protocolo não foi inicialmente pensado para ser utilizado em dispositivos que operam com elevadas restrições no consumo energético. Por esse motivo, durante muitos anos, esta tecnologia não foi considerada para aplicações de WBANs [32]. Com o aparecimento de pequenos dispositivos de muito baixo consumo que oferecem suporte para a tecnologia WiFi, tornou-se interessante a possibilidade de uso da mesma nesse tipo de redes. A utilização desses dispositivos no contexto das WBANs, oferece a vantagem de permitir a integração destas redes diretamente na larga infraestrutura de APs e dispositivos pessoais já existente, possibilitando ainda uma ligação direta à Internet. A reutilização da infraestrutura WiFi permite ainda uma redução significativa de custos e viabiliza desenvolvimentos mais rápidos [32]. Este protocolo e os mecanismos que habilitam os modos de funcionamento de baixo consumo são descritos com detalhe no capitulo 3.

2.4 Sumário

Neste capítulo foram apresentados os principais requisitos de WSNs e WBANs e um conjunto de aplicações típicas que fazem uso deste tipo de redes para recolher e transmitir informação em diferentes cenários. A transmissão dessa informação é habitualmente realizada utilizando protocolos de comunicação apropriados, como HTTP, CoAP ou MQTT. Neste capítulo foram também discutidos alguns desses protocolos, bem como as tecnologias de comunicação *wireless* habitualmente associadas a esse tipo de redes.

Capítulo 3

IEEE 802.11

Este capítulo apresenta uma introdução à norma IEEE 802.11, focando nos mecanismos de poupança de energia previstos na mesma. Para tal, serão abordados aspetos relacionados com a arquitetura e modos de funcionamento de uma rede deste tipo. Será também dado especial realce a algumas especificações da camada de acesso ao meio (MAC), focando na explicação dos diferentes tipos de tramas e de como as mesmas são utilizadas para habilitar a operação das estações no modo de poupança de energia.

3.1 Arquitetura

A arquitetura básica de uma rede IEEE 802.11 é constituída por constituída por domínios (*clusters*) denominados *Basic Service Set* (BSS). Este tipo de topologia caracteriza-se por um conjunto de estações que pretendem comunicar entre si, dentro de uma área comum, denominada de *Basic Service Area* (BSA). Neste contexto, uma estação (STA) pode ser qualquer dispositivo que implemente as camadas física (PHY) e de acesso ao meio (MAC) e disponha de uma interface com o meio físico de transmissão (WNIC). A coordenação das comunicações e modo de operação dessas estações, depende do tipo de BSS em que estão inseridas. Uma BSS pode ser de dois tipos distintos [33]:

- **Infraestruturada**: Uma rede infraestruturada caracteriza-se pela presença obrigatória de um *Access Point* (AP) que define o alcance e coordena as comunicações entre os dispositivos pertencentes à BSS. Desta forma, o AP funciona como um intermediário entre todas as comunicações, não existindo comunicação direta entre os diversos dispositivos. A BSA é assim definida pelo alcance máximo desse mesmo AP.

  Adicionalmente, a utilização de um AP possibilita o armazenamento temporário de tramas de dados destinadas a uma estação que não se encontra ativa. Estes mecanismos, apresentados com mais detalhe na secção 3.3, permitem a implementação de técnicas de poupança de energia, particularmente importantes em dispositivos móveis alimentados com baterias de pequena dimensão.

- **Independente** (*Ad hoc*): Numa BSS do tipo independente (IBSS), as estações constituintes comunicam diretamente entre si. Dispensa assim uma entidade central que medeie e defina o alcance das comunicações (AP). Por esse motivo, o alcance máximo das comunicações diretas é normalmente mais reduzido.

Quando uma única BSS não possui cobertura suficiente numa determinada área, é possível a criação de áreas de cobertura alargada através da interligação de várias BSSs numa *Extended Service Set* (ESS). Essa interligação pode ser realizada através de um sistema de distribuição (DS) e a área de cobertura tem neste caso o nome de *Extended Service Area* (ESA). Tal como apresenta a figura 3.1, a integração com outras redes LAN pode ser realizada através de um *portal*. A lógica desse *portal* pode ser implementada utilizando um *router* ou *bridge*, que está conectado ao DS.

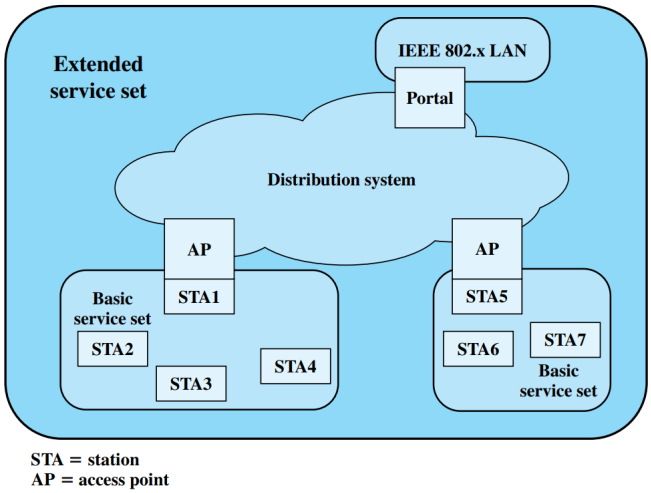


Figura 3.1: Arquitetura de uma rede IEEE 802.11 [7]

3.2 Camada de acesso ao meio (MAC)

3.2.1 Entrega confiável de dados

Como em qualquer tipo de rede sem fios, as comunicações numa WLAN baseada na norma IEEE 802.11 estão sujeitas a ruído, interferência e outros efeitos de propagação que podem significar a perda de tramas. Esta situação pode ser solucionada com mecanismos presentes em protocolos de camadas superiores, como é o caso do TCP. Apesar disso, os temporizadores utilizados para despoletar retransmissões nessas camadas são tipicamente na ordem dos segundos, tornando mais eficiente a existência de retransmissões na camada MAC. Com este propósito, a norma IEEE 802.11 inclui mecanismos que permitem a retransmissão de tramas. Quando uma estação recebe uma trama de dados de outra estação (ou do AP), deve retornar uma trama de confirmação (ACK) para a estação remetente da mesma. Se ao fim de um curto período de tempo a receção desse ACK pela estação remetente não ocorrer, a mesma deverá retransmitir a trama enviada originalmente. Desta forma, o mecanismo básico de transferência de dados entre duas estações (*unicast*) envolve a troca de pelo menos duas tramas entre as estações intervenientes.

De forma a aumentar ainda mais confiabilidade na entrega de dados, um procedimento que envolve a troca de 4 tramas pode ser utilizado, permitindo reduzir a possibilidade de ocorrência do fenómeno conhecido como "terminal escondido". Neste cenário, a estação remetente envia uma primeira trama *Request to Send* (RTS) para a estação destinatária. A mesma responde com uma trama *Clear to Send* (CTS). Após receber uma CTS, a estação remetente transmite a trama de dados que será depois seguida da transmissão do ACK respetivo. Desta forma, o RTS alerta todas as outras estações na vizinhança do emissor, enquanto o CTS avisa as estações na vizinhança do recetor, que uma outra estação pretende transmitir, devendo essas retrair-se de possíveis transmissões de forma a evitar a ocorrência de colisões enquanto a transmissão original decorre.

3.2.2 Controlo de acesso ao meio

A norma 802.11 considera duas abordagens distintas para os algoritmos de controlo de acesso ao meio: cenários em que o mecanismo de acesso ao meio é distribuído, ou seja, em que a decisão de quando deve ser realizada a transmissão é distribuída por todas as estações, com um mecanismo de prevenção e resolução de colisões; e mecanismos de acesso centralizado, que envolvem a regulação da transmissão por um nó centralizado, tipicamente o AP. Desta forma, surgem dois mecanismos distintos de controlo de acesso ao meio [7]:

- ***Distributed Coordination Function*** (DCF): O DCF utiliza um algoritmo de *Carrier Sense Multiple Access* (CSMA), que reduzir a probabilidade de colisões (CSMA/CA). Tal como apresenta a figura 3.2, uma estação que pretende enviar uma trama MAC, deve escutar em primeiro lugar o meio. No caso de verificar que não existe nenhuma outra estação a ocupar o mesmo, deve aguardar por um período de tempo denominado de *Interframe Space* (IFS) até poder transmitir. Se após esse período de tempo o meio permanecer livre, a estação pode iniciar a transmissão.

  Pelo contrário, se após aguardar esse intervalo de tempo, a estação verificar que o meio foi ocupado, deve esperar até que o mesmo seja libertado. Uma vez que tal ocorra, deve esperar novamente por um período igual ao IFS. Após esse período, a estação deve entrar num período aleatório de *backoff*, devendo o contador desse temporizador parar sempre que seja detetado que o meio foi ocupado, voltando a decrementar assim o meio seja libertado. No fim do período de *backoff* a estação pode iniciar a transmissão. Se no fim deste procedimento a transmissão não for concluída com sucesso, ou seja, não for recebido um ACK de receção da trama, é assumido que existiu uma colisão e o procedimento de *backoff* é repetido.

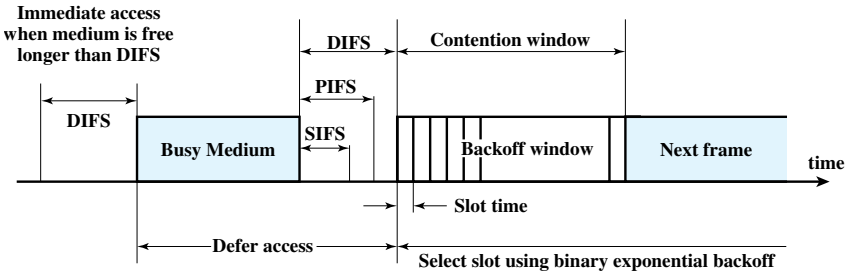


Figura 3.2: Mecanismo de acesso ao meio utilizando DCF [7]

De forma a priorizar certos tipos de tramas, o período inicial de espera (IFS) depende do tipo de trama a transmitir. Assim, o *Short IFS* (SIF), é utilizado para receber mensagens de confirmação (ACK) e na transmissão de múltiplos fragmentos de uma trama. O *Point Coordination Function IFS* (PIFS) é utilizado em operações sobre a função de *Point Coordination Function IFS* (PCF); e o *Distributed Coordination Function* IFS (DIFS) é utilizado pelas estações que utilizam DCF na transmissão de tramas de dados e de gestão.

- ***Point Coordination Function*** (PCF): O PCF é um método alternativo para controlo de acesso ao meio, implementado paralelamente ao DCF. O PCF implementa um sistema de *polling* centralizado para suportar a transmissão assíncrona de mensagens, onde o *Point Coordinator* (PC) opera como coordenador central das transmissões. Devido à necessidade de existência de um coordenador central, este mecanismo opcional é utilizado em redes do tipo infraestruturadas. Apesar disso, grande parte dos APs no mercado não suportam esta tecnologia.

3.2.3 Tramas MAC

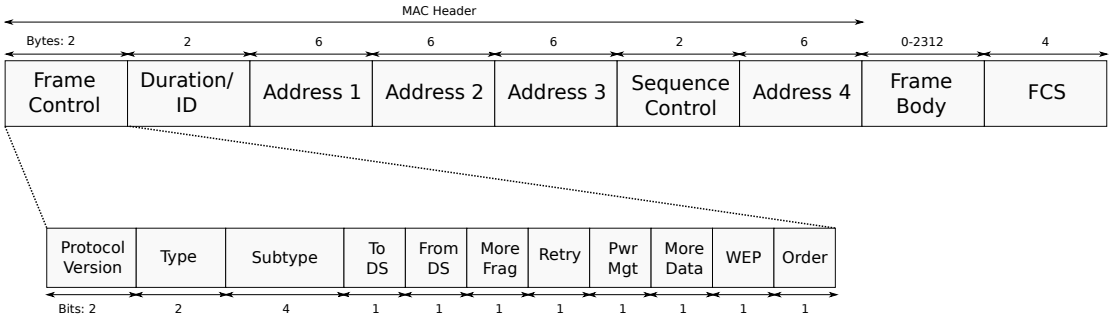


Figura 3.3: Formato genérico de uma trama MAC

As tramas MAC podem ser descritas por uma sequência de parâmetros numa ordem bem definida [34]. Esta é constituída por um cabeçalho (MAC *Header*), pelo conteúdo (*Frame Body*) e por um último campo utilizado para verificação de erros com um código de redundância cíclica (*Frame Check Sequence*). Este formato genérico é utilizado para todo os tipos de tramas (controlo, dados ou gestão), mas nem todos os parâmetros são utilizados em todos os contextos. Em

particular, a informação relativa ao tipo de trama está presente no campo ***Frame control*** que, tal como mostra a figura 3.3, contém um conjunto de parâmetros de controlo relevantes à operação das estações. São eles:

- ***Protocol version***: Indica a versão do protocolo utilizada, devendo as estações descartar tramas referentes a versões do protocolo que as mesmas não suportem.
- ***Type***: Indica qual o tipo de trama: dados, utilizada para a transmissão de dados; gestão, utilizada no controlo de acesso ao meio (RTC, CTS e ACK); ou de controlo, utilizada para a troca de informação de gestão com as estações.
- ***Subtype***: Indica o subtipo da trama, tal como descrito na secção 3.2.4.
- ***To DS***: Apresenta o valor 1 no caso de tramas destinadas a uma estação fora da BSS mas dentro do sistema de distribuição (DS).
- ***From DS***: Apresenta o valor 1 no caso de tramas com origem numa estação fora da BSS mas no sistema de distribuição (DS).
- ***More Flag***: Indica a existência de mais fragmentos pertencentes à trama atual.
- ***Retry***: Indica se a trama atual é uma retransmissão.
- ***Pwr Mgt***: Indica se a estação que enviou a trama vai entrar num estado de poupança de energia. Em caso afirmativo, informa o AP que deve armazenar todas as tramas destinadas a essa mesma estação.
- ***More data***: Serve de indicação às estações da existência de mais tramas armazenadas e a aguardar envio. Em caso afirmativo, indica que as mesmas não devem entrar no estado de poupança de energia até receberem as restantes tramas.
- ***WEP***: Indica se o conteúdo da trama está encriptado.
- ***Order***: Indica que as tramas recebidas terão de ser processadas tendo em conta o número de sequência.

Além do *Frame control*, existe outro conjunto de campos, no cabeçalho MAC, que fornece informação detalhada sobre o endereçamento e ordem de sequência da trama. Apresenta-se de seguida uma descrição de cada um desses campos:

- **Duration/ID**: Indica o tempo (em microsegundos) que determinado canal vai estar alocado para uma transmissão de uma trama MAC. Em algumas tramas de controlo, este campo contém um identificador da associação ou da conexão.
- ***Campos de endereços*** (*Address 1*, *Address 2*, *Address 3* e *Address 4*): O número e significado destes campos depende do contexto. O endereço do transmissor (TA) e do recetor

(RA) são os endereços MAC da estação que emitiu a trama e da próxima estação a recebe-la, respetivamente. Por outro lado, o endereço da estação de destino (DA) e da estação fonte (SA), indicam o endereço MAC da estação final de destino da trama e da estação que criou inicialmente a mesma, respetivamente. Por ultimo, o parâmetro BSSID indica, no caso de uma rede do tipo infraestruturada, o endereço MAC do AP e no caso de uma rede Ad-hoc o endereço MAC da estação que cria a rede. A tabela 3.1, sintetiza as diversas combinações possíveis dos campos de endereços na trama MAC.

Tabela 3.1: Diferentes combinações do campo de endereços na trama MAC

To DS	**From DS**	**Address 1**	**Address 2**	**Address 3**	**Address 4**
0	0	DA	SA	BSSID	N/D
0	1	DA	BSSID	SA	N/D
1	0	BSSID	SA	DA	N/D
1	1	RA	TA	DA	SA

- ***Sequence control***: É composto por dois campos distintos: *Sequence number* (12 bit) que indica um número de sequência que identifica a trama, sendo igual em todas as tramas fragmentadas; e *Fragment numbe*r (4 bit) que indica o número do fragmento, no caso de tramas fragmentadas.
- ***Frame Check Sequence*** (FCS): é um *Cyclic Redundancy Check* (CRC) calculado sobre todos os campos do cabeçalho e conteúdo da trama, permitindo à estação recetora verificar a integridade da mesma.

3.2.4 Subtipos de tramas

Como anteriormente referido, existem três tipos de tramas MAC distintos: controlo, dados e gestão. Cada um destes tipos pode apresentar um conjunto de subtipos, tal como descrito de seguida.

- As tramas de **controlo** permitem assistir a entrega confiável das tramas de dados. Existem três subtipos distintos:
  - ***Power Save-Poll*** (PS-Pool): Esta trama pode ser enviada por qualquer estação para o AP correspondente. O seu propósito é requerer que o AP transmita tramas destinadas a essa estação em particular, que o mesmo tenha armazenado durante o período em que a estação esteve a operar no modo de poupança de energia. Mais detalhes deste mecanismo são referidos na secção 3.3.2.
  - ***Request to Send*** (RTS): A estação que envia esta mensagem indica que pretende enviar uma trama de dados.
  - ***Clear to Send*** (CTS): Serve como resposta ao RTS e informa a estação que originou o mesmo que tem permissão para iniciar o envio da trama de dados.

  - ***Acknowledgment*** (ACK): Fornece a confirmação à estação que enviou uma trama de dados, de gestão ou um pedido PS-Poll, que a mesma foi corretamente recebida pelo destinatário.
  - ***Contention-Free*** (CF)-end: Anuncia o fim do período de contenção.
  - ***CF-End + CF-Ack***: Confirma o *CF-end*, permitindo às estações a libertação de todas as restrições associadas à operação nesse período.

- As tramas de **dados** podem ser divididas em oito subtipos, organizadas em dois grupos distintos. Os primeiros quatro subtipos transportam dados de camadas superiores de uma estação emissora até à estação de destino. São eles:

  - ***Data***: Este é o mais simples tipo de trama de dados. Pode ser utilizado durante o período de contenção e de não contenção.
  - ***Data + CF-Ack***: Deve apenas ser enviado durante um período de não contenção e permite, em adição ao envio de dados, confirmar a receção de tramas de dados anteriores.
  - ***Data + CF-Poll***: Utilizado pelo *point coordinator* para entregar tramas de dados a uma estação móvel e ao mesmo tempo requerer que essa estação envie tramas de dados que tenha armazenado.
  - ***Data + CF-Ack + CF-Poll***: Combina as funções *Data + CF-Ack* e *CF-Poll* numa única trama.

  O restante grupo de tramas deste subtipo, não transporta de facto nenhuns dados da camada de aplicação. Assim, a trama ***Null Function Data*** é utilizada pelas estações móveis para transportar o bit de *Power management* e informar o AP que a mesma está a sair do estado de poupança de energia. Este mecanismo é apresentado com mais detalhe na secção 3.3.2.1. Por outro lado, as três tramas restantes (*CF-Ack*, *CF-Poll* e *CF-Ack + CF-Poll*) têm a mesma funcionalidade dos subtipos de tramas de dados correspondentes (Data + CF-Ack, Data + CF-Poll e Data + CF-Ack + CF-Poll) mas sem a componente de dados.

- As tramas de ***gestão*** são utilizadas para gerir as comunicações entre as estações e os respetivos APs. Existem onze subtipos distintos:

  - ***Beacon***: Esta trama é periodicamente enviado pelo AP de forma a anunciar a sua presença e transmitir um conjunto de informação, como *timestamp*, SSID, o *Traffic indication map* (TIM) ou *Delivery Traffic Indication Map* (DTIM). Este subtipo de trama é analisado com pormenor na secção 3.3.1.
  - ***Probe request***: Esta trama é enviada quando uma estação pretende obter informação sobre outra estação. É, por isso, habitualmente utilizada pelas estações para pesquisar e obter as características de APs existentes nas proximidades. Pode ser enviada com o objetivo de obter informação de um AP com um SSID especifico ou, deixando esse parâmetro a NULL, obter informação de todos os APs que receberem o pedido.

- ***Probe response***: Esta trama é enviada a uma estação como resposta a um *Probe request* e contém informação das capacidades de determinado AP.

- ***Authentication***: O processo de autenticação permite verificar a entidade de uma estação perante um AP, sendo o processo iniciado com um envio de uma trama deste tipo para o AP com o qual a estação se pretende autenticar. Se o sistema de autenticação suportado for *Open System Authentication*, o processo culmina com o envio pelo AP de outra trama deste tipo, indicado à estação o sucesso ou falha do processo. No caso da existência do mecanismo de *Shared Key Authentication*, o AP responde ao pedido inicial com um desafio de texto desencriptado. De seguida, a estação deve encriptar o mesmo utilizado uma chave secreta *Wired Equivalent Privacy* (WEP), remetendo a mesma para o AP. Se o desafio desencriptado pelo AP corresponder ao inicialmente enviado, é enviado a confirmação do sucesso do processo de autenticação à estação.

- ***Deauthentication***: Esta trama é enviada quando uma estação pretende sinalizar o AP que pretende terminar a troca de mensagens seguras entre ambos.

- ***Association request***: O envio desta trama inicia o processo de associação de uma estação com um AP. Esta contém informação referente as especificações suportadas (ex: taxas de transmissão de dados) e o SSID do AP a que a estação se pretende associar. Após receber uma trama deste subtipo, o AP verifica a compatibilidade com as especificações suportadas pela estação, e (se aceite) reserva memória e atribui um *Association ID* (AID) à estação.

- ***Association response***: Esta trama é enviada pelo AP a uma estação que iniciou o pedido de associação e indica a aceitação ou rejeição do mesmo. Nesta trama é transmitido o AID consignado à estação, bem como um conjunto de parâmetros referentes à associação (ex: taxas de transmissão de dados). No caso do processo de associação ser concluído com sucesso, a estação pode iniciar a transmissão de tramas de dados.

- ***Reassociation request***: No caso de uma estação sair do alcance do AP ao qual se encontra associada, pode enviar uma trama deste subtipo para o novo AP ao qual se pretende associar. Este coordena a retransmissão de tramas de dados que ainda estejam armazenadas, para esta estação, no AP anterior.

- ***Reassociation response***: Esta trama contém parâmetros similares aos presentes num *Association response* e é enviada pelo novo AP a uma estação que iniciou o pedido de reassociação, contendo informação sobre a aceitação ou rejeição do pedido.

- ***Disassociation***: Esta trama é enviada quando uma estação pretende informar o AP que deseja terminar graciosamente a associação existente. O AP pode assim libertar memória e recursos que estejam reservados aquela estação em particular.

3.3 Gestão do consumo de energia

Como referido no capítulo 2, as interfaces de comunicação WiFi podem representar uma grande percentagem do consumo total de energia em dispositivos que façam uso das mesmas. Isso é particularmente critico, se considerarmos o cenário típico de IoT e de sistemas de monitorização em saúde, onde os dispositivos, apesar das baterias de tamanho reduzido, devem possuir uma autonomia alargada. Assim, de forma a usufruir dos benefícios da tecnologia WiFi nesses cenários, torna-se imprescindível a utilização de mecanismos de poupança de energia associados a essas mesmas interfaces. De uma forma genérica, uma estação, sem qualquer mecanismo de poupança de energia implementado, deve manter a respetiva interface de comunicação sem fios sempre ativa e pronta a receber pacotes do AP ao qual se encontra associada. Este modo de funcionamento requer que muitos dos circuitos elétricos relativos a essas interfaces se mantenham ininterruptamente ligados, o que pode significar uma redução significativa da duração da bateria nesses dispositivos. Ao longo desta secção, serão discutidos os mecanismos previstos na norma IEEE 802.11 que permitem que as estações desliguem as referidas interfaces, durante períodos de inatividade nas comunicações.

3.3.1 *Beacon frame*

Algumas das funcionalidades previstas na norma IEEE 802.11 [34] requerem que todas as estações (e o AP) permaneçam com os seus relógios sincronizados. Essa sincronização tem um papel especialmente preponderante em estações que estejam a operar em modo de poupança de energia, uma vez que devem comutar de um estado de inatividade para um estado ativo em instantes específicos. O principal mecanismo que suporta a referida sincronização é a transmissão periódica de uma trama de gestão especifica pelo AP: o *Beacon*. Os *Beacons* além de permitirem a sincronização dos relógios de todas as estações pertencentes a determinada BSS, são também utilizados para transmitir informação especifica das características da conexão oferecida, por determinado AP, aos restantes membros dessa BSS. A figura 3.4 mostra uma captura de um *Beacon frame* transmitido por um AP e os respetivos parâmetros presentes no mesmo.

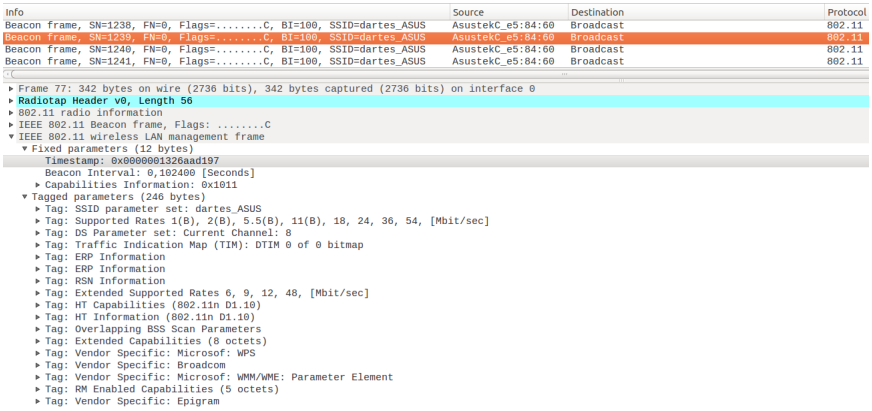


Figura 3.4: Captura de um *Beacon frame*

Como anteriormente mencionado, este tipo de trama suporta alguns dos mecanismos de poupança de energia disponíveis para as estações WiFi. Apresenta-se de seguida uma descrição mais detalhada de algum dos elementos mais relevantes para a implementação dos referidos mecanismos:

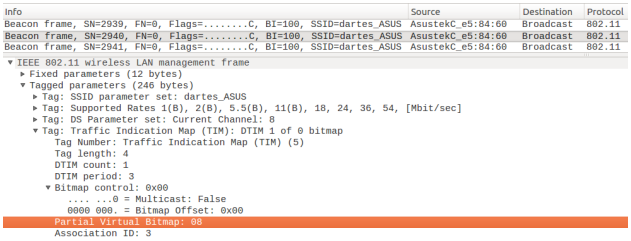
1. ***Timestamp*** (8 bytes): Este parâmetro representa o número de microsegundos desde que o AP iniciou o serviço e é o que permite suportar a sincronização temporal das diferentes estações, podendo as mesmas atualizar os seus relógios locais através deste parâmetro.

2. ***Beacon Interval*** (2 bytes): Este parâmetro representa o intervalo de tempo em *Time units* (TUs) entre duas *Target Beacon Transmission Times* (TBTT). Desta forma, um *Beacon interval* de 100 representa:

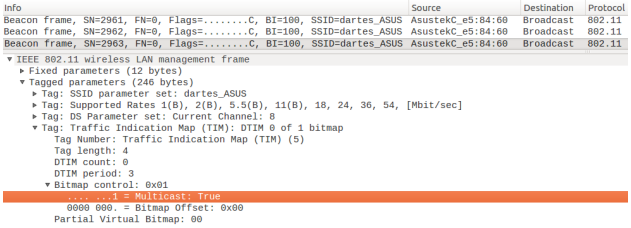
$$BeaconInterval = 100TU = 100 * 1024us = 102.4ms$$

3. ***Traffic Indication Map*** (TIM): O *Traffic Indication Map* (TIM) serve como indicação de quais estações associadas possuem tráfego pendente e armazenado no AP. Este elemento contém 4 campos distintos: *DTIM count*, *DTIM period*, *Bitmap control* e o *Partial Virtual Bitmap*.

   Durante o processo de associação, cada estação obtém do AP um *Association ID* (AID). Esse AID é usado pelo AP para identificar no *Partial Virtual Bitmap* do TIM quais as estações que possuem tráfego *unicast* pendente. Assim, no caso do bit número N deste elemento ser igual a um significa que a estação cujo AID é N tem tráfego pendente no ponto de acesso. Pelo contrário, um zero no bit número N significa que a estação cujo AID é N não possui tráfego pendente. Na eventualidade de não existir tráfego *unicast* pendente para nenhuma das estações associadas, o *Partial Virtual Bitmap* é codificado como um único *byte*.



(a) TIM com indicação de tráfego *unicast* armazenado para a estação com o AID 3



(b) (D)TIM com indicação de tráfego *multicast* armazenado no AP

Figura 3.5: Captura de um *Beacon frame* mostrando a mensagem (D)TIM

O *Delivery Traffic Indication Map* (DTIM) é transmitido com o *Beacon* a cada DTIM *Period* e é o equivalente ao TIM, mas para o tráfego *multicast*/*broadcast*. O campo DTIM *count* indica quantos *Beacons* (incluindo o atual) faltam até ao próximo DTIM, sendo que um DTIM *count* de valor zero indica que o atual TIM é um DTIM. Assim, o bit zero do *Bitmap control* assume o valor um no caso de existirem *frames multicast* ou *broadcast* pendentes; ou o valor zero na situação contrária. Após um DTIM, o AP procede ao envio dos *frames multicast*/*broadcast* armazenados, antes do envio de quaisquer *frames unicast*.

No decorrer da próxima secção, serão objeto de estudo os mecanismos utilizados pelas estações integradas numa topologia do tipo infraestruturada e a forma como utilizam a informação presente no *Beacon* para ativar os mecanismos de poupança de energia.

3.3.2 Poupança de energia nas estações de redes infraestruturadas

Numa topologia do tipo infraestruturada, o AP deve manter um registo atualizado do modo de operação atual de cada uma das estações associadas. Por seu lado, as estações devem comunicar o estado do seu modo de operação, refletindo essa configuração no bit de *Pwr Mgt* no *Frame Control* (3.2.3) das tramas transmitidas. No caso do AP receber tramas de dados, destinadas a uma estação a operar no modo de poupança de energia, deve armazenar as mesmas e sinalizar a sua existência, utilizando o AID dessa estação, no TIM dos *Beacons* transmitidos. Por seu lado, uma estação que detete que o bit correspondente ao seu AID é colocado a *true* no TIM, deve requerer a entrega das tramas armazenadas enviado um PS-Poll para o AP.

Após a receção de um PS-Poll, o AP procede à transmissão de uma trama armazenada e aguarda a confirmação de receção da mesma por parte da estação que efetuou o pedido. A receção de pedidos adicionais de PS-Poll, enviados pela mesma estação, é ignorada até que a receção da trama enviada anteriormente seja confirmada. Isto previne que retransmissões desses pedidos sejam consideradas como novas solicitações. Após a receção com sucesso de uma trama, as estações devem consultar o bit *More Data* no *Frame Control* (3.2.3) e verificar a existência de tramas adicionais pendentes no AP. No caso de as mesmas existirem, deve ser repetida a sequência descrita.

Desta forma, as estações podem permanecer com as interfaces de comunicação sem fios desligadas, durante períodos de inatividade nas comunicações. O período máximo para esse intervalo é comunicado ao AP pela estação, no processo de associação pelo parâmetro *Listen Interval*. Este indica o número máximo de *Beacons* que a estação pode ficar sem retornar do estado de poupança de energia.

Adicionalmente, uma vez que o TIM apenas sinaliza a presença de tramas *unicast* armazenadas, as estações devem interromper o modo de poupança de energia para receber tráfego *broadcast* ou *multicast*. O envio desse tipo de tráfego é realizado em momentos predeterminados pelo AP, e é indicado no Beacon pelo parâmetro DTIM *interval*. Desta forma, as tramas *broadcast* ou *multicast* são remetidas pelo AP logo após um Beacon com o parâmetro DTIM *Count* a zero e a flag de *multicast* a um (1).

A figura 3.6 mostra um exemplo [8] de operação do modo de poupança de energia, assumindo um período de *Listen interval* de um *Beacon* para a STA 1, de dois *Beacons* para a STA 2 e um DTIM *Period* de cinco.

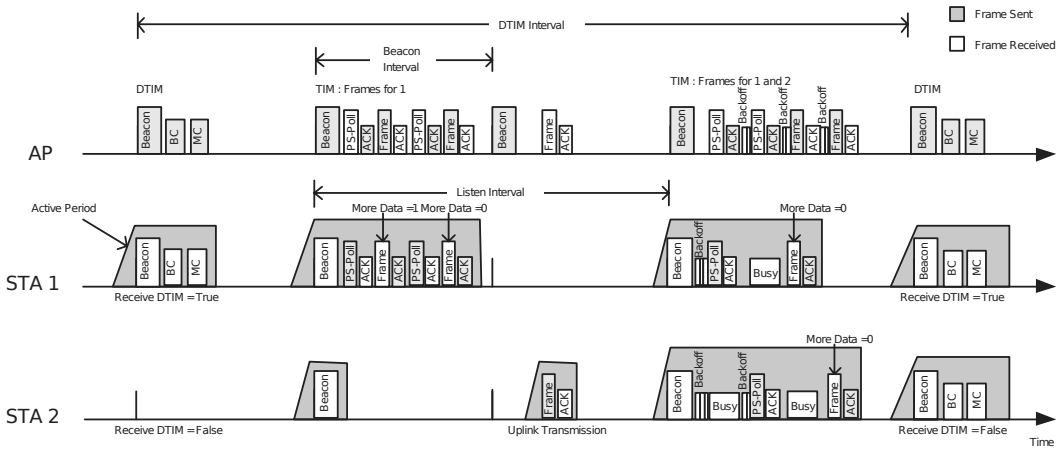


Figura 3.6: Exemplo de operação do modo de poupança de energia [8]

3.3.2.1 *Flipping* do bit de gestão de consumo energético em tramas de dados

O método descrito na secção 3.3.2 não é o único utilizado por estações para recuperar tramas *unicast* armazenadas no AP. Em vez do envio de pedidos explícitos de PS-Poll ao AP, as estações podem comunicar a mudança do modo de funcionamento, através do envio de tramas do tipo *Null data* com o bit de Pwr Mgt do *Frame control* a um. De forma sucinta, esse procedimento pode ser realizado da seguinte forma:

1. A estação envia uma trama do tipo *Null Data* com o bit de Pwr Mgt a um, antes de desligar as interfaces de comunicação e entrar num modo de poupança de energia. Isto sinaliza o AP que deve iniciar o processo de armazenamento de todos as tramas *unicast* destinadas aquela estação em especifico.

2. Posteriormente, assim que a estação retorne do estado de poupança de energia, deve enviar novamente uma trama do tipo *Null Data* com o bit de Pwr Mgt a zero. O procedimento de retornar do estado de poupança de energia, pode ser realizado periodicamente e independentemente do estado do TIM (de acordo com o parâmetro *Listen interval*). Algumas implementações consideram o despertar, por curtos períodos de tempo, apenas da interface WiFi de receção para ouvir o conteúdo da informação contida no TIM do *Beacon*. No caso deste indicar que existem tramas armazenadas, a estação pode então recuperar esse tráfego, utilizando a estratégia referida.

3. Após a receção de uma trama desse tipo, o AP para o armazenamento de tráfego para aquela estação em especifico, e envia as tramas armazenadas para a mesma. Ao contrário do que acontecia com os pedidos de PS-Poll, as tramas armazenadas são enviadas todas de seguida, evitando a necessidade de envio de um pedido PS-Poll por cada trama armazenada.

4. Posteriormente à receção com sucesso de todos as tramas armazenadas, a estação pode retornar ao estado de poupança de energia repetindo o processo descrito em 1.

3.4 Sumário

Neste capítulo foram apresentados os aspetos mais relevantes relacionados com a camada de acesso ao meio (MAC), definidos na norma IEEE 802.11, e que permitem a operação de estações WiFi em modos de poupança de energia. Foi identificando que a transmissão periódica pelo AP de uma trama do tipo *Beacon*, é um procedimento essencial para a habilitação desses modos de funcionamento, permitindo a sincronização entre as estações e o respetivo AP. Adicionalmente, esta trama permite a identificação da existência de tráfego armazenado no AP destinado a estações a operar no modo de poupança de energia. A informação sistematizada ao longo deste capítulo, irá ser utilizada para melhor compreender a operação nos diferentes modos de poupança de energia do módulo WiFi ESP8266 descritos no capítulos 4 e 5.

Capítulo 4

Módulo WiFi ESP8266

Este capítulo apresenta as principais características do módulo ESP8266, que vai ser usado como base para o trabalho desta dissertação. Trata-se de um módulo recente com interface WiFi que é apresentado como de baixo custo e de consumo energético ultra-baixo, propriedades que vamos investigar quantitativamente.

4.1 Visão geral

O módulo WiFi ESP8266, fabricado pela Espressif, oferece uma solução *System-on-a-chip* (SoC) integrada, que afirma satisfazer os requisitos de aplicações de *Internet of Things* (IoT) e de comunicação M2M, ao garantir a disponibilidade de mecanismos de poupança de energia associados à transmissão de informação utilizando a tecnologia WiFi.

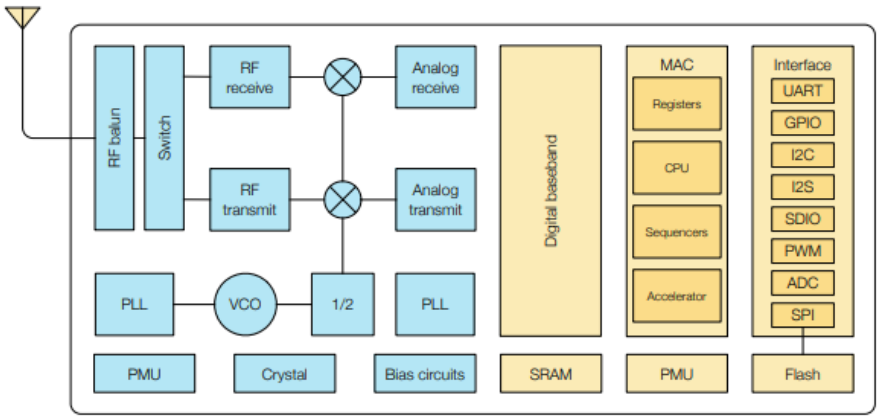


Figura 4.1: Diagrama de blocos funcional do ESP8266 [9].

Na figura 4.1 encontra-se representado o diagrama de blocos de alto nível deste módulo. De entre as principais características do mesmo que vão de encontro às necessidades de aplicações IoT, podem-se destacar as seguintes [9]:

- Tamanho reduzido (24x16mm), possibilitando a integração em diferentes sistemas embebidos, onde se incluem os wearables.
- Conformidade com o protocolo IEEE 802.11b/g/n (WiFi) e antena embutida, permitindo aplicações de sensorização com uma ligação à Internet sem a utilização de gateways.
- Suporte para protocolos de segurança para a comunicação WiFi (WPA/WPA2).
- Vários modos de poupança de energia disponíveis.
- Integra um processador de 32 bits, com capacidades de baixo consumo e com uma frequência máxima de operação de 160MHz (Tensilica L106 32-bit RISC).
- Dispõe de suporte para um sistema operativo de tempo real (FreeRTOS).
- Modo de operação como estação WiFi, *softAP* e estação+*softAP*.
- Implementação de toda a *stack* TCP/IP.
- 36kB de memória interna SRAM e 4MB de memoria externa SPI *flash*.
- Baixo custo quando comparado com outros módulos WiFi com características semelhantes.

4.2 *Frameworks* de desenvolvimento

O *ESP8266_SDK* é o *Software Development Kit* (SDK) oficialmente suportado pelo ESP8266 e disponibilizado pela Espressif Systems. Providencia aos programadores uma serie de interfaces que permitem a conexão a uma rede WiFi, a ativação dos diferentes modos de poupança de energia e o envio e receção de dados. Está disponível em duas versões distintas:

- RTOS SDK [35]: Esta versão é baseada no sistema operativo de tempo real FreeRTOS, disponibilizando desta forma as interfaces *standard* de gestão orientada a tarefas que este sistema operativo oferece [36]. Adicionalmente, a integração com com a pilha protocolar TCP/IP é realizada através da API *open-source* lwIP (*lightweight* IP). Fornece suporte a um conjunto importante de bibliotecas como cJSON ou o MQTT Paho.
- Non-OS SDK [37]: Ao contrário da versão anterior, esta não é baseada em nenhum sistema operativo de tempo real. Ainda assim, disponibiliza *timers* e permite a programação de funções que podem ser despoletadas periodicamente. A integração com com a pilha protocolar TCP/IP é realizada através de uma biblioteca proprietária (espconn).

Existem ainda algumas alternativas não oficiais a estes dois últimos SDKs. Uma das que mais se destaca é o *esp-open-rtos* [38], por ser um projeto totalmente *open-source* e dispor de uma grande comunidade ativa. Apesar de ser baseado no RTOS SDK da *Espressif Systems*, não suporta nenhum dos modos de poupança de energia descritos na secção 4.3 e não dispõe de documentação de apoio ao desenvolvimento.

4.3 Gestão do consumo de energia

Como referido anteriormente este módulo dispõe de modos de funcionamento que possibilitam que o CPU e/ou as interfaces de comunicação *wireless* sejam desligadas durante períodos de inatividade, permitindo um incremento da autonomia de dispositivos alimentados por bateria. Tal como discutido no capítulo 3.3, esses modos de funcionamento baseiam-se nos princípios definidos no standard IEEE 802.11, e permitem que os referidos periféricos sejam desligados durante períodos de tempo iguais ao fator $BeaconInterval \times DTIMPeriod$. Tendo em conta esse principio, o módulo em estudo suporta o funcionamento em três modos de poupança de energia distintos: *Modem-sleep*, *Light-sleep* e *Deep-sleep*. A tabela 4.1 apresenta um resumo das principais características de cada um.

Tabela 4.1: Modos de poupança de energia: distinção a nível dos componentes em funcionamento e da corrente consumida em cada um dos modos, tal como anunciado na *datasheet* [9].

Item	**Modem-sleep**	**Light-sleep**	**Deep-sleep**
Interface WiFi	OFF	OFF	OFF
Associação ao AP	Conectado	Conectado	Desconectado
Relógio de sistema	ON	OFF	OFF
RTC	ON	ON	ON
CPU	ON	Pendente	OFF
Corrente no substrato	15mA	0.4mA	∼20uA
Corrente média (DTIM=1)	16.2mA	1.8mA	-
Corrente média (DTIM=3)	15.4mA	0.9mA	-
Corrente média (DTIM=10)	15.2mA	0.55mA	-

4.3.1 Características dos modos de poupança de energia

4.3.1.1 *Modem-sleep*

Este modo de funcionamento é o que está definido como configuração padrão e permite que, de forma automática, as interfaces de comunicação *wireless* sejam desligadas entre intervalos de mensagens DTIM. Antes da chegada da DTIM seguinte, o módulo deve retornar do estado de poupança de energia e verificar a existência de tráfego armazenado no AP, realizando os procedimentos de recuperação dos pacotes armazenados nos casos aplicáveis. Durante o período de baixo consumo, os restantes periféricos (CPU, relógio de sistema, ...) são mantidos em funcionamento, permitindo a sua utilização pelas aplicações mesmo durante esses períodos.

4.3.1.2 *Light-sleep*

O principio de funcionamento deste modo é semelhante ao aplicado ao modo de *Modem-sleep*, com a diferença que o relógio de sistema é desligado e o CPU é colocado num estado de suspensão, permitindo uma economia de energia extra. Desta forma, este modo pode ser utilizado em aplicações que não necessitam de uma exatidão temporal elevada (apenas o RTC se mantem em funcionamento) e que podem dispensar o uso do CPU durante os períodos de baixo consumo.

Adicionalmente, tal como a figura 4.2 mostra, a ativação deste modo de funcionamento inclui um mecanismo que comuta automaticamente entre este último e o modo de *Modem-sleep*. Esse procedimento acontece sempre que durante o período em que o módulo estaria com o CPU desligado, existirem tarefas do utilizador ou do sistema operativo que necessitem de ser executadas.

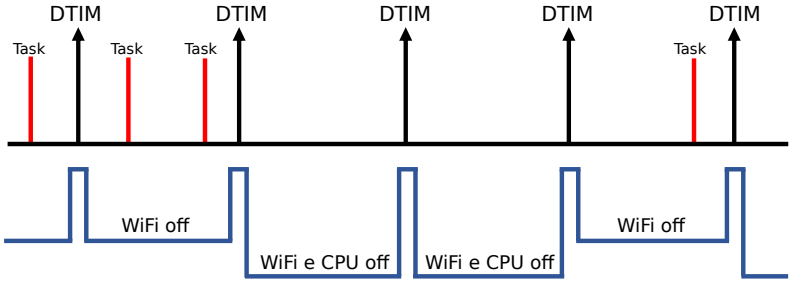


Figura 4.2: Mecanismo de comutação automática entre o modo de *Modem-sleep* e modo de *Light-sleep*. Baseado em [10].

4.3.1.3 Deep-sleep

Este modo de funcionamento, ao contrário dos restantes, só pode ser ativado com a programação de uma ordem especifica para o fazer e não de forma automática. Durante o período de baixo consumo, apenas o RTC se mantêm em funcionamento e é o mesmo que é responsável por fazer o módulo retornar do estado de baixo consumo. De forma a habilitar esse procedimento, o pino GPIO16 deve ser ligado ao EXT_RSTB.

Como durante a operação neste modo de funcionamento todos os restantes periféricos são desligados, os conteúdos e variáveis do programa a executar são perdidos. No caso de se pretender guardar alguma informação entre dois períodos de ativação deste modo de funcionamento, é obrigatória a utilização da memória não volátil associada ao RTC. Adicionalmente, e pela mesma razão, a associação com o AP será perdida durante a ativação deste modo. É por isso apenas recomendável a utilização deste para aplicações que não pretendem transmitir ou receber dados por longos períodos de tempo, uma vez que o custo energético e latência de constantes associações ao AP são elevados.

4.3.2 Modelo teórico do consumo de corrente

Como referido na secção 4.3, a implementação dos modos de poupança de energia, *Modem-sleep* e *Light-sleep*, nos módulos ESP8266, estabelece que o intervalo de tempo de operação, du-

rante o período de baixo consumo, deve estar relacionado com a configuração do fator $BeaconInterval \times DTIMPeriod$. Assim, o consumo de corrente médio, em cada um dos modos de funcionamento, estará intimamente relacionado com a configuração desses parâmetros no AP. Desta forma, o fabricante estabelece em [39] um modelo para o consumo de corrente médio expectável para diferentes configurações daqueles parãmetros.

Esse modelo é apresentado na Equação 4.1 e estabelece um patamar base de $I_{inativo}$, referente ao consumo médio quando o módulo se encontra no estado de baixo consumo, em cada um dos modos de funcionamento, e prevê que o módulo acorde periodicamente de acordo com o fator $BeaconInterval \times DTIMPeriod$ durante um intervalo de tempo igual a t_{ativo}. Como explicado, durante esse período de tempo as interfaces WiFi e os restantes periféricos são ativados, o que representa um consumo de corrente adicional de I_{ativo}.

$$I_{sleep-mode} = I_{inativo} + \frac{t_{ativo}}{DTIMPeriod \times BeaconInterval} \times I_{ativo} \tag{4.1}$$

A folha de características do módulo ESP8266 [9] anuncia, para cada um dos dois modos de poupança de energia mencionados, os seguintes parâmetros:

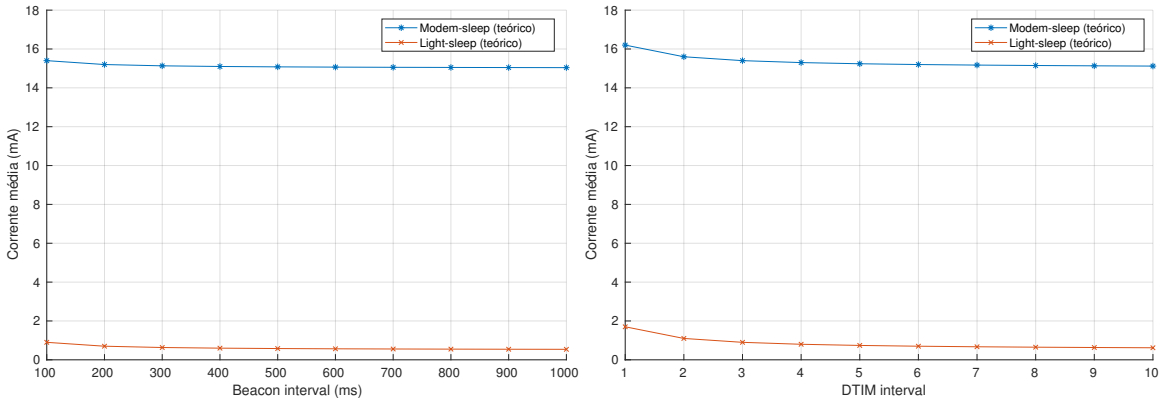
- ***Modem-sleep***: consumo de corrente de 15mA quando a interface WiFi se encontra desligada e um período de ativação da mesma de 2ms, a cada DTIM *Period*, com um consumo de corrente médio adicional de 60mA durante esse período, o que se traduz no seguinte:

$$I_{modem-sleep} = 15mA + \frac{2ms}{DTIMPeriod \times BeaconInterval} \times 60mA \tag{4.2}$$

- ***Light-sleep***: consumo de corrente de 0.9mA quando a interface WiFi, o CPU e o relógio de sistema se encontram desligados e um período de ativação desses periféricos de 2ms, a cada DTIM *Period*, com um consumo de corrente médio adicional de 60mA durante esse período, o que se traduz no seguinte:

$$I_{light-sleep} = 0.5mA + \frac{2ms}{DTIMPeriod \times BeaconInterval} \times 60mA \tag{4.3}$$

Os gráficos da figura 4.3 mostram a evolução do consumo médio de corrente teórico calculado com base nas equações 4.2 (*Modem-sleep*) e 4.3 (*Light-sleep)*. A figura 4.3a projeta os valores expectáveis no caso em que o DTIM *Period* é fixado em 3, e é alterado o valor do *Beacon Interval* de 100 a 1000ms. Por outro lado, a figura 4.3b apresenta esses valores numa configuração onde o *Beacon Interval* é fixado em 100ms, e é alterado o valor do DTIM *Period* de 1 a 10.



(a) DTIM *Period* = 3 e *Beacon Interval* = 100-1000ms (b) DTIM *Period* = 1-10 e *Beacon Interval* = 100ms

Figura 4.3: Comportamento do modelo teórico de consumo de corrente *Beacon Interval* e DTIM *Period*

4.4 Sumário

Este capítulo apresentou as características principais dos módulo WiFi ESP8266, com particular destaque para os diferentes modos de poupança de energia disponíveis. Estes modos diferem entre si no número e tipo de periféricos que se mantêm ativos no período de baixo consumo. No caso do modo de *Modem-sleep* e de *Light-sleep* a alternância entre o modo de poupança de energia e o modo ativo pode ser feita de forma automática, estando o período de tempo que o módulo permanece no modo de poupança de energia relacionado com o fator $BeaconInterval \times DTIMPeriod$. Assim, o consumo de corrente médio nestes modos de funcionamento está relacionado com a configuração desses parâmetros, tendo sido apresentados os valores fornecidos pelo fabricante para diferentes valores dos mesmos.

Capítulo 5

Caracterização experimental do módulo ESP8266

Este capítulo descreve a metodologia seguida e os resultados obtidos na caracterização experimental do módulo WiFi ESP8266. Nesta caracterização foi realizada a medição e avaliação do padrão de consumo de corrente nos diferentes modos de poupança de energia disponíveis com a variação de configurações de rede, nomeadamente o *Beacon Interval* e o DTIM *Period*, quantificando desta forma o impacto da infraestrutura WiFi no consumo energético destes módulos. Foram ainda avaliados parâmetros de conectividade num ambiente interior, nomeadamente *Received Signal Strength Indicator* (RSSI), *Packet Delivery Ratio* (PDR) e *Round-trip delay* a várias distâncias e com diferentes obstáculos entre os emissor e o recetor.

5.1 Metodologia e ferramentas utilizadas

As experiências de caracterização do módulo WiFi ESP8266, apresentadas ao longo deste capítulo, foram realizadas tendo em vista um cenário onde se pretende estudar a incorporação deste módulo WiFi num dispositivo, ou *wearable*, para uma aplicação de monitorização móvel *indoor* de sinais fisiológicos. Desta forma, foi considerada a ligação à Internet, em continuo, através da conexão do módulo a uma rede WiFi do tipo infraestruturada, utilizando como AP um *ASUS RT-AC87U dual-band AC240030* [40]. Para a configuração dos diferentes modos de poupança de energia e programação das tarefas necessárias para o envio e receção de dados, foi utilizado um SDK baseado no sistema operativo de tempo-real FreeRTOS, apresentado em 4.2.

Nas experiências que envolveram a medição da corrente consumida pelo módulo foi utilizado o *Low Voltage Power Monitor* (FTA22D), fabricado pela *Monsoon Solutions Inc* [41]. Este dispositivo suporta tensões de saída entre 2.1V e 4.5V, uma corrente máxima de 3A, com uma resolução máxima de 2.86μA, e é capaz de realizar medições com uma taxa de amostragem de 5kHz. Desta forma o módulo foi alimentado diretamente pelo *Power Monitor* a 3.35V, sendo os registos do consumo de corrente guardados no formato CSV e posteriormente processados utilizando a ferramenta Matlab. Os gráficos de consumo de corrente, apresentados ao longo deste capítulo, contém

no eixo do YY o valor instantâneo ou o valor médio do consumo de corrente, e no eixo dos XX o intervalo de tempo ou a configuração em que esse valor médio foi obtido. Em alguns dos gráficos o eixo dos YY foi cortado para efeitos de visualização.

A figura 5.1 apresenta a montagem experimental realizada para a medição do consumo de energia, mostrando a alimentação direta do módulo ESP8266 pelo *Power Monitor*, com a separação dos restantes componentes da placa de desenvolvimento NodeMCU (interface serie, conversor de tensão 5V/3.3V, ...) de forma a isolar e aferir apenas o consumo relativo ao mesmo.

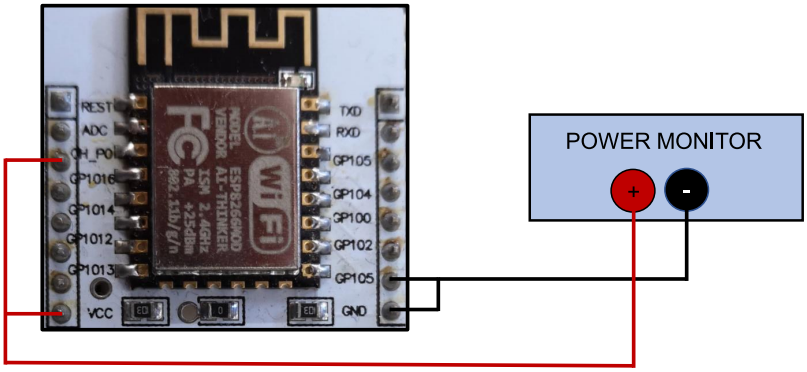


Figura 5.1: Esquema da montagem experimental para a medição do consumo de corrente

Por último, de forma a melhor interpretar e correlacionar o padrão de consumo observado com os pacotes trocados entre o módulo e o AP, foram realizadas, em simultâneo com as medições do consumo de corrente, capturas de pacotes de rede colocando um computador (com *Ubuntu 16.04 LTS*) e o respetivo adaptador *wireless* (*Intel Dual Band Wireless-AC 7265*) em modo monitor.

Em resumo, no decorrer do atual capítulo serão apresentados com detalhe os resultados e conclusões obtidas com cada uma das seguintes experiências:

- Impacto no consumo energético da manutenção da conexão à infraestrutura WiFi, quando o módulo é configurado para operar no modo de poupança de energia *Modem-sleep* (5.2).
- Impacto de diferentes configurações no AP dos parâmetros *Beacon* e DTIM *interval* no consumo energético nos modos de poupança de energia *Modem-sleep* e *Light-sleep* (5.3).
- Avaliação de parâmetros de conectividade num ambiente interior, tais como: *Received Signal Strength Indicator* (RSSI), *Packet Delivery Ratio* (PDR) e *Round-trip delay* (RTD) (5.4).

5.2 Impacto da infraestrutura WiFi

Esta secção apresenta as experiências preliminares realizadas para avaliar o impacto da conexão do módulo ESP8266 à infraestrutura WiFi, com o modo *Modem-sleep*. Nomeadamente, é analisado qual o padrão de alternância entre o modo ativo e o referido modo de poupança de energia, os respetivos consumos em cada um dos diferentes estados e o tipo de pacotes trocados entre o módulo e o AP nesses processos, estando o AP configurado para difundir o *Beacon* a cada

100ms e anunciar a mensagem DTIM com um intervalo de 3 *Beacons*. Adicionalmente, de forma a assegurar que o consumo observado era apenas relativo a fatores relacionados com o próprio módulo e às interações automáticas entre o mesmo e o AP, a presente experiência foi realizada sem transmissão ou receção de dados e sem outras estações WiFi associadas ao AP em questão. Desta forma, após o processo de autenticação e associação ao AP, o módulo foi configurado para executar uma única tarefa, com uma periodicidade de ativação bastante superior ao fator que despoleta o acordar do mesmo ($Beaconinterval \times DTIMinterval$), colocando assim o módulo num estado de inatividade, que permitiu assim aferir e isolar o impacto da manutenção da conexão à infraestrutura WiFi. Por último, de entre os quatro modos de operação disponíveis, foi utilizado o modo de poupança de energia *Modem-sleep*, pois permite preservar a associação ao AP e mantêm o CPU em funcionamento, permitindo eliminar também possíveis variações provocadas pela mudança de estado de funcionamento do CPU no decorrer da experiência.

5.2.1 Configurações padrão do AP

A figura 5.2 apresenta uma amostra de 5s do consumo de corrente observado nas condições descritas anteriormente e com todos os restantes parâmetros de rede deixados na configuração padrão. Adicionalmente, a realização de uma captura simultânea de pacotes permitiu correlacionar o padrão de consumo instantâneo observado com os pacotes enviados e recebidos pelo módulo.

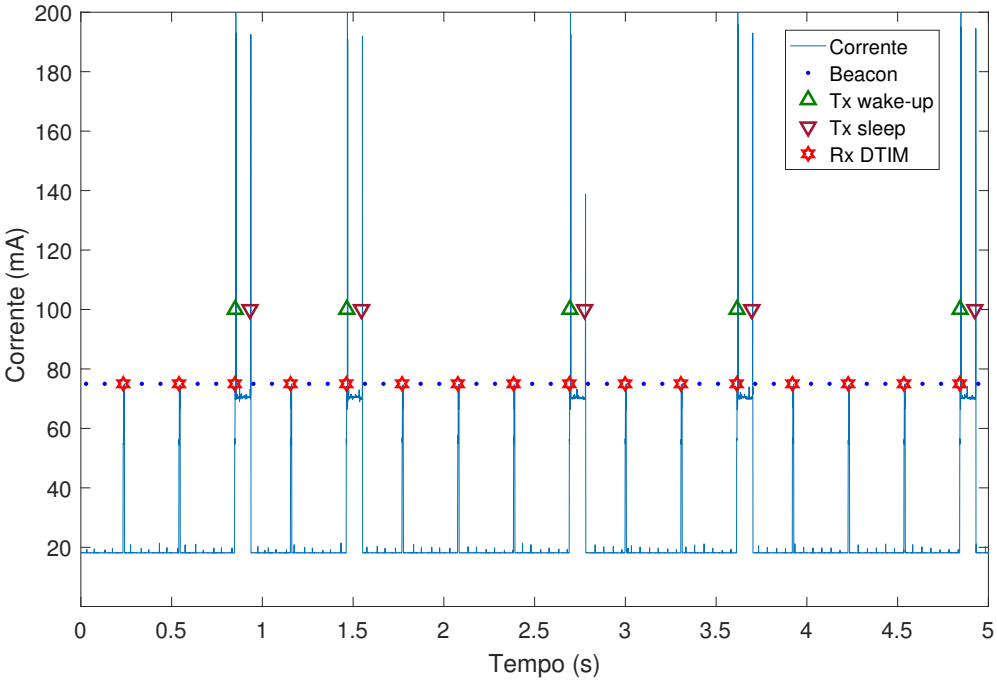
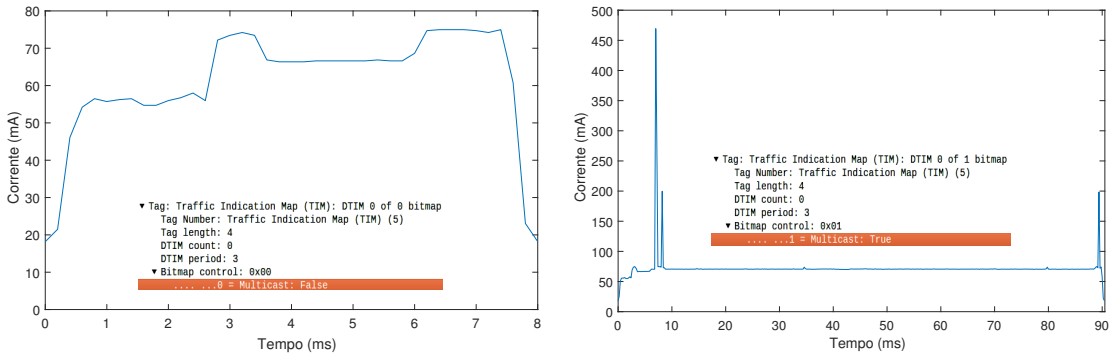


Figura 5.2: Padrão de consumo de corrente instantâneo no modo *Modem-sleep* anotado com informação da captura de pacotes simultânea (*Beacon interval* = 100 ms e DTIM period = 3)

Como se verifica na figura 5.2, o despertar do módulo é síncrono com a receção de cada um dos *Beacons* com a mensagem DTIM presente e na maior parte das ocasiões o módulo retorna do

estado de poupança de energia por uma pequena fração de milissegundos, voltando ao mesmo após receber e processar as referidas mensagens. Ainda assim, é possível verificar que em algumas das vezes que esse procedimento ocorre, o módulo mantém-se ativo por um período de tempo mais longo que o observado nas restantes situações. A análise dos registos de capturas permitiu aferir que os diferentes padrões de consumo observados podem ser explicados pelos valores distintos observados na *flag* de *Multicast* do TIM *bitmap control*, indicando a existência ou não de pacotes *multicast* armazenados no AP.



(a) Sem *multicast/broadcast frames* pendentes (b) Existência *multicast/broadcast frames* pendentes

Figura 5.3: Padrão de consumo instantâneo de corrente com diferentes valores do bit de *multicast* no TIM *bitmap control*

A figura 5.3a mostra a situação onde o referido bit de *multicast* é falso, indicando que o AP não possui tráfego *broadcast* ou *multicast* armazenado. Dessa forma, o módulo acorda apenas para receber e processar o Beacon e retoma o estado de baixo consumo imediatamente após. Este processo demora cerca de 8 ms e apresenta um consumo médio de corrente de 61.4 mA. Em contrapartida, a figura 5.3b mostra a situação onde mesmo o bit é verdadeiro, indicado dessa forma a existência de tráfego *broadcast* ou *multicast* armazenado e a aguardar envio. O módulo deve nestes casos acordar para receber os referidos pacotes, sendo a diferente duração do processo e os picos de corrente observados devido ao mecanismo de mudança de estado do modo de poupança de energia (de *Modem-sleep* para ativo) e a respetiva sinalização dessa alteração ao AP. Como discutido na Secção 3.3.2.1, esse procedimento pode ser efetuado através do envio de um *Null Data Frame* com o bit de gestão de energia configurado a zero para o AP, indicando a disponibilidade do módulo para receber os *frames* armazenados. Após esse procedimento, o módulo permanece no estado ativo por um curto período de tempo, regressando ao modo de poupança de energia com o envio de um *Null Data Frame* com o bit de gestão de energia configurado a um. Todo o processo tem uma duração de aproximadamente 90 ms (aumento de 1025% em relação ao procedimento de ouvir o DTIM) e apresenta um consumo médio de corrente de 71.4 mA (aumento de 16.3% em relação ao procedimento de ouvir o DTIM).

Uma vez que os módulos foram colocados num estado de inatividade e não existindo outras estações associadas, a existência de tráfego *broadcast*, sinalizada pelo AP, apenas poderá ser proveniente do próprio AP. Tal como apresentado na figura 5.4, verifica-se que o protocolo *Spanning*

Tree Protocol (STP) estava ativo (configuração padrão), sendo o responsável por gerar as frames de broadcast sinlaizadas pelo AP.

Info	Source	Destination	DTIM count	Multicast	Protocol
Beacon frame, SN=1941, FN=0, Flags=........C, BI=100, SSID=dartes_ASUS	AsustekC_e5:84:60	Broadcast	0	False	802.11
Beacon frame, SN=1942, FN=0, Flags=........C, BI=100, SSID=dartes_ASUS	AsustekC_e5:84:60	Broadcast	2	False	802.11
Beacon frame, SN=1943, FN=0, Flags=........C, BI=100, SSID=dartes_ASUS	AsustekC_e5:84:60	Broadcast	1	False	802.11
Beacon frame, SN=1944, FN=0, Flags=........C, BI=100, SSID=dartes_ASUS	AsustekC_e5:84:60	Broadcast	0	False	802.11
Beacon frame, SN=1945, FN=0, Flags=........C, BI=100, SSID=dartes_ASUS	AsustekC_e5:84:60	Broadcast	2	False	802.11
Beacon frame, SN=1946, FN=0, Flags=........C, BI=100, SSID=dartes_ASUS	AsustekC_e5:84:60	Broadcast	1	False	802.11
Beacon frame, SN=1947, FN=0, Flags=........C, BI=100, SSID=dartes_ASUS	AsustekC_e5:84:60	Broadcast	0	False	802.11
Beacon frame, SN=1948, FN=0, Flags=........C, BI=100, SSID=dartes_ASUS	AsustekC_e5:84:60	Broadcast	2	False	802.11
Beacon frame, SN=1949, FN=0, Flags=........C, BI=100, SSID=dartes_ASUS	AsustekC_e5:84:60	Broadcast	1	False	802.11
Beacon frame, SN=1950, FN=0, Flags=........C, BI=100, SSID=dartes_ASUS	AsustekC_e5:84:60	Broadcast	0	True	802.11
Conf. Root = 32768/0/54:a0:50:e5:84:60 Cost = 0 Port = 0x8002	AsustekC_e5:84:60	Spanning-tree-(for…			STP
Null function (No data), SN=15, FN=0, Flags=.......TC	Espressi_c0:cf:18	AsustekC_e5:84:60			802.11
Null function (No data), SN=16, FN=0, Flags=...P...TC	Espressi_c0:cf:18	AsustekC_e5:84:60			802.11
Beacon frame, SN=1952, FN=0, Flags=........C, BI=100, SSID=dartes_ASUS	AsustekC_e5:84:60	Broadcast	2	False	802.11
Beacon frame, SN=1953, FN=0, Flags=........C, BI=100, SSID=dartes_ASUS	AsustekC_e5:84:60	Broadcast	1	False	802.11
Beacon frame, SN=1955, FN=0, Flags=........C, BI=100, SSID=dartes_ASUS	AsustekC_e5:84:60	Broadcast	0	False	802.11
Beacon frame, SN=1956, FN=0, Flags=........C, BI=100, SSID=dartes_ASUS	AsustekC_e5:84:60	Broadcast	2	False	802.11
Beacon frame, SN=1957, FN=0, Flags=........C, BI=100, SSID=dartes_ASUS	AsustekC_e5:84:60	Broadcast	1	False	802.11

Figura 5.4: Registo de captura mostrando a influência do STP na *flag* de *multicast*

Os resultados obtidos mostram que o tráfego existente na rede (pacotes *broadcast/multicast* recebidos) pode ter um impacto bastante significativo no consumo médio de corrente dos módulos. Deste modo, cenários baseados na transmissão *broadcast* de dados, poderão obrigar o módulo acorde frequentemente para receber os mesmos (mesmo que não sejam destinados a si), reduzindo desta forma a eficiência do modo de *sleep* e aumentando assim o consumo energético do módulo.

5.2.2 Desativação do *Spanning Tree Protocol* (STP)

O *Spanning Tree Protocol* (STP) é um protocolo que se adequa a redes constituídas por *switches* e permite o cálculo do caminho mais eficiente entre quaisquer estações ligadas a essa rede, possibilitando a determinação de caminhos alternativos, no caso de falha de uma das ligações entre switches. No contexto de um segmento WiFi, este protocolo é inútil e pode ser desativado no AP utilizado através dos seguintes comandos:

```
#conexao ssh com o AP
ssh user@192.168.1.1
nvram set lan_stp=0
nvram set lan1_stp=0
nvram commit
reboot
```

Na secção 5.2.1 verificou-se que o protocolo STP se encontrava ativo no segmento WiFi, levando à disseminação de mensagens *broadcast* e, indiretamente, à mudança do estado de poupança de energia para o modo ativo, ao fazer com que o bit de *multicast* na mensagem DTIM estivesse ativo. Nesta secção, ao desativar o protocolo STP e evitar a propagação das respetivas mensagens *broadcast*, vamos repetir as mesmas experiências e avaliar o impacto no padrão de consumo do módulo. Na ausência completa de tráfego *multicast* ou *broadcast* a disseminar no AP, a transição entre o modo de poupança de energia e o modo ativo será controlada apenas pelo módulo. Nesse cenário, o intervalo máximo de tempo em que uma estação WiFi se pode manter no modo de poupança de energia é acordado no processo de associação com o AP, no parâmetro *Listen interval*,

sendo o mesmo definido de forma automática para um intervalo correspondente à transmissão de 100 *Beacons*. Desta forma, o AP irá manter armazenados pacotes unicast por um período máximo de 100 *Beacons*, mas as estações deverão continuar a ouvir o *Beacon* a cada DTIM *interval*, no caso de pretenderem receber tráfego *multicast* ou *broadcast*.

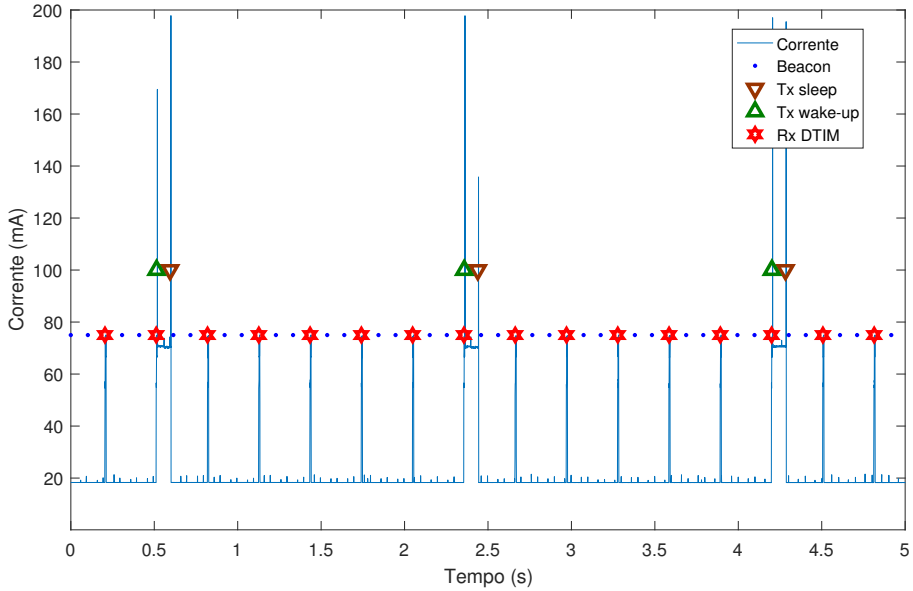


Figura 5.5: Padrão de consumo de corrente instantâneo no modo *Modem-sleep* e com STP desabilitado, anotado com informação da captura de pacotes simultânea (*Beacon interval* = 100 ms e DTIM period = 3)

A figura 5.5 mostra o padrão de consumo instantâneo obtido, para uma configuração do Beacon interval de 100ms e do DTIM interval de 3 *Beacons*, onde é possível observar um padrão de ativação periódico a cada 18 *Beacons* (6 DTIMs). Em conjunto com a amostra do registo da captura de pacotes apresentada na figura 5.6, é possível confirmar que o *bit* de *multicast* do *Bitmap control* se mantém a falso e desta forma não é sinalizada a existência de qualquer *frame broadcast* armazenado no AP. Curiosamente, verifica-se que mesmo na ausência da dessa sinalização, o módulo continua a estabelecer comunicação com o AP, causando a alternância entre os dois modos de funcionamento, ao fim de N mensagens DTIM.

Info	Source	Destination	DTIM count	Multicast	Protocol
Beacon frame, SN=1670, FN=0, Flags=........C, BI=100, SSID=dartes_ASUS	AsustekC_e5:84:60	Broadcast	0	False	802.11
Null function (No data), SN=1, FN=0, Flags=.......TC	Espressi_c0:cf:18	AsustekC_e5:84:60			802.11
Null function (No data), SN=2, FN=0, Flags=...P...TC	Espressi_c0:cf:18	AsustekC_e5:84:60			802.11
Beacon frame, SN=1671, FN=0, Flags=........C, BI=100, SSID=dartes_ASUS	AsustekC_e5:84:60	Broadcast	2	False	802.11
Beacon frame, SN=1672, FN=0, Flags=........C, BI=100, SSID=dartes_ASUS	AsustekC_e5:84:60	Broadcast	1	False	802.11
Beacon frame, SN=1673, FN=0, Flags=........C, BI=100, SSID=dartes_ASUS	AsustekC_e5:84:60	Broadcast	0	False	802.11
Beacon frame, SN=1674, FN=0, Flags=........C, BI=100, SSID=dartes_ASUS	AsustekC_e5:84:60	Broadcast	2	False	802.11
Beacon frame, SN=1675, FN=0, Flags=........C, BI=100, SSID=dartes_ASUS	AsustekC_e5:84:60	Broadcast	1	False	802.11
Beacon frame, SN=1676, FN=0, Flags=........C, BI=100, SSID=dartes_ASUS	AsustekC_e5:84:60	Broadcast	0	False	802.11
Beacon frame, SN=1677, FN=0, Flags=........C, BI=100, SSID=dartes_ASUS	AsustekC_e5:84:60	Broadcast	2	False	802.11
Beacon frame, SN=1678, FN=0, Flags=........C, BI=100, SSID=dartes_ASUS	AsustekC_e5:84:60	Broadcast	1	False	802.11
Beacon frame, SN=1680, FN=0, Flags=........C, BI=100, SSID=dartes_ASUS	AsustekC_e5:84:60	Broadcast	0	False	802.11
Beacon frame, SN=1682, FN=0, Flags=........C, BI=100, SSID=dartes_ASUS	AsustekC_e5:84:60	Broadcast	2	False	802.11
Beacon frame, SN=1683, FN=0, Flags=........C, BI=100, SSID=dartes_ASUS	AsustekC_e5:84:60	Broadcast	1	False	802.11
Beacon frame, SN=1684, FN=0, Flags=........C, BI=100, SSID=dartes_ASUS	AsustekC_e5:84:60	Broadcast	0	False	802.11
Beacon frame, SN=1685, FN=0, Flags=........C, BI=100, SSID=dartes_ASUS	AsustekC_e5:84:60	Broadcast	2	False	802.11
Beacon frame, SN=1686, FN=0, Flags=........C, BI=100, SSID=dartes_ASUS	AsustekC_e5:84:60	Broadcast	1	False	802.11
Beacon frame, SN=1687, FN=0, Flags=........C, BI=100, SSID=dartes_ASUS	AsustekC_e5:84:60	Broadcast	0	False	802.11
Beacon frame, SN=1688, FN=0, Flags=........C, BI=100, SSID=dartes_ASUS	AsustekC_e5:84:60	Broadcast	2	False	802.11
Beacon frame, SN=1689, FN=0, Flags=........C, BI=100, SSID=dartes_ASUS	AsustekC_e5:84:60	Broadcast	1	False	802.11
Beacon frame, SN=1690, FN=0, Flags=........C, BI=100, SSID=dartes_ASUS	AsustekC_e5:84:60	Broadcast	0	False	802.11
Null function (No data), SN=3, FN=0, Flags=.......TC	Espressi_c0:cf:18	AsustekC_e5:84:60			802.11
Null function (No data), SN=4, FN=0, Flags=...P...TC	Espressi_c0:cf:18	AsustekC_e5:84:60			802.11
Beacon frame, SN=1691, FN=0, Flags=........C, BI=100, SSID=dartes_ASUS	AsustekC_e5:84:60	Broadcast	2	False	802.11
Beacon frame, SN=1692, FN=0, Flags=........C, BI=100, SSID=dartes_ASUS	AsustekC_e5:84:60	Broadcast	1	False	802.11

Figura 5.6: Amostra da captura de pacotes simultânea à medição do consumo de energia com o STP desativado (*Beacon interval* = 100 ms e DTIM period = 3)

5.3 Impacto de diferentes configurações para o *Beacon* e DTIM *interval*

Como visto nas secções anteriores, a definição de diferentes períodos para a difusão dos *Beacons* (*Beacon interval*) e do DTIM (*DTIM interval*) pelo AP implica diferentes repercussões no consumo energético das respetivas estações associadas. A presente secção analisa qual o impacto dessas configurações no consumo de corrente, no caso particular dos módulos WiFi ESP8266.

5.3.1 Metodologia e cenários considerados

Como discutido, em dois dos quatro modos de funcionamento do módulo (*Modem-sleep* e *Light-sleep*), o período de tempo que o módulo se mantém num estado de baixo consumo energético depende essencialmente do valor dos parâmetros *Beacon Period* e DTIM *Interval*. Desta forma, para cada uma das diferentes configurações, foram realizadas medições do consumo de corrente durante um intervalo de tempo correspondente a 100s para os dois modos de funcionamento mencionados. Em particular, para cada um dos modos de operação, com e sem a habilitação do STP no AP, foram considerados o cenário onde o DTIM *interval* é fixado no valor 3 e é feito variar o *Beacon interval* (Secção 5.3.1.1) e o cenário onde o *Beacon interval* é fixado em 100ms e é feito variar o DTIM *interval* (Secção 5.3.1.2). Os valores "Teóricos"apresentados em ambos os casos correspondem aos modelos apresentados nas Equações 4.2 e 4.3.

5.3.1.1 Variação do *Beacon interval* com DTIM *interval* fixo

A figura 5.7 apresenta a evolução da corrente média consumida pelo módulo com a variação do *Beacon interval* de 100 a 1000 ms, em passos de 100 ms, mantendo o DTIM *interval* configurado a 3. Os resultados apresentados mostram que a configuração do módulo para operar no modo de *Modem-sleep*, permite que o mesmo beneficie de uma redução no consumo de corrente quando o

AP difunde os *Beacons* com períodos superiores, ao possibilitar que a interface WiFi se mantenha desligada por períodos mais longos de tempo. Foi ainda possível observar, com a desabilitação do STP, uma vantagem para o consumo energético com um *offset* negativo médio de 2.06mA ao longo de todas as medições no modo de *Modem-sleep*. Em resumo, foi possível obter um consumo de corrente mínimo de 20.35mA (com STP) e de 19.01mA (sem STP), na configuração com a difusão do *Beacon* a cada 1000ms, e um consumo máximo de 25.34mA (com STP) e de 22.29mA (sem STP), na configuração com a difusão do *Beacon* a cada 100ms.

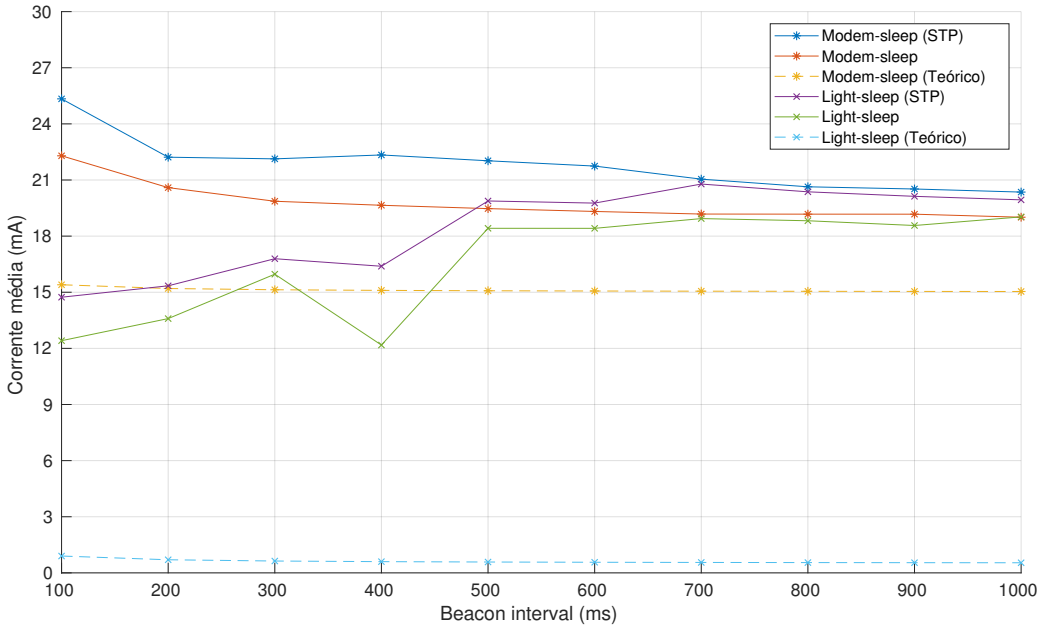
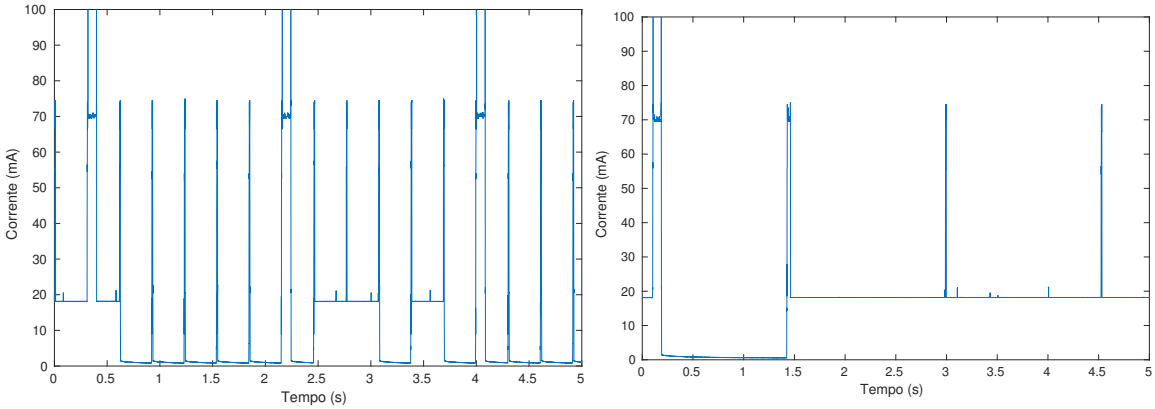


Figura 5.7: Consumo de corrente médio com *Beacon period*=100-1000ms e DTIM *interval*=3

Por outro lado, o modo de *Light-sleep* mostrou um comportamento distinto, com um aumento da corrente média consumida, quando o AP se encontrava configurado para difundir o *Beacon* com períodos superiores, mantendo-se um *offset* negativo médio de 1.77mA quando o STP foi desativado. A figura 5.9 mostra que para configurações do *Beacon interval* superiores a 700 ms o consumo de corrente converge para o mesmo observado no modo de Modem-sleep. Este efeito, já referido na Secção 4.3.1.2, pode ser explicado pela existência de tarefas do sistema operativo (FreeRTOS), com um período de ativação inferior ao fator $BeaconInterval \times DTIMInterval$, que necessitam, por isso, de ser ativadas durante o período de tempo em que o módulo poderia potencialmente estar com o CPU desligado (com o modo *Light-sleep* ativo). Desta forma para períodos de inatividade longos, mesmo que o módulo seja configurado para ativar o modo *Light-sleep*, o mesmo é ignorado e é feita a comutação automática da configuração para o modo de Modem-sleep (permanecendo desta forma o CPU ligado). Assim foi possível obter um consumo de corrente mínimo de 14.73mA (com STP) e de 12.18 (sem STP), na configuração com a difusão do *Beacon* a cada 100ms e a cada 400ms, respetivamente. Por outro lado, foi observado um consumo máximo de 20.78mA (com STP) e de 19.03mA (sem STP), na configuração com a difusão do *Beacon* a cada 700ms e a cada 1000ms, respetivamente.

A figura 5.8 mostra gráficos do consumo instantâneo de corrente no modo de *Light-sleep* para duas configurações distintas do *Beacon interval*: 100 ms (esquerda) e 500 ms (direita). Ambos os gráficos mostram os picos de consumo de corrente típicos relativos à receção do *Beacon* com a mensagem de DTIM. Apesar disso, verifica-se que em algumas situações após esse procedimento o módulo não retoma o modo de *Light-sleep*, fixando-se num patamar de consumo de corrente de 18.2mA. Este de patamar consumo é compatível com o observado quando o modo de *Modem-sleep* é ativado, e corresponde, por isso, ao desligar das interfaces de comunicação WiFi e não do CPU. Por outro lado, podemos verificar que por vezes o patamar de consumo no modo de poupança de energia é de cerca de 0.85mA (95.3% de redução no consumo de corrente), correspondendo à ativação efetiva do modo de *Light-sleep*, com o desligar das interfaces de comunicação WiFi e de suspensão do próprio CPU. Desta forma, apesar do incremento do fator $Beaconinterval \times DTIMinterval$ significar que o módulo pode permanecer mais tempo no modo de poupança de energia, incrementa também a probabilidade de existirem tarefas do próprio sistema operativo que devem ser ativadas durante o período em que o CPU estaria desligado. Assim, verifica-se para configurações do fator $Beaconinterval \times DTIMinterval$ mais baixas, um consumo de corrente médio inferior, no modo de *Light-sleep*, pois o módulo é capaz de permanecer efetivamente mais tempo nesse mesmo modo.



(a) Beacon interval=100ms

(b) Beacon interval=500ms

Figura 5.8: Padrão de consumo de corrente instantâneo no modo *Light-sleep* para diferentes configurações do *Beacon interval*.

5.3.1.2 Variação do DTIM *interval* com *Beacon interval* fixo

A figura 5.9 apresenta a evolução da corrente média consumida pelo módulo com a variação do DTIM *interval* de 1 a 10, em passos de 1, mantendo o *Beacon interval* configurado a 100ms. Os resultados obtidos com esta experiência vão de encontro aos apresentados na Secção 5.3.1.1, com a configuração de funcionamento no modo de *Modem-sleep* a beneficiar o consumo de corrente para configurações onde a mensagem DTIM é difundida no *Beacon* com períodos superiores. É possível ainda verificar que a desabilitação do STP, permite uma vantagem para o consumo energético, com um *offset* negativo médio de 1.59mA ao longo de todas as medições no modo de

Modem-sleep. Em resumo, foi possível obter um consumo de corrente mínimo de 21.7mA (com STP) e de 21.1mA (sem STP), na configuração do DTIM *interval* para 10, e um consumo máximo de de 28.49mA (com STP) e de 24.82mA (sem STP), na configuração do DTIM *interval* para 1.

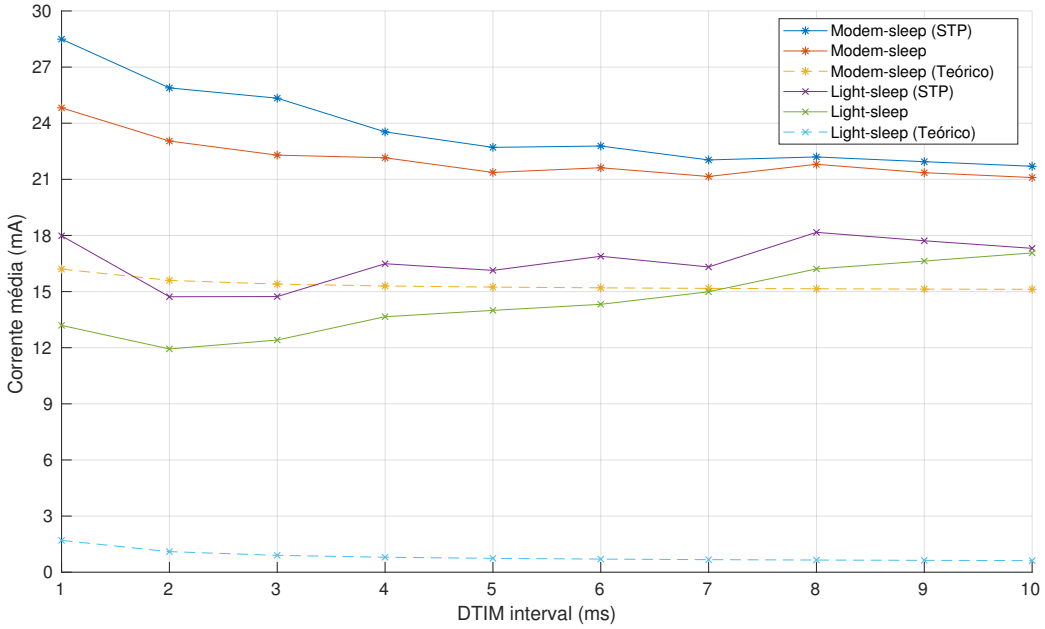


Figura 5.9: Consumo de corrente médio com *Beacon period*=100ms e DTIM *interval*=1-10

Já na configuração com o modo de *Light-sleep* ativado, obteve-se um *offset* negativo de 2.21mA com a desativação do STP. Foi ainda possível observar um comportamento semelhante ao apresentado na Secção 5.3.1.1, com o consumo observado a aproximar-se do obtido com a configuração de operação no modo de *Modem-sleep*, para valores do DTIM *interval* mais elevados. Em resumo, foi possível obter um consumo de corrente mínimo de 14.72mA (com STP) e de 11.93mA (sem STP), na configuração do DTIM *interval* para 2, e um consumo máximo de de 18.17mA (com STP) e de 17.07mA (sem STP), na configuração do DTIM *interval* para 8 e 10, respetivamente.

5.3.2 Modelo de consumo melhorado para o modo de *Modem-sleep*

As equações (4.2 e 4.3) que modelam o consumo de corrente em cada um dos modos de funcionamento, consideram um patamar de consumo, referente a cada um dos modos de operação, correspondente a 15mA para o modo de *Modem-sleep* e de 0.5mA para o modo de *Light-sleep*. Complementarmente, partem do pressuposto que quando num estado de inatividade (sem envio ou receção de dados), o módulo permanece no modo de poupança de energia por um período de tempo correspondente ao fator $BeaconInterval \times DTIMPeriod$, voltando ao estado ativo (CPU e interface WiFi são ligados), apenas por uma fração de tempo de 2ms para receber e processar um *Beacon* de forma a identificar a existência de tráfego *unicast* ou *broadcast* armazenado no AP. Por outro lado, os resultados experimentais obtidos mostram que o consumo no modo de *Modem-sleep* corresponde a 18.2mA (21% mais que o consumo anunciado) e a 0.85mA (70% mais que

o consumo anunciado) no modo de *Light-sleep*. Verifica-se ainda o despertar do módulo a cada *Beacon* com a mensagem DTIM (fator $BeaconInterval \times DTIMPeriod$), mas com um consumo de corrente médio de 61.4mA por um período de 8ms.

Adicionalmente, é ainda possível verificar a presença de fatores não considerados no modelo teórico. Constata-se que o despertar do módulo nem sempre é realizado por um curto intervalo de tempo (8ms), verificando-se que, a cada N mensagens DTIM, o módulo não acorda apenas para ouvir a mesma, mas realiza comunicações com o AP. Esse padrão de comunicação, assemelha-se ao observado quando existe tráfego armazenado e o módulo pretende sinalizar, ao AP, a mudança do estado de poupança de energia para o modo ativo, tendo-se observado um consumo médio de 71.4mA, durante 90ms, mesmo na ausência de quaisquer pacotes armazenados no AP (configuração com o STP desativado). A frequência desse procedimento mostrou ter uma variação não linear e dependente da configuração Beacon e do DTIM interval. A análise dos registos de captura da interface de rede WiFi (modo monitor), permitiu fazer a análise do parâmetro N com a variação do fator $BeaconInterval \times DTIMPeriod$ de 100 a 1000ms.

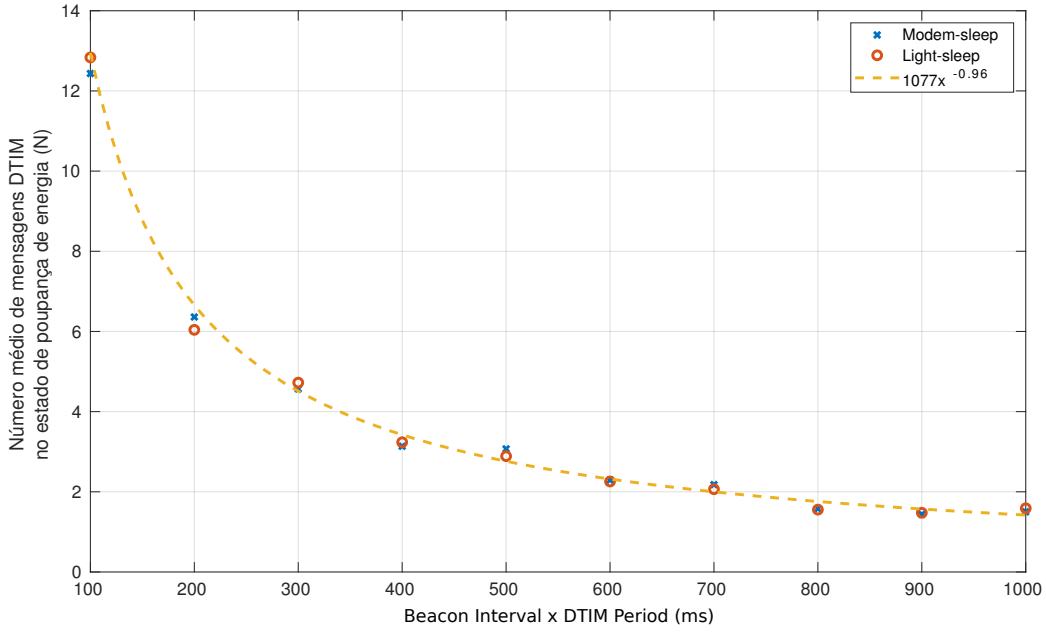


Figura 5.10: Número médio de mensagens DTIM no estado de poupança de energia

A figura 5.11 apresenta os valores médios de N, observados para cada uma das configurações, mostrando que esse parâmetro pode ser modelado pela seguinte equação:

$$N(x) = 1077x^{-0.96}, 100 \leq x \leq 1000 \tag{5.1}$$

Considerando que a cada $N(x)$ mensagens DTIM é estabelecida comunicação com o AP, a fração do número total de DTIM em que o módulo faz transmissões pode ser dada pela seguinte

equação:

$$F_1(x) = \frac{1}{N(x)} \tag{5.2}$$

Da mesma forma, a fração de mensagens DTIM que o módulo não faz transmissões pode ser dada pela seguinte equação:

$$F_2(x) = 1 - \frac{1}{N(x)} = \frac{N(x)-1}{N(x)} \tag{5.3}$$

Assim, a equação 5.4 modela o consumo médio considerando os dois cenários, onde x pode ser dado pelo fator $DTIMInterval \times BeaconInterval$, t_ativo_transm e t_ativo_ouvir representam o intervalo de tempo em que o módulo se mantêm ativo para fazer transmissões (90ms) e para ouvir apenas o *Beacon* (8ms), respetivamente e $I\_inativo$ (18.2mA), I_ativo_transm (71.4mA) e I_ativo_ouvir (61.4mA) são as correntes consumidas nos modos de poupança de energia, ativo para fazer transmissões e ativo para ouvir o *Beacon*, respetivamente.

$$I_{modem-sleep(1)}(x) = I_{inativo} + F_1(x) \times \frac{t_{ativo\_transm.}}{x} \times I_{ativo\_transm.} + F_2(x) \times \frac{t_{ativo\_ouvir}}{x} \times I_{ativo\_ouvir} \tag{5.4}$$

Desta forma, o gráfico da figura 5.11 mostra os resultados experimentais obtidos para uma variação do fator $BeaconInterval \times DTIMInterval$ entre 100ms e 1000ms, em comparação com o modelo de consumo melhorado (1) e o teórico inicial (2).

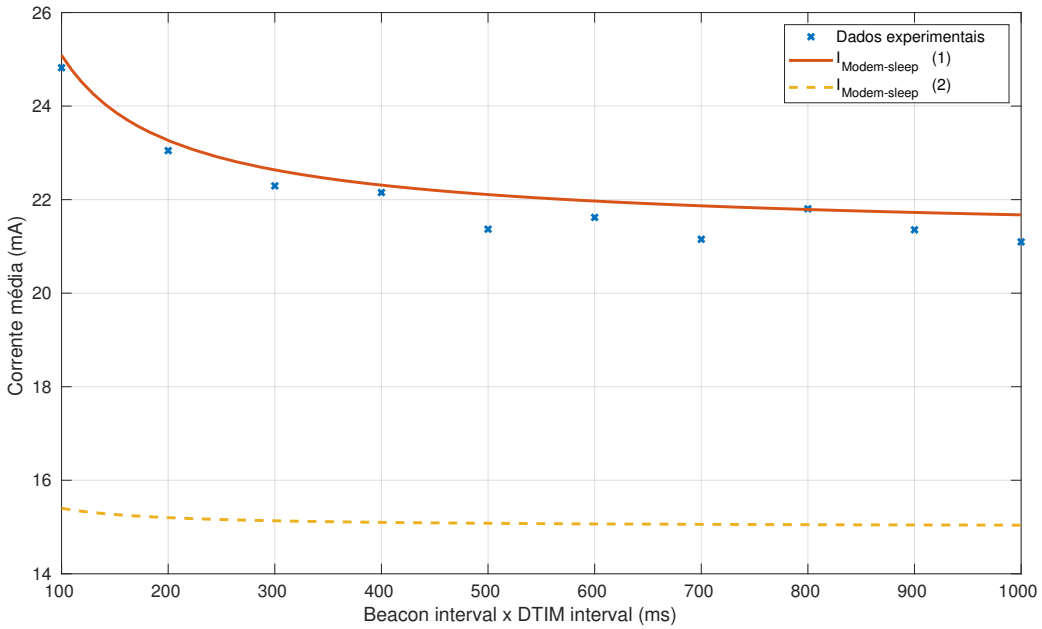


Figura 5.11: Consumo de corrente médio com *Beacon period*=100ms e DTIM *interval*=1-10

5.4 Conectividade no meio interior

Nesta secção abordamos a qualidade e fiabilidade da conectividade entre o módulo e o respetivo AP, num cenário onde se pretende utilizar a antena integrada do mesmo nas comunicações num ambiente *indoor*. Para tal analisou-se o o *Received Signal Strength Indicator* (RSSI) de pacotes enviados pelo módulo a diferentes distâncias, bem como os limites de conectividade do mesmo, medindo o *Packet Delivery Ratio* (PDR) e *Round-trip delay* (RTD).

5.4.1 Medições de RSSI de transmissões do módulo

Sendo o RSSI frequentemente utilizado como um dos parâmetros que permite avaliar a qualidade de determinada conexão WiFi, nesta experiência o módulo foi programado para enviar, com uma periodicidade de 1s, pacotes UDP de 85B de tamanho para um computador recetor passivo (em modo de monitor), em linha de vista a diferentes distâncias (de 1m a 8m). Tendo em conta o padrão de radiação não uniforme da antena integrada do módulo, seria expectável que diferentes posições da antena resultassem em diferentes valores para o RSSI. Desta forma, para cada uma das distâncias, foi medido o valor do RSSI para 4 posições diferentes do módulo em relação ao recetor colocado em linha de vista com este, tal como apresentado na figura 5.12.

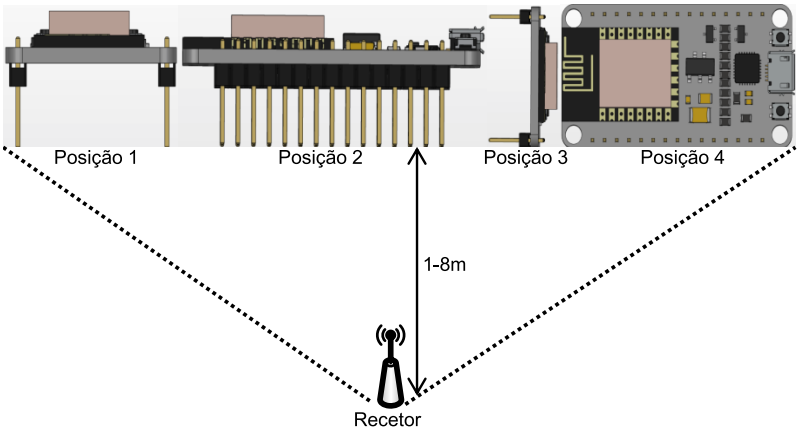


Figura 5.12: Diferentes posições da antena embebida do módulo em relação ao recetor passivo

Para cada distância e orientação foram recolhidas 60 medições de RSSI, estando o valor médio para cada uma das posições representado no gráfico da figura 5.13. Para distâncias superiores a 4m, podemos observar um incremento da variabilidade dos valores obtidos para o RSSI, provocado pela existência de reflexões e de outro tipo de influências, típicas de cenários de comunicação no interior de edifícios. Foi ainda possível mostrar que a orientação relativa do módulo em relação ao recetor influência, de facto, o valor obtido para o RSSI, mas que essa influência depende do material e paredes envolventes sendo por isso alterada com a transmissão a diferentes distâncias. Ainda assim os valores obtidos vão de encontro aos valores típicos [42] de uma receção WiFi com boa qualidade, variando de uma média global de -52.9dBm para a posição 2 a -57.3dBm, -57.3dBm e 58.4dBm para as posições 1, 3 e 4, respetivamente.

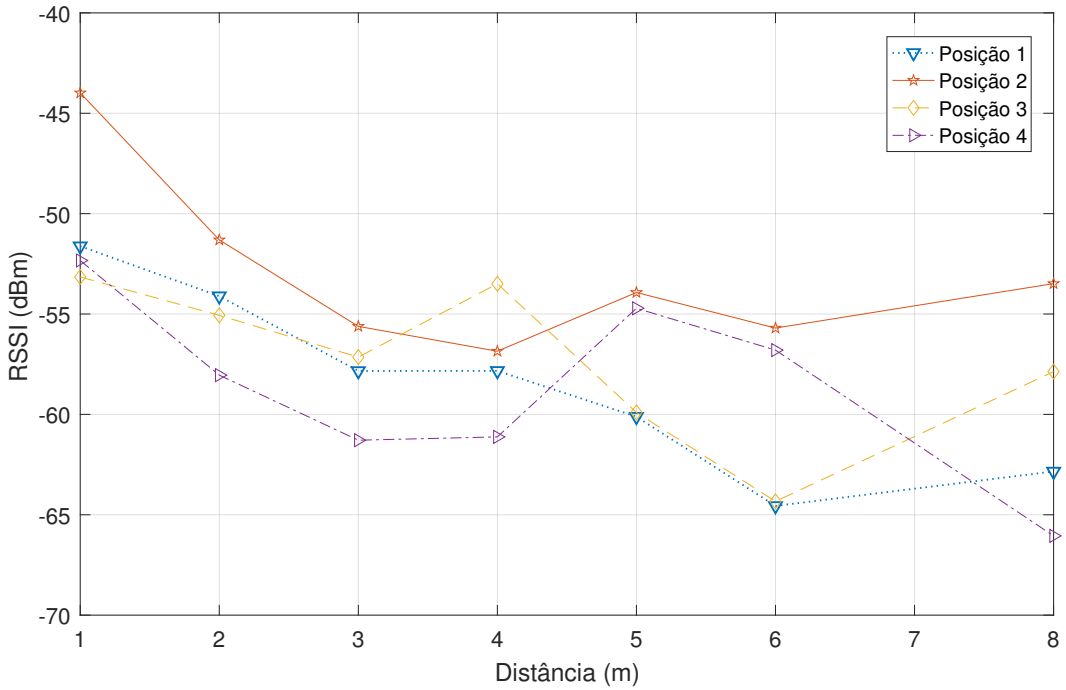


Figura 5.13: Valores de RSSI médios obtidos para diferentes orientações e distâncias de um recetor

5.4.2 Limites de conectividade

De forma a avaliar o limite prático de operação destes módulos num ambiente de comunicação no interior de edifícios, mediu-se o *Packet Delivery Ratio* (PDR) e o *Round-trip delay* (RTD) em diferentes localizações de um edifício. A figura 5.14 apresenta as diferentes localizações consideradas, encontrando-se algumas em linha de vista com o recetor e outras com paredes e outros obstáculos entre os mesmos. Estas experiências foram levadas a cabo utilizando o comando *ping* com a transmissão de 100 pacotes, com um *payload* de 85B, para cada um das posições em análise. O referido pedido de ping teve origem no computador conectado ao AP através de uma ligação Ethernet, tendo como destino o módulo ESP8266 conectado ao mesmo AP através de uma ligação WiFi. Esta abordagem pretendeu garantir que as únicas transmissões no segmento WiFi (rede BSS) deste AP eram o pedido e a resposta do comando ping.

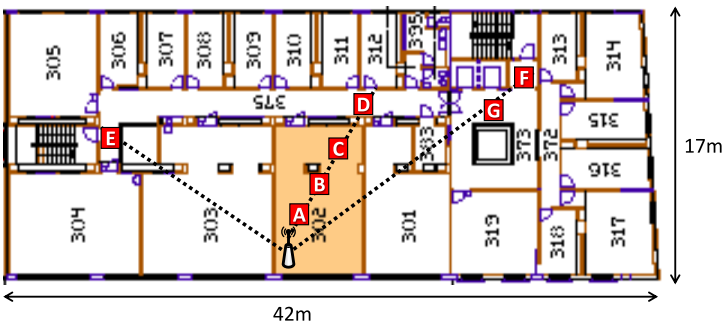


Figura 5.14: Posições do módulo ESP8266 no edifício onde os limites de conectividade foram testados

A figura 5.15 mostra a distribuição das medições do RTD, com as letras A, B, C, D, E, F e G a corresponderam às posições marcadas na figura 5.14. A posição G.2 corresponde à mesma posição que G, mas no andar inferior. Como esperado, as medições em A, B, C e D apresentam resultados semelhantes com uma média (e mediana) de 6.8 (4.07), 11.54 (4.10), 9.01 (4.06) e 9.76 (4.34) ms, respetivamente. Por outro lado, à medida que a distância do AP e o consequente número de paredes a atravessar aumenta, assistimos a um incremento do número de retransmissões WiFi, uma vez que o RTD aumenta consideravelmente. Nas posições E, F, G e G.2 obtemos uma média (e mediana) para o RTD de 11.87 (4.8), 14.08 (5.36), 15.95 (6.00) e 30.87 (15.3) ms. Adicionalmente, com as configurações padrão do módulo, obtivemos um PDR de 100% para as posições A, B, C, D, E e F, uma vez que todas as perdas foram recuperadas com os mecanismos de retransmissão do WiFi. Pelo contrário, tal não foi possível de observar nas posições G e G.2, com um PDR de 99% e 79%, respetivamente.

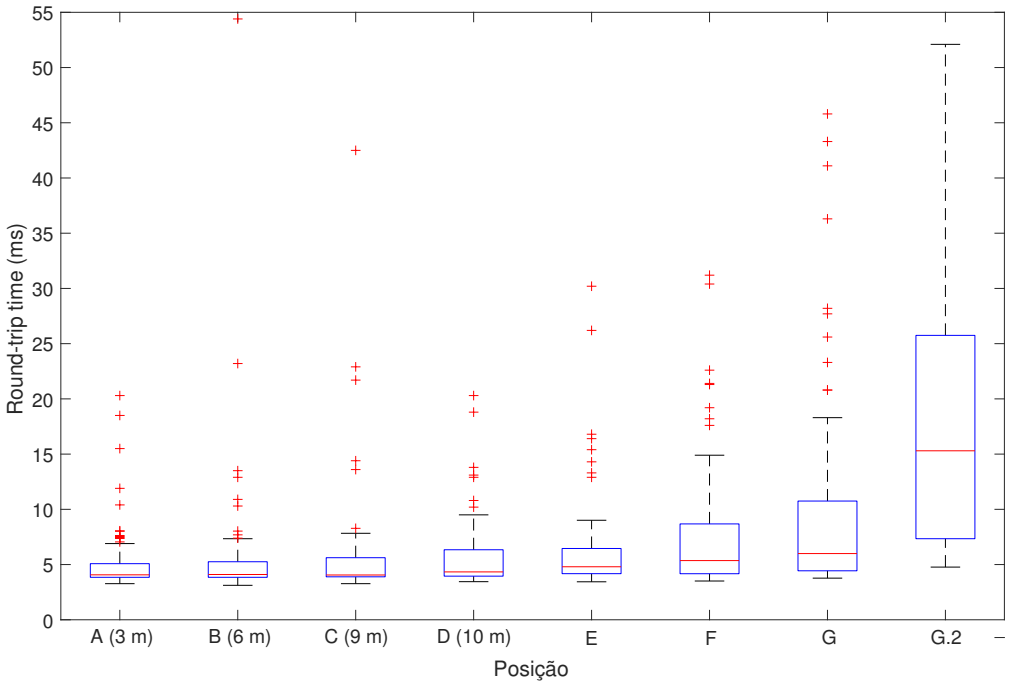


Figura 5.15: Distribuição das medições do RTD

5.5 Sumário

Este capítulo apresentou o modo de funcionamento das interações automáticas do módulo WiFi ESP8266 com a infraestrutura WiFi e de como a configuração de parâmetros como o *Beacon interval* e o DTIM *period* afetam o consumo de energia e eficácia dos diferentes modos de poupança de energia. Ao contrário do esperado, os resultados obtidos mostram que o incremento desses fatores nem sempre beneficia o consumo energético, com o modo de *Light-sleep* a mostrar perda de eficácia com configurações superiores desses parâmetros. Por último, foi possível

verificar que o módulo consegue fornecer garantias de uma boa conectividade em cenários de comunicação no interior de edifícios, mostrando PDRs superiores ou iguais 99% no mesmo piso.

Capítulo 6

Comunicação M2M utilizando o módulo ESP8266

Como referido no capítulo 2, a monitorização de pacientes, em ambiente hospitalar, pode ser realizado com a utilização de *wearables* (*smartwatches* ou *wristbands*) que recolhem e enviam para um servidor central, os sinais vitais e informação fisiológica de cada paciente. Esses dados podem incluir parâmetros como a frequência e variabilidade do batimento cardíaco, a pressão sanguínea, o nível de oxigénio no sangue ou a temperatura corporal. Apesar disso, é comum que estes dispositivos apenas suportem protocolos de comunicação de baixo alcance, tipicamente o Bluetooth, necessitando por isso da utilização de *gateways* (GWs) para estabelecer ligação à Internet. Desta forma, pretende-se estudar a possibilidade de construir *wearables*, fornecidos aos pacientes à chegada ao serviço de urgências ou durante o período de internamento, o módulo WiFi ESP8266 permitindo desta forma uma conexão direta do dispositivo à infraestrutura WiFi do hospital. Em particular, este capítulo pretende abordar os aspetos referentes à implementação dos componentes de software necessários para a integração do referido módulo, numa *framework* [43] que pretende habilitar o seguimento em continuo de pacientes em ambiente hospitalar, baseada no standard oneM2M. Este standard suporta dois modelos distintos de comunicação, são eles *publish-subscribe* e *request-response*. Suporta, ainda, a utilização de protocolos de comunicação como MQTT, COAP ou HTTP. Esta implementação focou-se na utilização do protocolo MQTT e no modelo *publish-subscribe*, por ser considerado mais orientado a aplicações de sensorização e monitorização remota, permitindo uma maior eficiência das comunicações em aplicações deste tipo. Assim, considerámos um cenário onde o módulo WiFi publica os dados que recolhe do paciente e uma outra entidade, como por exemplo um sistema de *Electronic Health Record* (EHR), subscreve essa mesma informação e associa a mesma aos registos do paciente. Ao longo deste capitulo é apresentada a arquitetura do sistema e os respetivos componentes de software desenvolvidos.

6.1 Arquitetura

Considerando um sistema baseado no standard oneM2M, o referido dispositivo de monitorização e o respetivo módulo WiFi são *Application Dedicated Nodes* (ADN). Um ADN é um nodo do sistema, no domínio da aplicação (field domain), que contém pelo menos uma *Application Entity* (AE), mas não possui nenhuma *Common Service Entity* (CSE). Tal como definido no standard, o AE é a entidade da camada de aplicação que implementa a lógica dos serviços disponibilizados pela aplicação M2M, e o CSE representa a instanciação de um conjunto comum de serviços, disponíveis a outras entidades através dos pontos de referência (Mca e Mcc). No cenário considerado o AE no ADN comunica, utilizando o ponto de referencia Mca, com o CSE que reside num *Infrastructure Node* no domínio da infraestrutura (*Infrastructure domain*). No padrão oneM2M, a informação é representada utilizando um conceito de recursos, seguindo a arquitetura RESTful, sendo as propriedades desses recursos mutáveis ao longo do tempo e endereçáveis por um único endereço usando um *Universal Resource Identifier* (URI). Esses recursos são armazenados no IN-CSE, que hospeda os mesmos utilizando uma estrutura hierárquica em árvore. Da mesma forma, as subscrições devem ser operadas e representadas como recursos na referida árvore.

O ADN-AE, ou seja, a AE no ADN, é responsável pelo sensorização e recolha de informação. No cenário considerado, toda a lógica do ADN-AE é executada no módulo ESP8266. Podendo o mesmo ser configurado para enviar informação com diferentes períodos de transmissão e para ajustar o modo de funcionamento (*Modem-sleep*, *Light-sleep* ou *Deep-sleep*) consoante a autonomia disponível ou outro tipo de restrições impostas.

Toda a lógica do IN-CSE é assente na plataforma *open source* Eclipse OM2M, que suporta protocolos de comunicação como CoAP, MQTT ou HTTP. No cenário considerado, o ADN-AE publica a informação recolhida no IN-CSE, utilizando o protocolo MQTT, por intermédio de um servidor MQTT que atua como broker das comunicações entre as duas entidades. O servidor MQTT utilizado foi o *Eclipse Mosquitto*, tendo o mesmo sido instalado e configurado na mesma máquina onde o IN-CSE era executado. O software implementado no ADN-AE utilizou a biblioteca *Eclipse Paho* para a comunicação MQTT com o *broker* e a biblioteca cJSON para a serialização dos *payloads* das mensagens transmitidas. Por seu lado, o IN-AE, ou seja, uma AE registada no CSE como IN, subscreve no IN-CSE os recursos criados pelo ADN-AE, recebendo uma notificação sempre que é publicada nova informação.

A figura 6.1 apresenta um esquemático que resume a arquitetura do sistema previamente descrito.

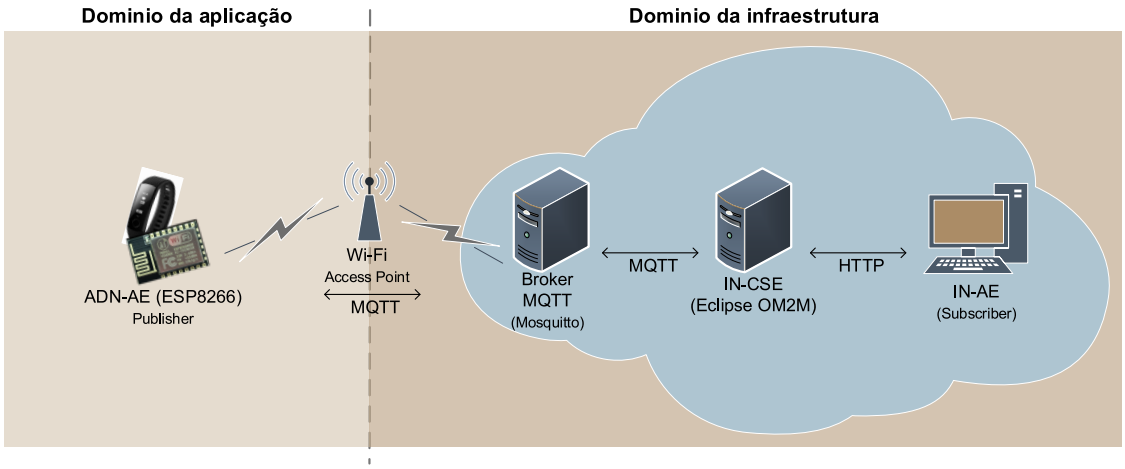


Figura 6.1: Arquitetura do sistema utilizando entidades oneM2M *standard*

6.2 Software no ADN-AE

6.2.1 Envio de pedidos ao IN-CSE

Um pedido oneM2M tem de ser constituído por um conjunto obrigatório de parâmetros, que devem ser incluídos no *payload* da mensagem MQTT correspondente. Em particular, deve conter o seguinte conjunto de parâmetros obrigatórios: To (to), que indica o URI do destinatário da mensagem; From (fr) que indica o ID do remetente da mensagem; e Operation (op), que indica o tipo de operação a que se destina o pedido. Este último pode ser de um dos seguintes tipos: *Create* (1), *Retrieve* (2), *Update* (3), *Delete* (4), *Notify* (5) ou *Discovery* (6).

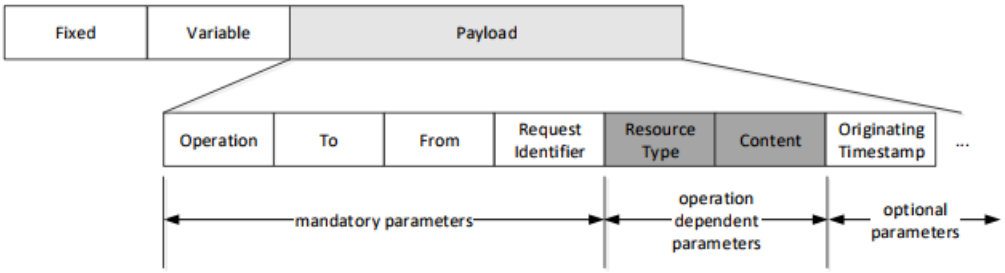


Figura 6.2: Estrutura de um pedido oneM2M utilizando MQTT [11]

Como mostra a figura 6.2, para além dos parâmetros obrigatórios referidos anteriormente, um pedido pode ter um conjunto de parâmetros adicionais que dependem do tipo de operação que se pretende realizar.

Para habilitar a comunicação e a publicação de dados do ADN-AE no IN-CSE, foram criadas funções genéricas que permitem a criação do *payload*, no formato JSON, de um conjunto de operações. Esse conjunto inclui a geração do *payload* para criação do registo do ADN-AE no IN-CSE, a criação de *containers* onde a informação será guardada e a criação de instâncias de

conteúdo num *container* já existente. Apresenta-se de seguida exemplos de *payloads* resultantes da execução de cada umas das funções enumeradas:

- Criação (m2m:op) em "/in-cse/dartes"(m2m:to) de uma nova AE (m2m:ty) com o nome "ESP_PUB"(rn)

```
{
"m2m:rqp":{
    "m2m:fr":"admin:admin",
    "m2m:to":"/in-cse/dartes",
    "m2m:op":1,
    "m2m:rqi":123456,
    "m2m:pc":{
        "m2m:ae":{
            "rr":"true",
            "poa":[
            "mqtt://192.168.1.119:1883/oneM2M/req/+/ESP_PUB/json"
            ],
            "api":12345,
            "rn":"ESP_PUB"
        }
    },
    "m2m:ty":2
}
}
```

- Criação (m2m:op) em "/in-cse/dartes/ESP_PUB"(m2m:to) de um novo container (m2m:ty) com o nome "HR"(rn)

```
{
"m2m:rqp":{
    "m2m:fr":"admin:admin",
    "m2m:to":"/in-cse/dartes/ESP_PUB",
    "m2m:op":1,
    "m2m:rqi":12334,
    "m2m:pc":{
        "m2m:cnt":{
            "rn":"HR"
        }
    },
    "m2m:ty":3
}
```

```
}
```

- Criação (m2m:op) em "/in-cse/dartes/ESP_PUB/HR"(m2m:to) de uma nova instância de conteúdo (m2m:ty) com o nome "00000163fba878b4"(rn)

```
{
"m2m:rqp":{
    "m2m:fr":"admin:admin",
    "m2m:to":"/in-cse/dartes/ESP_PUB/HR",
    "m2m:op":1,
    "m2m:rqi":12345,
    "m2m:pc":{
        "m2m:cin":{
            "con":"{\"average1\":602569874}",
            "cnf":"application/json",
            "rn":"00000163fba878b4"
        }
    },
    "m2m:ty":4
}
}
```

6.2.2 Tópicos MQTT para envio e receção de pedidos

O standard oneM2M prevê que a interação entre as AE e as CSE, seja realizada através de um paradigma de comunicação do tipo *request-response*. Pelo contrário, o protocolo MQTT, utilizado para estabelecer a comunicação entre a ADN-AE e a IN-CSE, utiliza um modelo de comunicação distinto (*publish-subscribe*). Assim, de forma a compatibilizar a integração de ambas as tecnologias foi necessário ter em consideração algumas premissas. Nomeadamente, de maneira a que os pedidos de realização de operações (como criação, atualização ou eliminação de recursos), por parte da ADN-AE, sejam corretamente interpretados pela IN-CSE, os mesmos devem ser realizados enviado um pacote do tipo MQTT PUBLISH para um tópico com uma estrutura pré-definida. Este tópico identifica a origem, destinatário e o formato de serialização do *payload* da mensagem, da seguinte forma:

- /oneM2M/req/<originator>/<target-id>/<serialization-format>
  - "oneM2M": identifica que o tópico é utilizado por entidades oneM2M.
  - "req": identifica que o tópico é relativo a um pedido.
  - <originator>: identifica o ID da AE que envia o pedido.
  - <target-id>: identifica o ID da CSE destinatária do pedido.

  - <serialization-format>: identifica o tipo de serialização utilizado no pedido (JSON ou XML).

Da mesma forma, o ADN-AE deve esperar receber a resposta, da IN-CSE, aos pedidos enviados, para um tópico com a seguinte estrutura:

- /oneM2M/resp/<target-id>/<originator>/<serialization-format>

  - "oneM2M": identifica que o tópico é utilizado por entidades oneM2M.
  - "resp": identifica que o tópico é relativo à resposta a um pedido.
  - <target-id>: identifica o ID da CSE que responde ao pedido.
  - <originator>: identifica o ID da AE que originou o pedido.
  - <serialization-format>: identifica o tipo de serialização utilizado no pedido (JSON ou XML).

6.2.3 Troca de mensagens

A troca de mensagens entre as duas entidades, deve ser precedida de um processo de inicialização, onde ambas estabelecem conexão ao servidor MQTT e subscrevem aos tópicos referidos anteriormente. A figura 6.3 apresenta um diagrama de sequências que mostra os procedimentos realizados no processo de inicialização, considerando um ADN-AE com um ID "ESP_SUB"que aceita o formato JSON no *payload* das mensagens de resposta que recebe; e um IN-CSE com um ID "in-cse"que subscreve, utilizando o caráter '+' (*wildcard*), todos os pedidos a si destinados.

Após a conclusão do processo de inicialização, ambas as entidades estão habilitadas a iniciar a troca de mensagens. Tal como apresentado na figura 6.4, esse processo começa com o envio de uma publicação para efetuar o registo da AE no IN-CSE. Essa publicação é recebida pelo servidor MQTT, que remete a mesma para o IN-CSE. Após a validação do pedido e respetiva criação do recurso, o IN-CSE envia uma mensagem de confirmação de execução do mesmo. Essa mensagem é enviada para o tópico de resposta definido e é remetida pelo servidor MQTT para o ADN-AE. Seguindo o mesmo procedimento, sucede-se a criação do container, onde os valores referentes as medições podem ser armazenados. Após isso o IN-AE, regista a AE no IN-CSE e subscreve o container onde as medições enviadas pelo ADN-AE são mantidas.

Por último, o diagrama de sequências da figura 6.5 mostra o processo de criação da instância de conteúdo pelo ADN-AE no IN-CSE e a respetiva notificação dessa criação ao IN-AE.

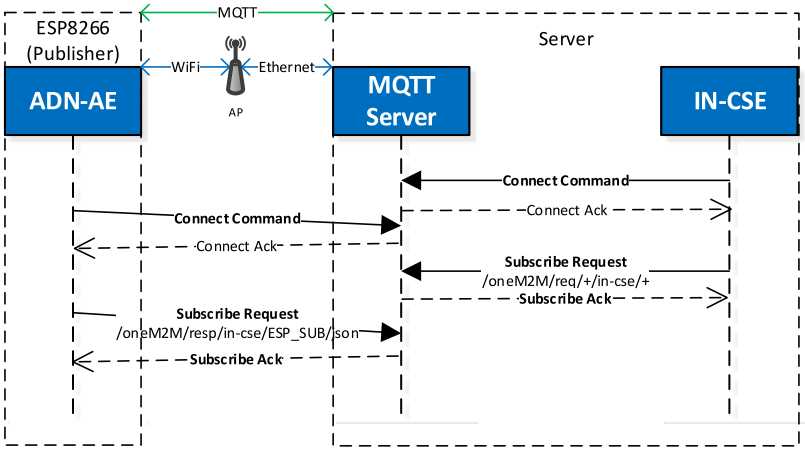


Figura 6.3: Diagrama de sequências representativo do processo de inicialização do ADN-AE e do IN-CSE

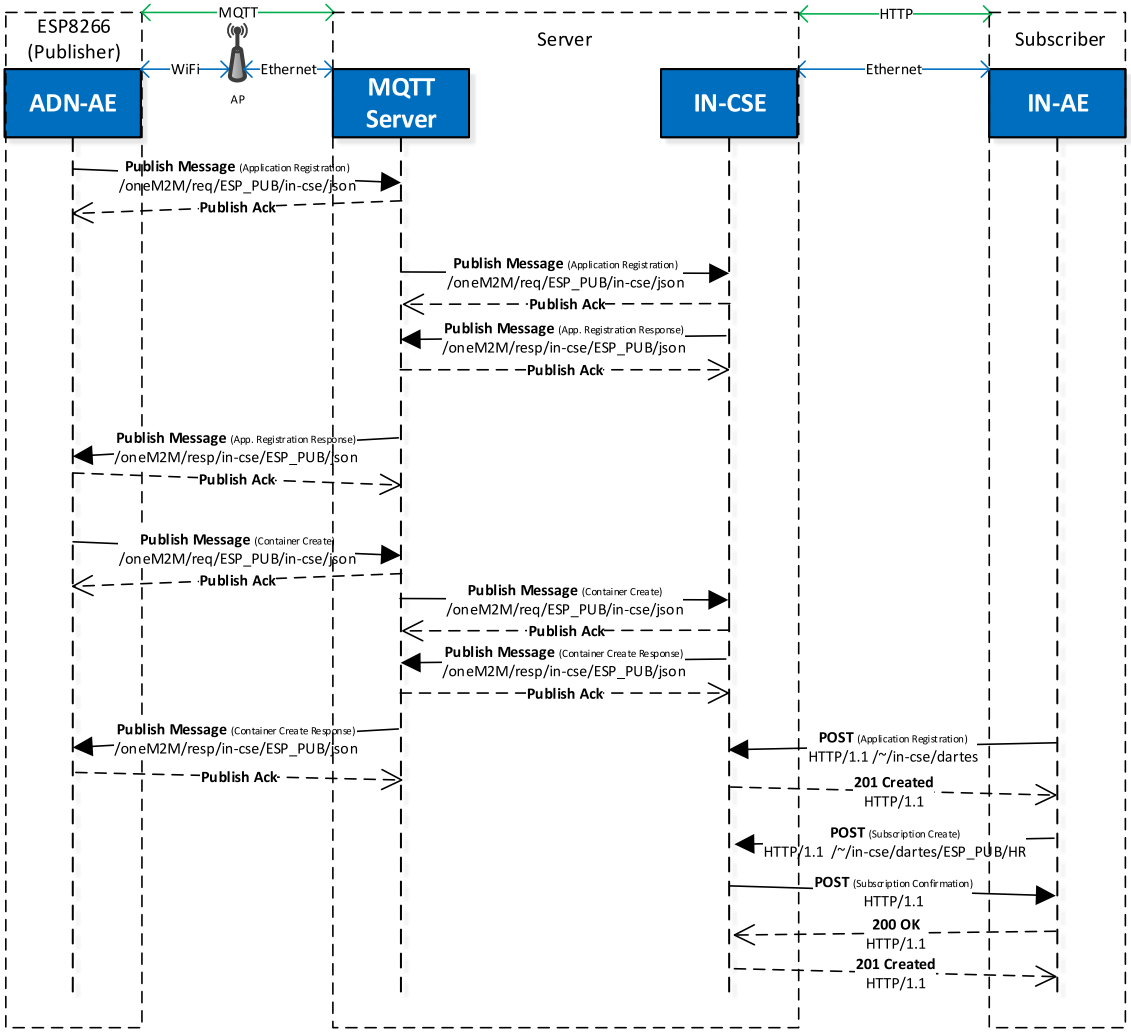


Figura 6.4: Diagrama de sequências representativo do processo de criação da AE e do *container* no IN-CSE, e respetiva subscrição do *container* criado pelo IN-AE

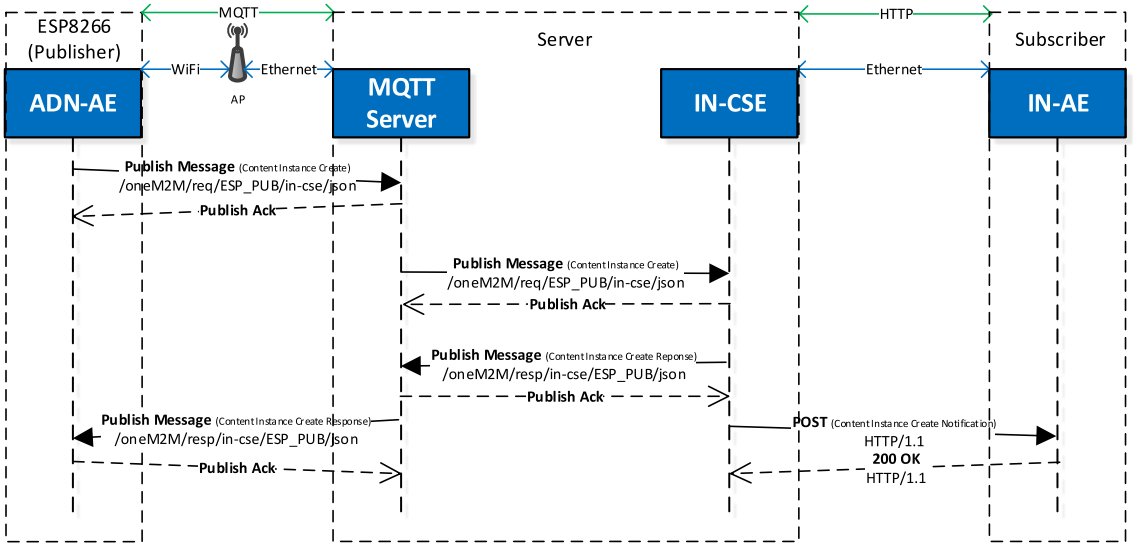


Figura 6.5: Diagrama de sequências representativo do processo de criação de uma instância de conteúdo pelo ADN-AE e respetiva notificação dessa criação ao IN-AE

6.3 Sumário

Ao longo deste capítulo foram apresentadas as considerações de *software* relacionadas com a integração do módulo ESP8266 numa *framework* para monitorização móvel de parâmetros fisiológicos de um conjunto de utilizadores. A arquitetura proposta é compatível com o *standard* oneM2M, utiliza o protocolo MQTT na camada de transporte e baseia-se num modelo de comunicação do tipo *publish-subscribe*. Neste contexto, o módulo ESP8266 publica periodicamente a informação recolhida por um sensor de batimento cardíaco, que é subscrita por uma unidade central que recebe notificações por cada uma dessas publicações. No capítulo 7 é analisada a performance do módulo ESP866 na arquitetura apresentada.

Capítulo 7

Resultados de performance na comunicação M2M

O uso de *middlewares* de comunicação com o apresentado no capítulo 6, promove a interoperabilidade entre sistemas, facilitando a conexão entre dispositivos com diferentes capacidades de processamento e diferentes interfaces de comunicação. Apesar disso, a interoperabilidade e standardização alcançadas, podem introduzir sobrecargas e atrasos adicionais nas comunicações entre os diversos intervenientes. Isto reflete-se na quantidade adicional de informação que tem de ser adicionada em cada transmissão e, consequentemente, pelo tempo adicional que cada transmissão necessita para ser concluída. De forma a avaliar a adequação do módulo ESP8266 a este tipo de soluções, este capítulo apresenta os resultados da avaliação de performance realizada. Foram considerados cenários com a transmissão de informação a diferentes frequências e diferentes tamanhos do *payload* da mensagem. Para cada um dos cenários considerados, foram feitas medições de consumo e de latência nas comunicações para todos os modos de funcionamento disponíveis (*Ativo*, *Modem-sleep*, *Light-sleep* e *Deep-sleep*).

7.1 Setup e metodologia utilizada

As experiências de avaliação de performance do módulo WiFi ESP8266, apresentadas ao longo deste capítulo, foram realizadas num ambiente *indoor*. O servidor M2M (IN-CSE) e o broker de MQTT foram instalados num servidor dedicado, com o sistema operativo CentOS 6.9, um processador Intel(R) Core(TM) i5-2500 CPU @3.30GHz e 8GB de memória RAM. Este servidor, foi ligado por uma conexão Ethernet de 100Mb/s ao subscriber (IN-AE), instalado num computador com Ubuntu 16.04 LTS, com um processador Intel(R) Core(TM) i7 - 4700HQ CPU @ 2.40GHz e 8GB de memória RAM.

Por sua vez, o módulo ESP8266 (ADN-AE) foi conectado a uma rede WiFi do tipo infraestruturada, utilizando como AP um *ASUS RT-AC87U dual-band AC240030*, configurado para difundir o Beacon a cada 100ms e a mensagem DTIM a cada 3 Beacons. Por fim, o AP foi conectado ao servidor através de uma conexão Ethernet de 100Mb/s.

Nas experiências que envolveram a medição da corrente consumida pelo módulo, seguiu-se a mesma metodologia apresentada no capítulo 5, com a utilização do *Low Voltage Power Monitor* (FTA22D), fabricado pela *Monsoon Solutions Inc*. A duração estimada da bateria apresentada para cada um dos modos de funcionamento, foi calculada considerando o tempo necessário para a mesma atingir 10% da sua capacidade, assumindo que o módulo era alimentado por uma bateria de 3.3V/1000mAh e a sua depleção era linear.

Nas medições de latência ponto-a-ponto, foi necessário estabelecer a sincronização dos relógios entre o ADN-AE (*Publisher*) e o IN-AE (*Subscriber*), uma vez que as duas entidades não se encontram na mesma máquina. Para tal, utilizou-se o *Network Time Protocol* (NTP), selecionando o mesmo servidor para sincronização de ambas as entidades. No caso dos modos de funcionamento, *Ativo*, *Modem-sleep* e *Light-sleep*, as atualizações do relógio interno do ADN-AE foram realizadas a cada 20s. Já nas experiências no modo de funcionamento *Deep-sleep*, uma vez que durante o período de funcionamento no modo de poupança de energia os registos de memória são apagados, é realizada uma atualização NTP antes de cada transmissão. As *timestamps* são registadas antes do envio da publicação pelo ADN-AE e após a receção e processamento da mesma no IN-AE.

Não são considerados nos resultados apresentados a corrente necessária para recolher informação do sensor de batimento cardíaco ou outro. Em vez disso, foram criados *payloads* representativos da informação que seria recolhida por um sensor desse tipo. Assim, tal como em [43], considerou-se um sensor de batimento cardíaco (HR) onde cada medição é representada por um *payload* de 85B. Foram considerados cenários com transmissões a cada 1s e a cada 10s, mantendo o mesmo *goodput* de dados em ambos os casos. Desta forma, foi avaliado o impacto no consumo e latência da comunicação com o envio de 1 medição de HR (85B) a cada 1s e acumulação de de 10 medições de HR (850B) a cada 10s. Com exceção do modo de *Deep-sleep*, onde o intervalo de tempo (superior a 1s) necessário para o estabelecimento da ligação ao AP, apenas permitiu o envio com intervalos de transmissão de 10s. Cada umas das medições é publicada pelo ADN-AE, criando uma nova instância num *container* pré-existente, ao qual o IN-AE subscreve.

Por último, não foram consideradas nas medições a troca inicial de mensagens (registo do ADN-AE, criação do *container*, ...). Para cada um dos cenários referidos, foram realizadas medidas durante um intervalo de tempo necessário à realização de 100 publicações.

7.2 Medições de consumo

Na figura 7.1 é apresentada a distribuição dos resultados das medições de consumo de corrente para cada um dos cenários referidos, considerando o nível de QoS1 no protocolo MQTT. A corrente média observada no modo ativo foi de aproximadamente 70.3mA para transmissões a cada 1s e o mesmo para transmissões a cada 10s. Os consumos de corrente verificados equivalem a uma autonomia de aproximadamente 12.8h, mostrando que, quando os modos de poupança de energia estão desativados, a alteração da frequência de envio não influência consideravelmente os

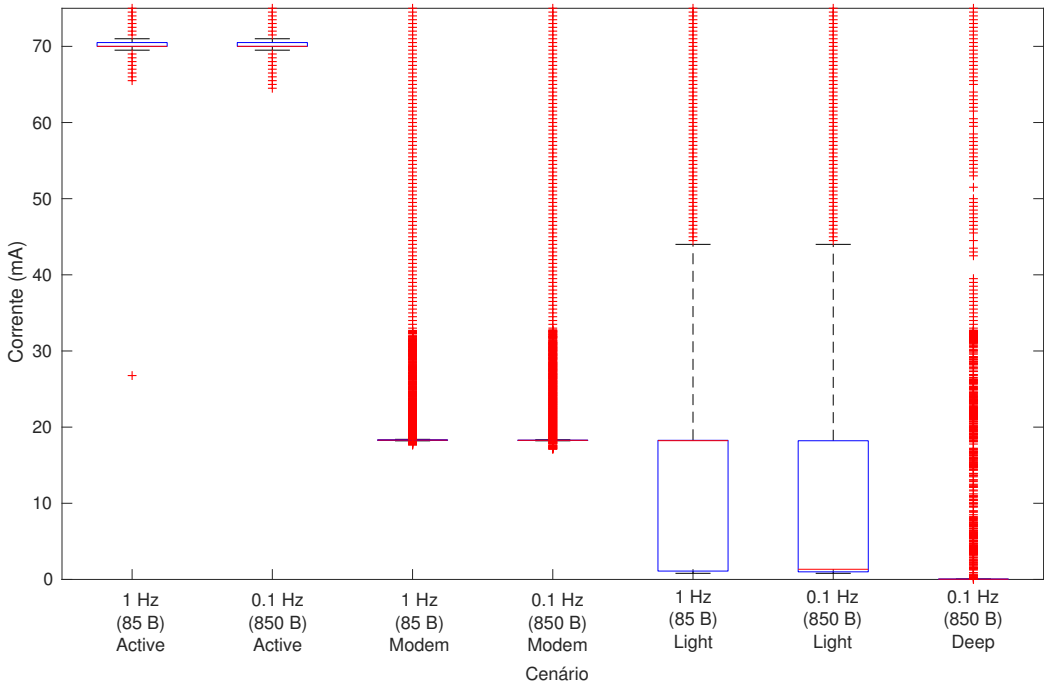


Figura 7.1: Distribuição do consumo de corrente em cada um dos cenários considerados

consumos. Por outro lado, o modo de *Modem-sleep* apresentou uma redução assinalável no consumo, quando comparado com o modo ativo. No cenário com transmissões a cada 1s verificou-se um consumo médio de corrente de 30.7mA (56.3% de redução quando comparado com o modo Ativo) e o cenário com transmissões a cada 10s requereu um consumo médio de 23.8mA (66.1% de redução quando comparado com o modo Ativo). Assim, neste modo de funcionamento é possível obter-se uma autonomia de 29.4h e 37.7h, respetivamente. Adicionalmente, verificou-se que, tal como esperado, a ativação do modo de *Light-sleep* permite atingir uma maior eficiência no consumo energético. Apesar da elevada variabilidade do consumo de corrente verificada neste modo de funcionamento causada pelo desligar intermitente do CPU, foi possível alcançar um consumo de 22.2mA (38.3% de redução quando comparado com o *Modem-sleep*) com transmissões a cada 1s e de 14.2mA (67.6% de redução quando comparado com o *Modem-sleep*) com transmissões a cada 10s. O modo *Light-sleep* mostrou assim permitir um incremento de 67.6% (quando comparado com o *Modem-sleep*) na autonomia máxima do módulo, ao atingir um valor esperado de 63h, no caso de transmissões a cada 10s. Por último, o modo de *Deep-sleep* mostrou ser o mais eficiente, a nível energético, ao atingir consumos médios de apenas 10.0mA (29.6% de redução quando comparado com o *Light-sleep*) para transmissões a cada 10s, permitindo assim uma autonomia superior a 90h.

Adicionalmente, os resultados apresentados mostram que o aumento da frequência de envio pode, de facto, beneficiar largamente o consumo energético nos modos de *Modem-sleep* e *Light-sleep*. Por fim, a tabela 7.1 sumariza a corrente média/mediana consumida e apresenta a duração estimada da bateria para cada um dos modos de funcionamento.

Tabela 7.1: Corrente média e autonomia da bateria expectável para cada cenário

Cenário	**Corrente média /mediana (mA)**	**Duração bateria estimada (h)**
Ativo - 1Hz (85B)	70.3 / 70.0	12.8
Ativo - 0.1Hz (850B)	70.3 / 70.0	12.8
Modem-sleep - 1Hz (85B)	30.7 / 18.3	29.4
Modem-sleep - 0.1Hz (850B)	23.8 / 18.3	37.7
Light-sleep - 1Hz (85B)	22.2 / 18.2	40.6
Light-sleep - 0.1Hz (850B)	14.2 / 1.3	63.2
Deep-sleep - 0.1Hz (850B)	10.0 / 0.03	89.9

7.3 Avaliação da latência ponto-a-ponto

A figura 7.2 mostra em detalhe a distribuição da latência ponto-a-ponto para cada umas das 100 publicações, na configuração com um *payload* de 85B e envio periódico a cada 1s. Como referido no capítulo 4, a operação no modo de *Light-sleep* implica que o relógio de sistema (com oscilador de cristal preciso) seja desligado. Em vez desse último, é utilizado um oscilador RC interno com menor precisão. Como a figura 7.2 permite observar, as medições comprovaram a baixa precisão desse oscilador RC, sendo possível observar um evidente *drift* nas medições de latência obtidas no modo de funcionamento *Light-sleep*. Esse desvio é corrigido, a cada 20s, com a atualização do relógio interno pelo servidor NTP.

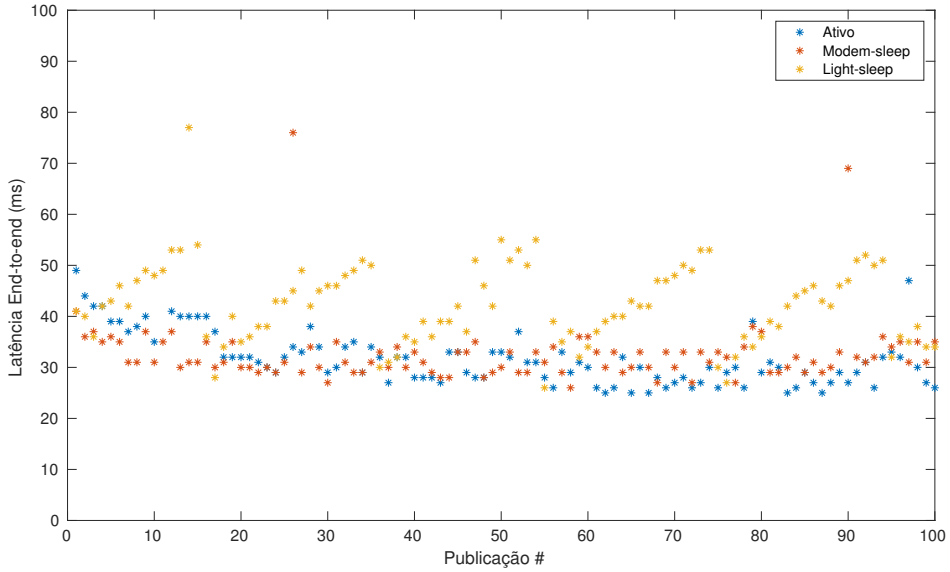


Figura 7.2: Latência ponto-a-ponto medida para publicações a cada 1s mostrando o *drift* do relógio no modo de *Light-sleep*

De forma a minimizar os efeitos da existência desse *drift*, no cálculo da latência média no modo de *Light-sleep*, realizou-se a estimação do mesmo através da determinação do declive da reta associada às medições nesse modo. A figura 7.3 mostra a reta obtida, para medições de latência por um intervalo de tempo de 20s (sem atualização do relógio pelo servidor NTP). Foi então aplicado um fator de correção de acordo com o declive determinado (1.14), para todas as medições nesse modo de funcionamento, tal como mostra a figura 7.4.

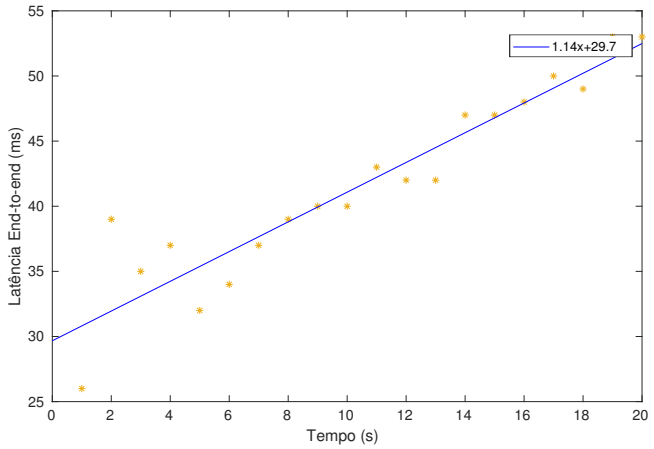


Figura 7.3: Drift do relógio no modo de *Light-sleep*

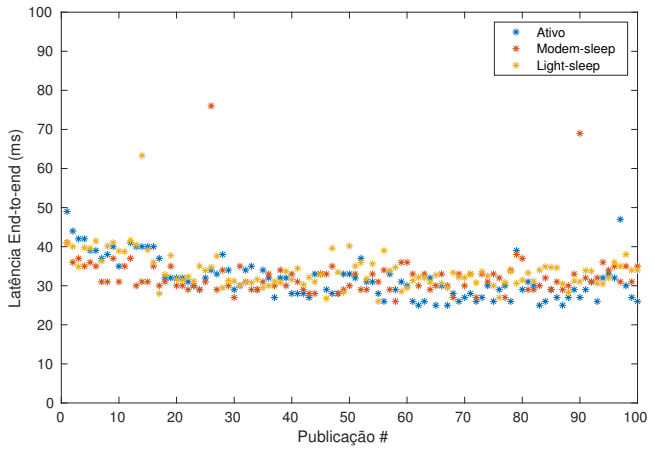


Figura 7.4: Latência ponto-a-ponto medida para publicações a cada 1s com correção do *drift* do relógio no modo de *Light-sleep*

A figura 7.5, mostra a distribuição da latência ponto-a-ponto para cada um dos cenários considerados, já refletindo a correção do *drift* do relógio no caso do modo de *Light-sleep*. Os cenários com uma frequência de transmissão de 1Hz (85B) levaram a uma latência ponto-a-ponto média de 31ms, 33ms e 34ms, no modo Ativo, *Modem-sleep* e *Light-Sleep*, respetivamente. Os resultados obtidos evidenciam que a configuração de operação no modo Ativo, beneficia ligeiramente o valor de atraso observado, quando comparado com os modos *Modem-sleep* e *Light-sleep*. Uma vez que

durante o período de baixo consumo as interfaces de comunicação WiFi se encontram desligadas, a diferença observada pode ser explicada pelo tempo adicional necessário à ativação das mesmas.

No caso dos cenários com frequências de transmissão de 0.1Hz (850B), obteve-se uma latência ponto-a-ponto média de 32ms, 34ms e 35ms, no modo Ativo, *Modem-sleep* e *Light-Sleep*, respetivamente. Estes resultados mostram valores semelhantes aos obtidos com a configuração com transmissão a cada 1Hz (85B), indicando que a diferença no tamanho do *payload* da mensagem (85B vs 850B) têm um impacto negligenciável na latência obtida.

Por último, foi possível observar uma latência média de 1230ms no modo de *Deep-sleep*. O elevado valor observado quando comparado com os restantes modos de funcionamento, deve-se à necessidade de efetuar os mecanismos para estabelecer uma nova conexão WiFi cada vez que o módulo retorna do estado de *Deep-sleep*, obrigando à execução dos procedimentos de autenticação, associação e obtenção de um endereço IP junto do AP.

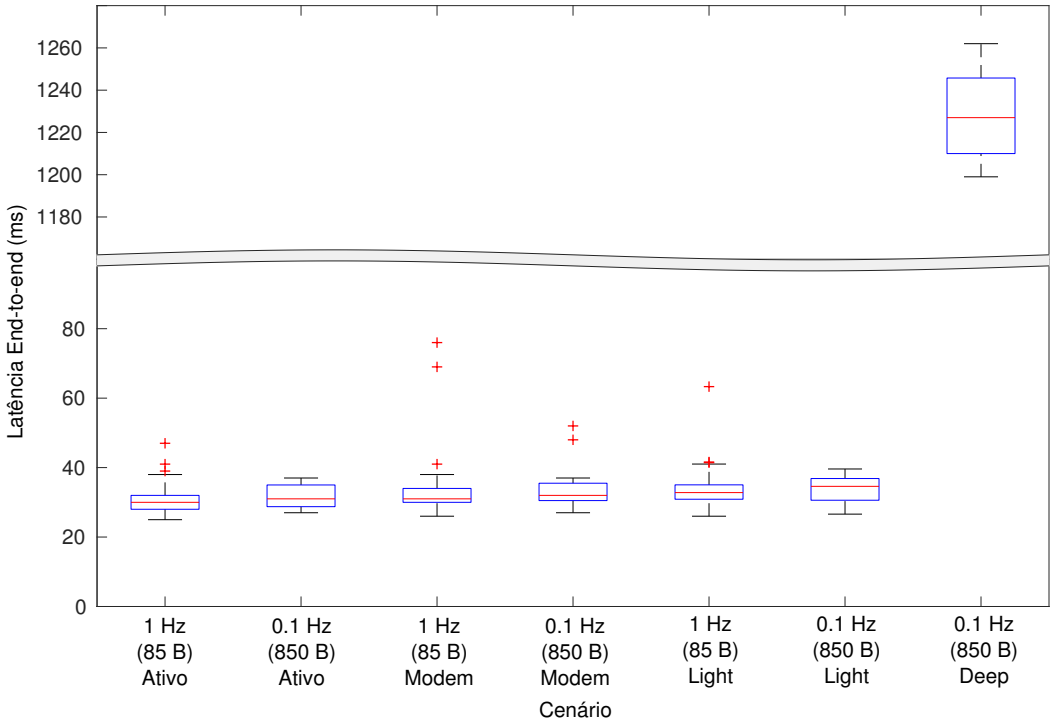


Figura 7.5: Distribuição da latência ponto-a-ponto para cada um dos cenários

7.4 Variação do QoS em MQTT e comparação com CoAP

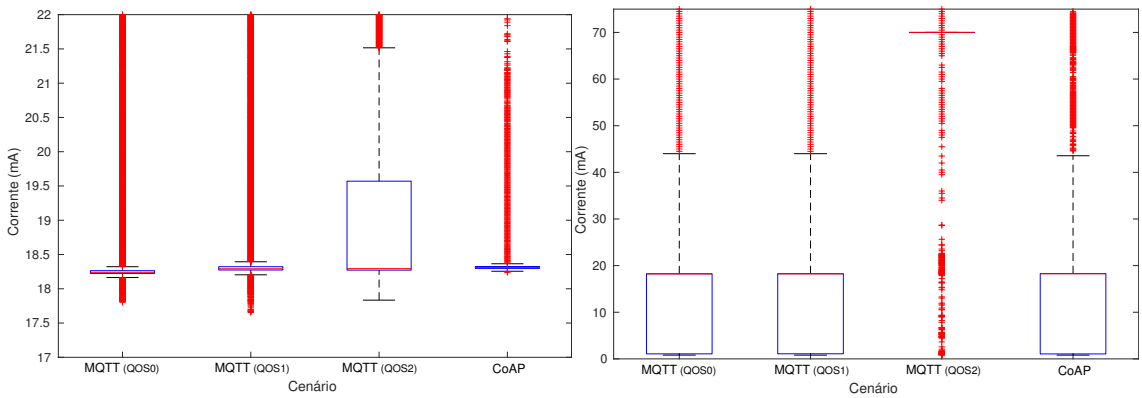
Como explicado no capítulo 2, o protocolo de troca de mensagens MQTT suporta diferentes níveis de QoS na entrega das publicações efetuadas. Enquanto o nível de QoS0 oferece níveis de garantia de entrega semelhantes ao protocolo TCP subjacente, ao envolver a troca de apenas uma mensagem entre o cliente e o *broker* MQTT, os níveis de QoS1 e QoS2 conseguem oferecer garantias adicionais de que uma publicação enviada foi de facto entregue. Assim, o nível de QoS1 envolve a troca de duas mensagens (envio da publicação e receção da confirmação) garantindo

assim que a publicação foi entregue pelo menos uma vez. Por outro lado, o nível de QoS2 oferece a garantia que a publicação é entregue apenas uma vez, mas envolve a troca de quatro mensagens por cada publicação. Assim, torna-se evidente que diferentes configurações para o nível de QoS de uma mensagem MQTT, podem significar atrasos adicionais e um consumo energético superior.

Adicionalmente, foram realizadas experiências semelhantes com uma implementação [**?**] que utiliza o protocolo CoAP, com mensagens não confirmáveis, para a mesma aplicação. Desta forma, torna-se relevante a realização de uma comparação da variação verificada no consumo energético e na latência entre os diferentes níveis de QoS oferecidos pelo protocolo MQTT (TCP) e a referida implementação de CoAP (UDP). Esta secção pretende assim avaliar esse impacto no módulo WiFi em estudo.

7.4.1 Consumo de corrente médio

A figura 7.6 mostra a distribuição do consumo energético para as diferentes configurações no modo de *Modem-sleep* (7.6a) e *Light-sleep* (7.6b). No caso do modo *Modem-sleep*, foi possível obter uma corrente média (e autonomia) de 27.2mA (33.1h) e 27.0mA(33.4h), para o nível de QoS0 e CoAP, respetivamente. Estes resultados mostram uma diferença negligível entre a utilização de MQTT com o nível de QoS0 (TCP) e CoAP (UDP), apesar da utilização de protocolos diferentes na camada de transporte. Tal como mostra a figura 7.7, o tempo em que a interface WiFi se mantêm ligada em ambas as configurações é igual a 90ms, o que explica o consumo médio semelhante verificado. Pelo contrário, a troca de mensagens e confirmações adicionais inerentes aos níveis de QoS1 e QoS2 provocou um incremento de 11.4% e 14.2% no consumo, com um valor de corrente médio (e autonomia) de 30.7mA (29.4h) e 31.7mA (28.4h), respetivamente.



(a) Modem-sleep

(b) Light-sleep

Figura 7.6: Distribuição do consumo de corrente nos diferentes cenários considerados

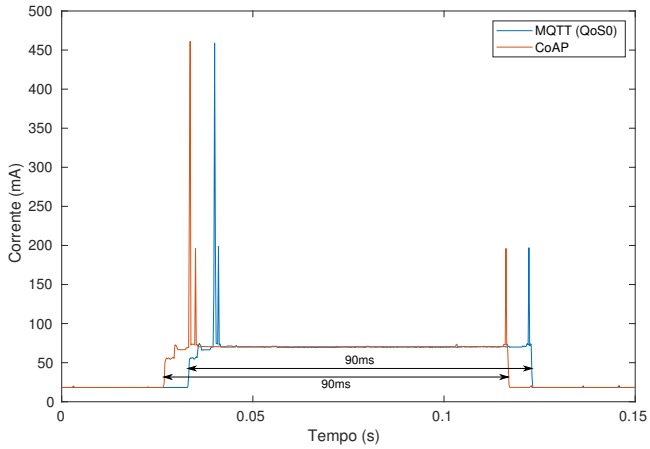


Figura 7.7: Consumo de corrente no envio de uma publicação MQTT e CoAP

Os cenários com a ativação do modo de *Light-sleep* também resultaram em valores de corrente média (e autonomia) muito semelhantes entre si com 22.0mA (40.9h), 22.2mA (40.6h) e 22.1mA (40.7h), para as configurações com QoS0, QoS1 e CoAP, respetivamente. Pelo contrário, a configuração do QoS para o nível 2 levou a um aumento do consumo de corrente para uns expressivos 69.2mA (13.0h). Neste caso, verificou-se que, apesar de o módulo ser configurado para operar no modo de *Light-sleep*, manteve o CPU e as interfaces WiFi ligadas durante todo o período de teste, o que corresponde ao funcionamento no modo Ativo. Adicionalmente, foi verificado se o mesmo comportamento era observado para frequências de transmissão superiores (10s), o que não aconteceu. Assim, uma vez que na operação com QoS2 no modo de *Modem-sleep* não foi verificado o mesmo comportamento, a situação reportada pode dever-se a alguma falha na implementação do *firmware* que controla o modo de poupança de energia *Light-sleep*.

7.4.2 Latência ponto-a-ponto

Uma vez que os resultados discutidos na secção 7.3 mostraram que o modo de *Light-sleep* apresenta valores de latência médios semelhantes ao modo de *Modem-sleep* e uma vez que a precisão do relógio ativo nesse modo é reduzida, optou-se por excluir o mesmo desta comparação. Assim, a figura 7.8 apresenta a distribuição da latência ponto-a-ponto observada para as diferentes configurações, nos modos *Ativo* e *Modem-sleep*.

Nos cenários com o modo de operação *Ativo*, foi possível observar uma latência média 26ms, 31ms e 32ms, para as configurações utilizando o protocolo MQTT com QoS0, QoS1 e QoS2, respetivamente. Estes resultados mostram um incremento significativo (19%) entre o nível de QoS mínimo e os níveis 1 e 2, mostrando a influência da troca de mensagens e confirmações adicionais inerentes a essas configurações. Por outro lado, a utilização de CoAP mostrou um decréscimo de 12% (em relação a QoS0), ao obter-se um valor de 23ms de latência média. Assim, ficou evidenciada a influência da utilização de diferentes protocolos na camada de transporte na latência das comunicações, com um claro beneficio nesse domínio para UDP (CoAP).

A operação no modo de *Modem-sleep* mostrou resultados semelhantes. Neste modo, foi possível observar uma latência média de 31ms, 33ms, 38ms, para as configurações utilizando o protocolo MQTT com QoS0, QoS1 e QoS2, respetivamente. Já na configuração utilizando o protocolo CoAP, foi possível obter uma latência média de 27ms (13% de redução quando comparado com MQTT QoS0).

Em suma, os resultados obtidos mostram que, de facto, o incremento da fiabilidade das comunicações pode introduzir atrasos adicionais que, dependendo do contexto de cada aplicação, podem não ser negligenciáveis. Ainda assim, a diferença máxima de 8ms verificada entre o protocolo CoAP e o nível intermédio de QoS em MQTT, pode justificar as garantias adicionais introduzidas por este último.

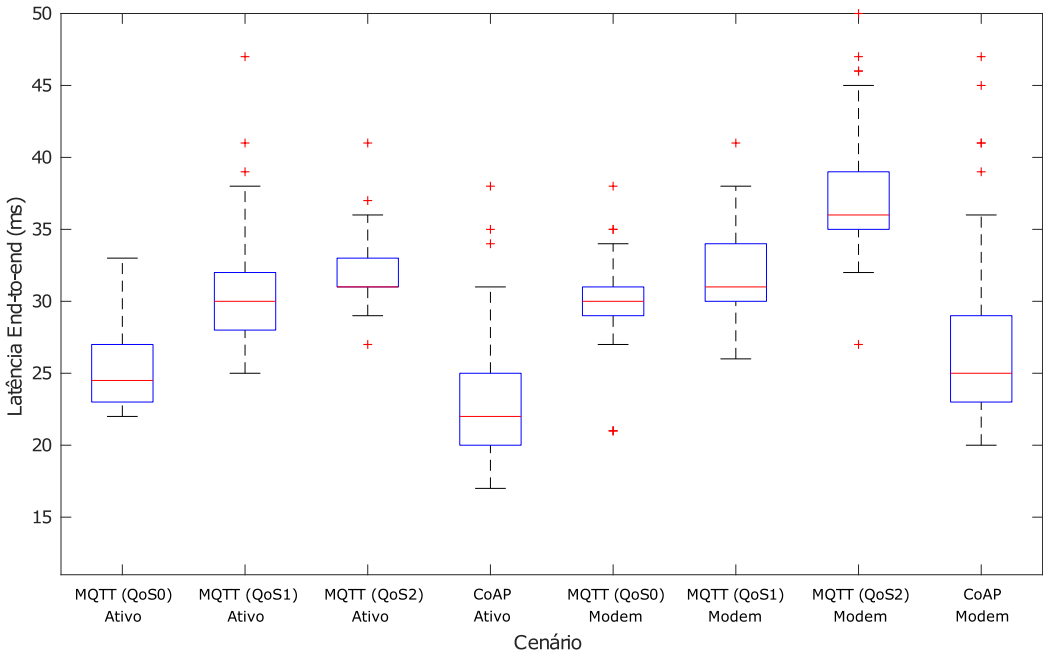


Figura 7.8: Distribuição da latência ponto-a-ponto para cada um dos cenários

7.5 Sumário

Ao longo deste capítulo foram apresentados a metodologia e os resultados das experiências de avaliação de performance do módulo ESP8266 na arquitetura de comunicação proposta no capítulo 6. Em suma, os resultados obtidos mostram a adequação do módulo ESP8266 a aplicações deste tipo, ao ser possível a transmissão em continuo de informação com intervalos entre 1s e 10s, com uma autonomia que depende do modo de funcionamento, mas que pode chegar até cerca de 90h (modo de *Deep-sleep*). Por último, a latência média associada as comunicações mostrou ser reduzida e variar ligeiramente, entre 32ms e os 35ms, para os diferentes modos de funcionamento.

Capítulo 8

Conclusões e trabalho futuro

Tendo em conta a importância que a tecnologia WiFi tem nos dias de hoje, a utilização de sistemas embarcados com suporte a essa tecnologia pode ser um passo em frente no desenvolvimento do paradigma da *Internet of Things* (IoT) e dos sistemas de monitorização em saúde. Em particular, a reutilização de toda a infraestrutura WiFi já existente permite evitar a utilização de *gateways* para a conversão entre protocolos, reduzindo os custos de implementação desses sistemas. Os mais recentes avanços tecnológicos levaram à introdução de pequenos sistemas embarcados com suporte à tecnologia WiFi e desenhados especificamente para atender aos requisitos desse tipo de aplicações. Apesar disso, são ainda escassas análises detalhadas do consumo energético, alcance das comunicações e da performance desse tipo de dispositivos nos referidos cenários.

O trabalho apresentado ao longo desta dissertação pretendeu contribuir para essa convergência em torno da tecnologia WiFi e da utilização desses dispositivos. Em particular, foi realizada uma caracterização em detalhe de um dispositivo recém introduzido no mercado com essas características, o modulo WiFi ESP8266 da *Espressif Systems*. Nomeadamente, foram considerados os diferentes modos de poupança de energia disponíveis no módulo em estudo, caracterizando o funcionamento das interações automáticas com a infraestrutura WiFi e do impacto que diferentes configurações de parâmetros como o *Beacon interval* e o DTIM *period* têm no consumo energético. Os resultados alcançados mostram que o consumo de corrente médio necessário para manter a associação do módulo ao AP é, nos casos mais favoráveis, de 12mA para o funcionamento no modo de *Light-sleep* e de cerca de 20mA para o modo de *Modem-sleep*. Foi ainda possível verificar que este módulo consegue fornecer garantias de conectividade em cenários de comunicação no interior de edifícios, mostrando PDRs superiores a 99% no mesmo piso de um edifício através de várias divisões.

Foram ainda implementados os componentes de *software* que permitem a integração deste módulo numa *framework*, baseada no *standard* oneM2M, que pretende habilitar o seguimento e monitorização em contínuo dos sinais vitais de pacientes em ambiente hospitalar. Foi considerado um modelo de comunicação do tipo *publish-subscribe*, onde se estudou a possibilidade de instalação do módulo WiFi num dispositivo *wearable*. Nesse contexto, seriam realizadas pelo módulo publicações periódicas de dados contendo informação dos sinais vitais de um utilizador, que eram

subscritas por outra entidade (um sistema de EHR, por exemplo). Os resultados das experiências de avaliação de performance nesse cenário, mostram que é possível manter associação ao AP e realizar transmissão de dados com um consumo médio de corrente mínimo (modo *Light-sleep*) de 22.17mA e 14.2mA, para frequências de transmissão de 1 e 10s, respetivamente. Considerando a alimentação do módulo com uma bateria de 1000mAh de capacidade, estes valores de corrente média garantem autonomias máximas de cerca de 63h. Se a aplicação dispensar a manutenção da associação com o AP, é possível ainda considerar a utilização do modo de *Deep-sleep*. Nesse caso foi possível atingir um consumo de corrente médio, com transmissão de dados a cada 10s, de apenas 10.0mA, permitindo um incremento da autonomia para cerca de 90h. Ao mesmo tempo, as experiências de avaliação de latência ponto-a-ponto (entre o *publisher* e o *subscriber*) mostram que os diferentes modos de funcionamento podem também influenciar esse parâmetro. Com a configuração de operação nos modos Ativo, *Modem-sleep* e *Light-sleep* foi possível obter latências média de cerca de 33ms, o que mostrou aptidão para operação no cenário considerado. Por outro lado, a operação no modo de *Deep-sleep* mostrou um grande incremento desse fator, com o mesmo a apresentar valores superiores a 1s, devido à necessidade de realização dos processos de associação ao AP antes de cada transmissão.

Em conjunto, os resultados obtidos mostram a aptidão da tecnologia WiFi e do módulo WiFi ESP8266 em particular, para operar em cenários de aplicações IoT e de comunicação M2M com restrições no consumo de energia. Ainda assim, indicam que é imprescindível um planeamento cuidado na escolha de diferentes valores para os parâmetros de configuração no AP, para os modos de funcionamento do módulo e para as frequências de transmissão utilizadas, uma vez que a escolha desses parâmetros mostrou ter uma grande influência em todos os restantes fatores em analise.

8.1 Trabalho futuro

O trabalho realizado pode ser continuado de algumas formas. Nomeadamente, podem ser considerados os modos de funcionamento do módulo como estação WiFi e softAP em simultâneo. Este modo de operação, conhecido como WiFi Direct, pode ser útil em cenários onde a infraestrutura WiFi não se encontra disponível.

No âmbito da integração do módulo na *framework* de monitorização remota, podem ser considerados cenários onde o módulo realiza subscrições de diferentes tópicos, para, por exemplo, configuração dos modos de operação ou da frequência de envio de dados. Nessa linha de trabalho, seria interessante avaliar o impacto do consumo energético da operação com essa funcionalidade, bem como podia ser quantificado o atraso envolvido desde a transmissão de determinada ordem de funcionamento até a sua execução. Adicionalmente, apesar dos mecanismos de segurança inerentes à tecnologia WiFi, pode ser considerada a utilização nas publicações de dados pelo módulo da versão segura do protocolo MQTT. A utilização de mecanismos de segurança como TLS (*Transport Layer Security*), permitem assegurar a privacidade das comunicações, mas necessitam

de capacidade de processamento adicional. Seria por isso interessante avaliar esse impacto no módulo em estudo.

Referências

[1] Population structure and ageing - Statistics Explained. URL: http://ec.europa.eu/eurostat/statistics-explained/index.php/Population_structure_and_ageing.

[2] Giuseppe Anastasi, Marco Conti, Mario Di Francesco, e Andrea Passarella. Energy conservation in wireless sensor networks: A survey. *Ad Hoc Networks*, 7(3):537–568, 5 2009. URL: https://www.sciencedirect.com/science/article/pii/S1570870508000954?via%3Dihub, doi:10.1016/J.ADHOC.2008.06.003.

[3] Jennifer Yick, Biswanath Mukherjee, e Dipak Ghosal. Wireless sensor network survey. *Computer Networks*, 2008. URL: https://www.sciencedirect.com/science/article/pii/S1389128608001254, doi:10.1016/j.comnet.2008.04.002.

[4] Benoît Latré, Bart Braem, Ingrid Moerman, Chris Blondia, e Piet Demeester. A survey on wireless body area networks. *Wireless Networks*, 2011. URL: https://link.springer.com/article/10.1007/s11276-010-0252-4, doi:10.1007/s11276-010-0252-4.

[5] D. P. Tobon, T. H. Falk, e M. Maier. Context awareness in WBANs: a survey on medical and non-medical applications. *IEEE Wireless Communications*, 20(4):30–37, 8 2013. URL: http://ieeexplore.ieee.org/document/6590048/, doi:10.1109/MWC.2013.6590048.

[6] Ala Al-Fuqaha, Mohsen Guizani, Mehdi Mohammadi, Mohammed Aledhari, e Moussa Ayyash. Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications. *IEEE Communications Surveys & Tutorials*, 17(4):2347–2376, 24 2015. URL: http://ieeexplore.ieee.org/document/7123563/, doi:10.1109/COMST.2015.2444095.

[7] William Stallings, Marcia J Horton, Mack Patterson, e Pearson Prentice Hall. DATA AND COMPUTER COMMUNICATIONS Upper Saddle River, New Jersey 07458 Library of Congress Cataloging-in-Publication Data on File. URL: http://www.portcity.edu.bd/ELibrary/CSE/Dataandcomputercommunications.pdf.

[8] Xavier Pérez-Costa, Daniel Camps-Mur, e Albert Vidal. On distributed power saving mechanisms of wireless LANs 802.11e U-APSD vs 802.11 power save mode. *Computer Networks*, 51(9):2326–2344, 6 2007. URL: https://www.sciencedirect.com/science/article/pii/S1389128607000291, doi:10.1016/J.COMNET.2007.01.026.

[9] ESP8266 Datasheet. URL: https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf.

[10] ESP8266 Low Power Solutions. URL: https://www.espressif.com/sites/default/files/9b-esp8266-low_power_solutions_en_0.pdf.

[11] oneM. MQTT Protocol Binding. 2018. URL: http://www.onem2m.org/images/files/deliverables/Release2A/TS-0010-MQTT_protocol_binding-v_2_7_1.pdf.

[12] Mortality and life expectancy statistics - Statistics Explained. URL: http://ec.europa.eu/eurostat/statistics-explained/index.php/Mortality_and_life_expectancy_statistics.

[13] Guang-Zhong Yang. *Body Sensor Networks*. Springer-Verlag London, 2 edição, 2014. URL: http://www.springer.com/gp/book/9781447163732, doi:10.1007/978-1-4471-6374-9.

[14] V. Raghunathan, C. Schurgers, Sung Park, e M.B. Srivastava. Energy-aware wireless microsensor networks. *IEEE Signal Processing Magazine*, 19(2):40–50, 3 2002. URL: http://ieeexplore.ieee.org/document/985679/, doi:10.1109/79.985679.

[15] Ian F. Akyildiz e Mehmet Can Vuran. *Wireless Sensor Networks*. John Wiley & Sons, Ltd, Chichester, UK, 8 2010. URL: http://doi.wiley.com/10.1002/9780470515181, doi:10.1002/9780470515181.

[16] Raytheon: Boomerang III. URL: https://www.raytheon.com/capabilities/products/boomerang/.

[17] Vigilnet | Electronic Monitoring, GPS Tracking, Breath Alcohol Monitoring. URL: http://vigilnet.com/.

[18] Milsar - Home. URL: http://milsar.com/.

[19] AVTECH - Monitor Temperature and Environment Conditions with Room Alert. URL: https://avtech.com/.

[20] inovgrid. URL: http://www.inovgrid.pt/.

[21] MEO Smart Home | MEO. URL: https://www.meo.pt/pacotes/meo-smart-home.

[22] Thaier Hayajneh, @bullet Ghada, Almashaqbeh @bullet, Sana Ullah, @bullet Athanasios, e V Vasilakos. A survey of wireless technologies coexistence in WBAN: analysis and open research issues. 2014. URL: https://link.springer.com/content/pdf/10.1007%2Fs11276-014-0736-8.pdf, doi:10.1007/s11276-014-0736-8.

[23] BIOTRONIK Home Monitoring. URL: https://www.biotronik.com/en-de/products/home-monitoring.

[24] LATITUDE™ NXT Patient Management System - Boston Scientific. URL: http://www.bostonscientific.com/en-US/products/remote-patient-monitoring/latitude.html.

[25] V. Mighali, L. Patrono, M. L. Stefanizzi, Joel J. P. C. Rodrigues, e Petar Solic. A smart remote elderly monitoring system based on IoT technologies. Em *2017 Ninth*

International Conference on Ubiquitous and Future Networks (ICUFN), páginas 43–48. IEEE, 7 2017. URL: http://ieeexplore.ieee.org/document/7993745/, doi:10.1109/ICUFN.2017.7993745.

[26] P.S. Pandian, K. Mohanavelu, K.P. Safeer, T.M. Kotresh, D.T. Shakunthala, Parvati Gopal, e V.C. Padaki. Smart Vest: Wearable multi-parameter remote physiological monitoring system. *Medical Engineering & Physics*, 30(4):466–477, 5 2008. URL: https://www.sciencedirect.com/science/article/pii/S1350453307000975, doi:10.1016/J.MEDENGPHY.2007.05.014.

[27] Gérald Santucci e Sebastian Lange. Internet of Things in 2020 A ROADMAP FOR THE FUTURE RFID WORKING GROUP OF THE EUROPEAN TECHNOLOGY PLATFORM ON SMART SYSTEMS INTEGRATION (EPOSS). 2008. URL: https://www.smart-systems-integration.org/public/documents/publications/Internet-of-Things_in_2020_EC-EPoSS_Workshop_Report_2008_v3.pdf.

[28] MQTT V3.1 Protocol Specification. URL: http://public.dhe.ibm.com/software/dw/webservices/ws-mqtt/mqtt-v3r1.html.

[29] Cesare Pautasso, Olaf Zimmermann, e Frank Leymann. RESTful Web Services vs. " Big " Web Services: Making the Right Architectural Decision. 2008. URL: http://delivery.acm.org/10.1145/1370000/1367606/p805-pautasso.pdf?ip=193.136.33.225&id=1367606&acc=ACTIVE%20SERVICE&key=2E5699D25B4FE09E%2E6F699AA92A518455%2E4D4702B0C3E38B35%2E4D4702B0C3E38B35&__acm__=1529798283_60a727ff03abf364d2e8ff77afb58480.

[30] RFC 7252 - The Constrained Application Protocol CoAP. 2014. URL: http://www.rfc-editor.org/info/rfc7252.

[31] Radio Versions | Bluetooth Technology Website. URL: https://www.bluetooth.com/bluetooth-technology/radio-versions.

[32] Serbulent Tozlu. Feasibility of Wi-Fi enabled sensors for Internet of Things. Em *2011 7th International Wireless Communications and Mobile Computing Conference*, páginas 291–296. IEEE, 7 2011. URL: http://ieeexplore.ieee.org/document/5982548/, doi:10.1109/IWCMC.2011.5982548.

[33] New York, Chicago San, Francisco Lisbon, e London Madrid. CWAP Certified Wireless Analysis Professional ™ Official Study Guide (Exam PW0-205) McGraw-Hill/Osborne CWAP Certified Wireless Analysis Professional Official Study Guide (Exam PW0-205) First Edition. URL: https://www.cwnp.com/wp-content/uploads/pdf/CWAP_WLAN_ANALYSIS.pdf.

[34] Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications IEEE Computer Society. 2012. URL: https://legal.vvv.enseirb-matmeca.fr/download/amichel/%5BStandard%20LDPC%5D%20802.11-2012.pdf.

[35] Latest ESP8266 SDK based on FreeRTOS. URL: https://github.com/espressif/ESP8266_RTOS_SDK.

[36] The FreeRTOS™ Reference Manual. URL: https://www.freertos.org/Documentation/FreeRTOS_Reference_Manual_V9.0.0.pdf.

[37] ESP8266 nonOS SDK. URL: https://github.com/espressif/ESP8266_NONOS_SDK.

[38] esp-open-rtos framework. URL: https://github.com/SuperHouse/esp-open-rtos.

[39] ESP8266 Power Consumption - ESP8266 Developer Zone. URL: https://bbs.espressif.com/viewtopic.php?t=133.

[40] User Guide RT-AC87U Dual Band 4x4 Wireless-AC 2400 Gigabit Router. 2014. URL: http://www.produktinfo.conrad.com/datenblaetter/1200000-1299999/001276984-an-01-en-ASUS_RT_AC87U_AC2400_WLAN_ROUTER.pdf.

[41] Mobile Device Power Monitor Manual. URL: http://msoon.github.io/powermonitor/PowerTool/doc/LVPM%20Manual.pdf.

[42] Martin. Sauter. *From GSM to LTE-Advanced : an Introduction to Mobile Networks and Mobile Broadband.* Wiley, 2014.

[43] C. Pereira, D. Guimarães, J. Mesquita, F. Santos, L. Almeida, e A. Aguiar. Feasibility of Gateway-less IoT E-health Applications. Em *European Conference on Networks and Communications - EuCNC*, 6 2018.