

UNIVERSIDADE NOVA DE LISBOA  
Faculdade de Ciências e Tecnologia  
Departamento de Engenharia Electrotécnica e de Computadores

**PROTOCOLO MAC PARA ACESSO MULTI-MODO EM REDES  
DE SENSORES SEM FIOS MÓVEIS**

Por:  
Hugo Manuel Serrão Borda d' Água

Dissertação apresentada na Faculdade de Ciências e Tecnologia  
da Universidade Nova de Lisboa para a obtenção do grau de  
Mestre em Engenharia Electrotécnica e de Computadores

Orientador: Doutor Luís Filipe Lourenço Bernardo

**Lisboa  
2008**



*Aos meus pais,*

*aos meus avós*



---

## RESUMO

---

As redes de sensores sem fios móveis criam diversos requisitos em termos de acesso ao meio, como existência de um elevado débito, baixo consumo energético e manutenção de vizinhança. Nesta dissertação propõe-se um novo protocolo de acesso ao meio, o MMH-MAC, otimizado para funcionar de acordo com estes requisitos e para suportar aplicações que geram elevados picos de tráfego.

Este protocolo apresenta um modo de funcionamento assíncrono, destinado aos períodos em que os nós se encontram sem tráfego para transmitir (sem ligações), e um modo síncrono, a ser usado nos períodos em que se verifica um elevado débito. O modo de funcionamento é controlado pela aplicação através de uma aproximação *cross-layer*. O desenvolvimento deste protocolo foi efectuado de modo a permitir a presença de nós assíncronos na vizinhança de nós síncronos, possibilitando a existência de um reduzido tempo de associação e baixa interferência durante esse processo.

Os protocolos MAC existentes utilizam um método passivo para associar um nó assíncrono a um cluster estático, esperando pela recepção de um *beacon*. Neste protocolo é proposta uma aproximação activa que permite a um nó enviar pacotes de dados cerca de 300ms após entrar na vizinhança de um nó estático, praticamente sem existência de perda de dados nas ligações síncronas já existentes.

O desempenho do protocolo foi analisado através de simulações e tendo em conta vários parâmetros: tempo de associação, débito e consumo energético. Foram desenvolvidos vários modelos teóricos de modo a consolidar os resultados das simulações, de entre os quais assume maior relevo o referente ao tempo de associação.

---

## PALAVRAS-CHAVE

---

Redes de Sensores Sem fios Móveis, Protocolo de Controlo de Acesso ao Meio, Tráfego em Rajada, Comunicação entre Camadas.



---

## **ABSTRACT**

---

Mobile Wireless Sensor Networks (WSNs) present challenging requirements to the medium access control, such as high throughput, low energy consumption and neighborhood tracking. This thesis proposes a new medium access control protocol, the MMH-MAC, optimized to work in mobile wireless sensor networks and support applications that generate huge peaks of traffic.

This protocol provides an asynchronous mode for energy efficiency during idle (unconnected) periods, and a synchronous mode for high throughput periods. Mode change is controlled by the application in a cross-layer approach. The MMH-MAC protocol is specially designed to tolerate asynchronous mobile nodes in the neighborhood of synchronous nodes, providing fast transition and interference mitigation mechanisms.

Current MAC protocols rely on a passive method to associate a mobile node to a static cluster, triggered by the reception of a beacon. This protocol proposes an active approach that allows a mobile node to start sending data packets less than 300 ms after entering into data range of a static node, with negligible data loss in existing synchronous data connections.

This thesis includes simulation results focused in several parameters, such as association time, throughput and energy consumption. Theoretical models were proposed to support the simulation results, with the association time model assuming particular relevance.

---

## **KEYWORDS**

---

Mobile Wireless Sensor Networks, Medium Access Control Protocol, Bursty Traffic, Cross-layer.

---

---

## AGRADECIMENTOS

---

Na fase final da escrita desta dissertação, começo por agradecer a todos os Professores, amigos e colegas que contribuíram para a minha formação ao longo dos últimos anos.

Agradeço ao Prof. Luís Bernardo a orientação, a amizade e o apoio em todas as fases da elaboração desta dissertação. Nos momentos mais difíceis, em alguns casos quando estava perdido, incentivou-me e encaminhou-me, mas deixou sempre espaço para a criatividade individual. Agradeço-lhe a verificação cuidadosa do texto desta dissertação, mesmo em horários apertados, com correcções e melhoramentos valiosos em todas as secções.

Um agradecimento muito especial e sentido aos meus colegas e amigos Miguel Pereira, Miguel Silva e Ricardo Mendes, os quais já se encontravam a trabalharem nas suas dissertações na área das redes de sensores sem fios. Ao longo do desenvolvimento desta dissertação foram verdadeiramente incansáveis em todo o apoio prestado. Independentemente do horário, nas situações mais ou menos difíceis, sugerindo melhoramentos ou alargando horizontes, foram absolutamente fulcrais para a evolução desta dissertação.

Agradeço à Prof. Maria Helena Fino pelo apoio, entreaajuda e amizade durante o tempo em que paralelamente com o desenvolvimento desta dissertação fui monitor na FCT-UNL, tendo contribuído decisivamente para minha evolução humana e técnica. Nestas funções a contribuição dos meus colegas e amigos Pedro Monteiro e João Ferreira, foi também ela extraordinária, estando-lhes tremendamente grato

Agradeço a todos os outros colegas e amigos da secção de telecomunicações e da secção de electrónica pela entreaajuda na resolução de problemas, conselhos dados e convívio. Entre os quais, Cláudio Assunção, Michael Figueiredo, Bruno Esperança, Diogo Silva, Erik Snelling, Diogo Lourenço e João Martins.

Agradeço a todos os colegas que frequentaram comigo o curso nos últimos cinco anos na FCT-UNL, pelo convívio e por sempre se mostrarem prontos a ajudar e a cooperar nas mais diversas situações. Entre os quais, gostava de destacar o Jorge Ferreira, Bernardo

---

Barbosa, Luís Teixeira, Lourenço Sousa, Samuel Granadeiro, Tiago Santos, Luís Almeida, Bruno Ferreira, Alexandra Tavares, João Sousa e Nelson Silvério.

Um agradecimento muito especial à minha família, pai, mãe, avós, madrinha, padrinho, afilhado e amigos por terem partilhado comigo o outro lado da vida durante toda a realização do curso, e pela compreensão acerca de todas as horas que passei privado da sua companhia. Agradeço aos meus pais e avós por todo o apoio e carinho que me deram ao longo da minha vida e que sem eles nada disto teria sido possível.

Lisboa, Julho 2008

*Hugo Manuel Serrão Borda d' Água*

---

## ACRÓNIMOS

---

ABR – Associativity-Based Routing  
ACL – Auxiliary Clock  
ACK - Acknowledge  
AODV - Ad hoc On Demand Distance Vector  
API - Application Programming Interface  
BC – Basic Clock  
B-MAC - Berkeley - Medium Access Control  
CCA - Clear Access Channel  
CSMA - Carrier Sense Multiple Access  
CSMA/CA - Carrier Sense Multiple Access/Collision Avoidance  
CTS - Clear To Send  
DCO - Digitally Controlled Oscillator  
DRD – Data Rate Detection  
DRAND - Distributed Randomized  
TDMA Scheduling for Sensor Networks  
LEACH - Low-Energy Adaptive Clustering Hierarchy  
LPM – Lower Power Mode  
MAC - Medium Access Control  
MANET – Mobile Ad-hoc Network  
MCLK – Main Clock  
MCU - Microcontroller Unit  
MH-MAC - Multimode Hybrid - Medium Access Control  
MMH-MAC – Mobile Multimode Hybrid - Medium Access Control  
MobH-MAC - Mobility Adaptive Hybrid Protocol  
MS-MAC – Medium Access Protocol for Sensor Networks  
LMAC - Lightweight Medium Access Protocol  
LL-MAC - Low Latency Medium Access Protocol  
LPL - Low Power Listening

---

OLSR - Optimized Link State Routing  
OSPF - Open Shortest Path First  
PACK – Preamble Acknowledge  
RSSI - Received Signal Strength Indication  
RTS - Request To Send  
RTCM – Rede Temática de Comunicações Móveis  
S-MAC - Sensor - Medium Access Control  
SCP-MAC - Scheduled Channel Polling – Medium Access Control  
SMCLK – Sub-Main Clock  
SRBit - Slot Reserve Bit  
T-MAC - Timeout - Medium Access Control  
TDMA - Time Division Multiple Access  
Z-MAC - Zebra Medium Access Control

---

# ÍNDICE DE MATÉRIAS

---

|  |    |
|--|----|
| Capítulo 1. - Introdução .....                       | 1  |
| 1.1. Introdução .....                                | 1  |
| 1.2. Enquadramento.....                              | 2  |
| 1.3. Hipótese .....                                  | 3  |
| 1.4. Objectivos e Contribuições .....                | 4  |
| 1.5. Estrutura da Dissertação .....                  | 4  |
| Capítulo 2. - Trabalho relacionado.....              | 7  |
| 2.1. Redes de Sensores Sem Fios .....                | 7  |
| 2.2. Componentes de Software/Hardware .....          | 9  |
| 2.3. Nível MAC para Redes de Sensores sem Fios ..... | 9  |
| 2.3.1. Aproximação Assíncrona.....                   | 10 |
| 2.3.1.1. B-MAC.....                                  | 10 |
| 2.3.1.2. X-MAC .....                                 | 11 |
| 2.3.1.3. WiseMAC .....                               | 13 |
| 2.3.2. SyncWUF .....                                 | 14 |
| 2.3.3. Abordagem Síncrona.....                       | 16 |
| 2.3.3.1. S-MAC .....                                 | 16 |
| 2.3.3.2. T-MAC .....                                 | 18 |
| 2.3.3.3. SCP-MAC .....                               | 18 |
| 2.3.3.4. MS-MAC.....                                 | 20 |
| 2.3.3.5. LL-MAC.....                                 | 21 |
| 2.3.3.6. LEACH.....                                  | 24 |
| 2.3.4. Abordagem Híbrida.....                        | 25 |
| 2.3.4.1. Z-MAC .....                                 | 25 |
| 2.3.4.2. MH-MAC .....                                | 26 |
| 2.3.4.3. MobH-MAC .....                              | 28 |
| 2.3.4.4. <i>Funneling</i> -MAC .....                 | 30 |
| 2.4. Protocolos de Encaminhamento .....              | 31 |
| 2.4.1. ABR.....                                      | 32 |

---

|   |  |    |
|---|--|----|
| 2.4.2.  | OLSR .....   | 32 |
| 2.4.3.  | Conclusões .....   | 34 |
| 2.5.  | Aplicações das Redes de Sensores sem Fios .....                      | 34 |
| Capítulo 3. - Arquitectura Do Protocolo ..... |  | 35 |
| 3.1.  | Introdução .....   | 35 |
| 3.2.  | Estrutura do protocolo MMH-MAC .....                                 | 37 |
| 3.2.1.  | Modo Assíncrono .....  | 38 |
| 3.2.2.  | Modo Síncrono.....   | 46 |
| 3.2.2.1.                                      | Comunicação .....  | 46 |
| 3.2.2.2.                                      | Modelo.....  | 47 |
| 3.3.  | Funcionamento do Protocolo .....                                     | 49 |
| 3.3.1.  | Estabelecimento inicial de sincronismo entre nós Assíncronos.....    | 50 |
| 3.3.1.1.                                      | Protocolo de sincronização .....                                     | 50 |
| 3.3.2.  | Transição Assíncrono-Síncrono.....                                   | 56 |
| 3.3.2.1.                                      | Modelo Analítico para o Tempo de Associação .....                    | 60 |
| 3.3.3.  | Transição Síncrono - Assíncrono .....                                | 63 |
| 3.3.4.  | Gestão de Clusters .....   | 65 |
| 3.3.5.  | Manutenção do Sincronismo entre os nós.....                          | 66 |
| 3.4.  | Interface de programação da aplicação.....                           | 68 |
| Capítulo 4. - Análise Do Desempenho .....     |  | 73 |
| 4.1.  | Ambiente de Teste .....  | 73 |
| 4.1.1.  | Simulador TOSSIM .....   | 74 |
| 4.1.1.1.                                      | Suporte de Mobilidade no simulador TOSSIM .....                      | 76 |
| 4.2.  | Pontos de Análise .....  | 79 |
| 4.2.1.  | Consumo de Energia .....   | 79 |
| 4.2.2.  | Tempo de transição Assíncrono-Síncrono .....                         | 80 |
| 4.2.3.  | Débito.....  | 81 |
| 4.3.  | Resultados .....   | 82 |
| 4.3.1.  | Recolha de dados das simulações.....                                 | 82 |
| 4.3.2.  | Modelo de consumo energético nos Dispositivos TelosB utilizado ..... | 83 |
| 4.3.3.  | Desempenho do Protocolo .....  | 86 |
| 4.3.3.1.                                      | Tempo de Transição Assíncrono – Síncrono.....                        | 86 |

---

|   |     |
|---|-----|
| 4.3.3.2. Débito .....                             | 90  |
| 4.3.3.3. Consumo Energético.....                  | 95  |
| 4.4. Conclusões.....                              | 99  |
| Capítulo 5. - Conclusões.....                     | 101 |
| 5.1. Síntese Geral .....                          | 101 |
| 5.2. Conclusões.....                              | 102 |
| 5.3. Trabalho Futuro.....                         | 104 |
| Bibliografia .....                                | 105 |
| Apêndice A. - Mobilidade no simulador TOSSIM..... | 111 |

---

---

## ÍNDICE DE FIGURAS

---

|   |    |
|---|----|
| Figura 2. 1 - Comparação entre os protocolos B-MAC e X-MAC .....  | 12 |
| Figura 2. 2 – Transmissão de dados no protocolo WiseMAC. ....   | 13 |
| Figura 2. 3 - Descrição de um período de funcionamento do protocolo S-MAC. ....   | 17 |
| Figura 2. 4 - Transmissão de dados no protocolo SCP-MAC.....  | 19 |
| Figura 2. 5 - Protocolo MS-MAC, ciclos onde o nó permanece a dormir e ciclos de escuta ...  | 20 |
| Figura 2. 6 - Intervalos de Controlo e de Transmissão de dados no protocolo LL-MAC .....  | 23 |
| Figura 2. 7 - Transmissão de dados no protocolo MH-MAC.....   | 26 |
|   |    |
| Figura 3. 1 - Transmissão unicast no protocolo MMH-MAC.....   | 39 |
| Figura 3. 2 - Transmissão broadcast no protocolo MMH-MAC .....  | 39 |
| Figura 3. 4 - Transmissão de dados entre nós assíncronos – procedimentos relativos ao emissor.....  | 44 |
| Figura 3. 5 - Transmissão de dados entre nós assíncronos – procedimentos relativos ao receptor.....   | 45 |
| Figura 3. 6 - Estrutura do slot público no protocolo MMH-MAC .....  | 48 |
| Figura 3. 7 - Exemplo de estabelecimento inicial de Sincronismo entre nós assíncronos .....   | 52 |
| Figura 3. 8 – Procedimentos relativos ao nó que desencadeia o processo de sincronização.  | 55 |
| Figura 3. 9 – Procedimentos associados aos nós vizinhos do nó que origina o processo de sincronização.....  | 55 |
| Figura 3. 10 - Processo de Sincronização quando o nó fixo se encontra síncrono.....   | 58 |
| Figura 3. 11 – Transição Assíncrono-Síncrono caso o nó fixo seja síncrono, procedimentos realizados pelo nó assíncrono que desencadeia a sincronização.....                                 | 59 |
| Figura 3. 12 - Transição Assíncrono-Síncrono caso o nó fixo seja síncrono, comportamento de um nó receptor dos preâmbulos enviados pelo nó que desencadeia o processo de sincronização..... | 60 |
| Figura 3. 13 – Transição Síncrono-Assíncrono .....  | 65 |
| Figura 3. 14 - Comandos e Eventos utilizados no estabelecimento de sincronismo entre os nós.....  | 71 |

---

|  |     |
|--|-----|
| Figura 4. 1 - – Procedimentos relativos ao suporte para mobilidade dos nós no simulador TOSSIM.....                      | 78  |
| Figura 4. 2 - Percurso realizado pelo nó móvel M ao longo dos seis clusters .....  | 81  |
| Figura 4. 3 – Tempo de obtenção de um slot dedicado vs Número de slots alocados – sem slot de escuta .....               | 87  |
| Figura 4. 4 – Tempo de obtenção de um slot dedicado vs Número de slots alocados – com slot de escuta .....               | 88  |
| Figura 4. 5 – Comparação dos diversos tempos até à obtenção de um slot dedicado .....                                    | 89  |
| Figura 4. 6 – Percurso realizado pelo nó móvel no cenário considerado.....   | 90  |
| Figura 4. 7 – Débito vs Tempo de permanência para Assíncrono-Assíncrono e Assíncrono-Síncrono. ....                      | 93  |
| Figura 4. 8 - Débito Obtido na presença de vários emissores. A-A: Assíncrono-Assíncrono e A-S: Assíncrono-Síncrono. .... | 94  |
| Figura 4. 9 – Consumo energético no caso em que os nós permanecem assíncronos.....                                       | 96  |
| Figura 4. 10 – Consumo energético, ocorrendo o estabelecimento de sincronismo entre o nó móvel e o fixo.....             | 96  |
| Figura 4. 11 – Energia consumida pelos nós estáticos e nós móveis, em função do número de <i>slots</i> alocados.....     | 98  |
| <br>   |     |
| Figura A. 1 - Estrutura do Ficheiro de texto usado para definir a mobilidade na rede .....                               | 111 |

---

## ÍNDICE DE TABELAS

---

|  |    |
|--|----|
| Tabela 3. 1 – Comandos e eventos relativos à API do protocolo MMH-MAC.....   | 70 |
| Tabela 4. 1 - Características dos modos LPM existentes nos sensores TelosB .....   | 84 |
| Tabela 4. 2 - Consumo de corrente do TelosB nos vários modos de LPM, encontrando-se o dispositivo de rádio desligado ..... | 84 |
| Tabela 4. 3 – Consumos médios do MCU, juntamente com o rádio, nos vários modos LPM.  | 85 |
| Tabela 4. 4 - Resumo dos consumos médio do MCU mais o dispositivo flash .....  | 85 |
| Tabela 4. 5 - Consumos médios do MCU mais os sensores.....   | 86 |

---

# Capítulo 1.

## Introdução

---

### 1.1. Introdução

A tecnologia apresentou nos últimos anos um desenvolvimento notável, revolucionando o quotidiano e na maioria das situações contribuindo para o benefício do bem-estar da população em geral. Esta evolução tem-se igualmente verificado nos componentes electrónicos, permitindo a integração de hardware em pequenos dispositivos, com uma *interface* de rede, uma razoável capacidade de execução e um baixo custo. A existência destes dispositivos constitui o ponto de partida para o aparecimento das redes de sensores sem fios.

As redes de sensores sem fios são constituídas por nós sensores constituídos fisicamente por uma bateria, unidade de processamento, memória, sensores e unidade de comunicação. Além disso, estes elementos sensores possuem capacidade para monitorar e recolher informação acerca de fenómenos físicos, como a humidade, pressão, luminosidade, movimentos e temperatura.

Nestas redes sem fios, esses dispositivos autónomos espacialmente distribuídos comunicam entre si providos de autonomia. Assim, é possível recolher dados através dos sensores, processá-los localmente e transmitir a informação após o processamento, para um conjunto limitado de nós colectores.

Nas redes de sensores sem fios deve existir uma especial preocupação com as restrições energéticas. Os elementos sensores, com uma bateria de curta longevidade, ao

serem distribuídos numa determinada área geralmente não possuem intervenção humana. Ou seja, não há a possibilidade de recarga.

Por outro lado, além desses nós estáticos surge ainda a possibilidade de existirem nós móveis numa rede de sensores sem fios, os quais aumentam o desafio inerente aos protocolos e arquitecturas desenvolvidas de modo a otimizar o desempenho do sistema, mantendo um reduzido consumo energético.

Na presente dissertação procede-se à discussão da problemática relativa aos protocolos de controlo de acesso ao meio (MAC) aptos a otimizar a relação custo/desempenho em aplicações para redes de sensores sem fios na presença de mobilidade dos nós. Sucintamente, procede-se à análise da capacidade de gestão da coexistência de nós estáticos e móveis numa rede, assim como a manutenção do reduzido consumo energético e elevado débito na recolha de dados.

O principal contributo desta dissertação consiste no desenvolvimento de um protocolo de acesso ao meio com suporte para mobilidade em redes de sensores sem fios, possuindo a versatilidade de ser controlado pela aplicação através de diversos parâmetros.

## **1.2. Enquadramento**

Ao longo dos últimos anos as áreas de aplicação das redes de sensores sem fios sofreram uma considerável extensão, passando a ser aplicadas em situações do quotidiano como a prevenção de incêndios, a agricultura, a medicina e a monitorização ambiental.

No vasto leque de aplicações das rede de sensores sem fios, os requisitos a cumprir variam entre as aplicações. No entanto, existem características comuns à totalidade das aplicações, tais como: facilidade de implantação, cooperação com nós vizinhos, não requerer manutenção humana, longa autonomia e reduzido custo.

Estas redes permitem a monitorização de áreas de difícil acesso, durante um elevado período de tempo, graças ao seu reduzido consumo energético. Por outro, permitem a existência de nós móveis (e. g. localizados em automóveis) que se deslocam recolhendo informação sensorial.

Devido a essa crescente aplicabilidade das redes de sensores sem fios tornou-se indispensável o desenvolvimento de novos protocolos, cujas características permitam dar uma resposta otimizada aos requisitos de cada aplicação.

Ao longo do tempo foram desenvolvidos alguns protocolos de acesso ao meio específicos para suporte de mobilidade em redes de sensores sem fios, onde se procura minimizar o consumo energético mas cujo desempenho releva-se insuficiente de modo a cumprir os requisitos impostos relativamente a diversos pontos de análise (e. g. débito). Por outro lado, no domínio das MANET foram desenvolvidos protocolos otimizados para a gestão de redes móveis mas descurando a redução do consumo energético, já que os elementos constituintes dessas redes não apresentam limitações tão significativas ao nível das fontes de alimentação.

Nesta dissertação mostra-se que é possível conjugar “o melhor dos dois mundos”, isto é, aplicar os conceitos base inerentes aos protocolos desenvolvidos no domínio das MANET, em redes de sensores sem fios móveis. O protocolo proposto tem um desempenho elevado para cenários de mobilidade mas mantém um reduzido consumo energético, indispensável numa rede de sensores sem fios.

### **1.3. Hipótese**

É possível a coexistência entre nós sensores móveis e nós sensores estáticos, onde os nós estáticos têm perturbações a nível de débito e energéticas desprezáveis, e os nós móveis podem usufruir de desempenhos comparáveis aos nós estáticos, para tempos de permanência de ligações da ordem de grandeza de centenas de milissegundos.

A principal viabilidade de coexistirem os dois tipos de nós consiste na possibilidade de os nós estáticos procederem à monitorização do local onde foram colocados e os nós móveis deslocarem-se entre nós estáticos recolhendo os dados.

Por exemplo, um cenário no qual seria viável a aplicação do protocolo desenvolvido, consiste numa floresta onde nós estáticos realizam a monitorização de parâmetros característicos do ambiente (e. g. temperatura ou humidade) e nós móveis procedem à recolha dos dados resultantes desse processo. Estes nós móveis podem por exemplo, estar localizados em veículos dos bombeiros ou de entidades ligadas à conservação da natureza.

## **1.4. Objectivos e Contribuições**

O principal objectivo desta dissertação é validar a hipótese apresentada na secção anterior e comprovar a viabilidade da junção entre os conceitos base referentes às MANET e às redes de sensores sem fios, num único protocolo. Estes conceitos base consistem na gestão de mobilidade no caso das MANET e na redução do consumo energético no que respeita às redes de sensores sem fios.

Pretende-se provar e validar a possibilidade de ter um sistema constituído por ilhas de nós sensores estáticos com capacidade de monitorizar o ambiente envolvente, existindo nessa rede outros nós móveis que procedem à recolha dos dados armazenados, podendo trocar informação entre si.

O principal contributo desta dissertação consiste na definição de um protocolo de nível MAC com uma abordagem inovadora nos aspectos relativos ao suporte de mobilidade para redes de sensores sem fios. Realiza-se o estudo do desempenho do protocolo para os diferentes modos de funcionamento, relativamente a métricas como o débito e o consumo energético.

O trabalho desenvolvido conduziu à realização de uma apresentação no 8º seminário da Rede Temática de Comunicações Móveis (RTCM) [Água 2008].

## **1.5. Estrutura da Dissertação**

A dissertação encontra-se organizada em cinco capítulos e um anexo, conforme se resume em seguida.

No Capítulo 2 (“Trabalho Relacionado”) efectua-se uma análise dos protocolos de nível MAC existentes para redes de sensores sem fios. Em cada ponto efectua-se uma análise crítica acerca desses protocolos, a qual se foca sobretudo na capacidade de suportar mobilidade dos nós. Além disso, analisam-se alguns protocolos desenvolvidos no domínio das MANET, optimizados para gerir a mobilidade numa determinada rede mas não para reduzir o consumo energético. Por último, mencionam-se de forma sucinta as principais aplicações das redes de sensores sem fios, na actualidade.

No Capítulo 3 (“Arquitectura do protocolo”) é proposto o modelo de um protocolo de acesso ao meio para redes de sensores sem fios, cujo comportamento se encontra

otimizado para suportar a mobilidade dos nós. Procede-se à apresentação e um modelo matemático relativo ao comportamento do sistema num dos pontos de análise. Por último, realiza-se uma descrição da interface *cross layer* que permite à aplicação ajustar o comportamento do nível MAC de acordo com as suas necessidades.

No Capítulo 4 (“Análise do Desempenho”), procede-se inicialmente à descrição do ambiente de teste utilizado para analisar o desempenho do protocolo, assim como dos componentes utilizados. Posteriormente, descrevem-se os pontos de análise e realiza-se a análise do desempenho do protocolo proposto nesta dissertação.

No Capítulo 5 (“Conclusões”), efectua-se uma análise global do trabalho realizado, com base na hipótese originalmente estabelecida. Por último, identificam-se as principais contribuições deste trabalho, assim como as questões em aberto e consequentes trabalhos futuros.

No Apêndice A (“Mobilidade no simulador TOSSIM”) é apresentada de forma pormenorizada a extensão realizada no simulador TOSSIM para suportar mobilidade, e a sua parametrização.



## Capítulo 2.

# Trabalho relacionado

---

Nos últimos anos verificou-se uma considerável evolução qualitativa e quantitativa dos projectos de investigação na área das redes de sensores sem fios. Neste capítulo procede-se à descrição dos principais aspectos relacionados com a investigação em redes de sensores sem fio. O capítulo está dividido em quatro secções. Na primeira secção faz-se uma introdução sobre as redes de sensores sem fios e as componentes utilizadas no desenvolvimento de projectos neste domínio. Na segunda secção são abordados diversos protocolos de nível MAC desenvolvidos para redes de sensores sem fios. Para cada caso procura-se analisar as suas vantagens e desvantagens. Além disso, no final desta secção procede-se a uma análise acerca do comportamento de cada protocolo, na presença de mobilidade dos nós. Na terceira secção abordam-se diversos protocolos de encaminhamento em redes de sensores sem fios. Por último, na quarta secção descrevem-se algumas das principais aplicações das redes de sensores sem fios.

### **2.1. Redes de Sensores Sem Fios**

As redes de sensores sem fios constituem uma tecnologia emergente, fulcral para o desenvolvimento das possibilidades de interacção remotas de um utilizador com o meio físico.

Uma rede de sensores sem fios apresenta na sua constituição um conjunto de nós autónomos e geralmente densamente distribuídos, com capacidade de trocar informação entre si através de ligações sem fios.

A eliminação dos custos associados à interligação através de cabos, a possibilidade de serem colocadas em zonas de difícil acesso (colocadas com recurso a um helicóptero por exemplo), a flexibilidade e a portabilidade são alguns das vantagens que tornam esta tecnologia alvo de investigadores e empresários. No entanto, o desenvolvimento de aplicações utilizando as redes de sensores sem fios leva à necessidade de ultrapassar entraves relacionados com a limitação de recursos como energia, processamento e memória.

A mobilidade de nós em Redes de Sensores sem Fios (por exemplo em veículos automóveis) constitui um desafio para as aplicações desenvolvidas, sobretudo no que respeita à manutenção da capacidade de transmitir rajadas de tráfego espontâneas, entre diversos nós, mantendo otimizados parâmetros como o consumo energético e o *throughput* (débito).

Nos últimos anos diversas abordagens foram realizadas acerca de protocolos para suporte de mobilidade em MANET (Redes Ad Hoc Móveis), assim como algoritmos para redução do consumo energético no domínio das redes de sensores sem fios. No entanto, na área das redes de sensores sem fios verifica-se a quase ausência de trabalhos que conjuguem os dois aspectos.

No que diz respeito aos cenários onde se considera ideal a utilização de uma rede de sensores sem fios, considere-se uma floresta na qual os sensores dispostos no terreno são utilizados para efectuar a monitorização de diversos factores como a temperatura e a humidade. Nós móveis, por exemplo transportados por vigilantes da natureza ou bombeiros, procedem à recolha dos dados de modo a proceder à análise do estado actual do ambiente envolvente, podendo ou não existir troca de informações entre esses nós móveis. Assim, neste cenário coexistem nós estáticos e móveis, o que constitui um dos grandes focos actuais em termos dos projectos de investigação realizados em redes de sensores sem fios, nos quais se incluem os protocolos de nível MAC de seguida abordados.

## 2.2. Componentes de Software/Hardware

Durante o desenvolvimento de projectos no domínio das redes de sensores sem fios são utilizadas duas componentes distintas: *software* e *hardware*.

O hardware diz respeito aos dispositivos que constituem os diferentes nós de uma rede, os quais apresentam na sua constituição elementos como: sensores (por exemplo: temperatura, humidade, luminosidade), unidade de processamento, unidade de memória e dispositivo rádio para emissão-recepção. Alguns dos dispositivos utilizados são TelosB [Crossbow 2007a] [Polastre 2005], Mica2 [Crossbow 2007b], Micaz [Crossbow 2007c], SmartDust [Warneke 2003], entre outros.

Tendo em conta os diversos tipos de *hardware* existentes tornou-se imprescindível o desenvolvimento de sistemas operativos compatíveis com os baixos recursos deste género de *hardware*.

Neste âmbito o TinyOs-2.x [Levis 2005a] [Levis 2005b] [Levis 2006] [Levis 2007] apresenta-se como um dos sistemas operativos de maior importância em redes de sensores sem fios. O TinyOS utiliza a linguagem NesC [Gay 2003], tendo como parte integrante o simulador TOSSIM [Levis 2003a] [Levis 2003b] que corre o código nativo usado nos dispositivos.

## 2.3. Nível MAC para Redes de Sensores sem Fios

Numa rede de sensores sem fios o protocolo de controlo de acesso ao meio, *Medium Access Control* (MAC), especifica um conjunto de técnicas que permite aos nós pertencentes à rede, a partilha do meio através da definição de um procedimento para a transmissão de dados. Tendo em conta as características inerentes a uma rede de sensores sem fios na presença de mobilidade e à limitação energética dos nós, os protocolos tradicionais para controlo de acesso ao meio revelam-se inadequados.

Com o intuito de diminuir os consumos energéticos dos nós, verifica-se a existência de um período de adormecimento, seguido de um intervalo de tempo onde os nós se encontram activos, ou seja, um determinado *duty-cycle*. Consequentemente, um dos grandes desafios de um protocolo MAC consiste em manter um *throughput* elevado, um

atraso reduzido e uma elevada eficiência energética, na presença do *duty-cycle* dos diversos nós.

Os protocolos de nível MAC podem seguir uma abordagem assíncrona onde os *duty-cycles* dos nós da rede são absolutamente independentes, ou uma síncrona onde os nós transitam para activos ou adormecidos de forma coordenada no tempo. O terceiro grande conjunto de protocolos MAC consiste em modelos híbridos onde os nós transitam entre modos assíncronos e/ou diversas formas de modos síncronos de forma a usufruir das vantagens de várias abordagens.

Nas subsecções seguintes são abordados os protocolos de acesso ao meio de maior relevo no domínio das redes de sensores sem fios, comentando-se o seu funcionamento na presença de mobilidade.

### 2.3.1. Aproximação Assíncrona

#### 2.3.1.1. B-MAC

Recorrendo a uma interface flexível para configuração dos diversos parâmetros, o protocolo B-MAC (Berkeley-MAC) [Polastre 2004], apresenta como principais objectivos a existência de um reduzido consumo energético, ausência de colisões e uma elevada eficácia do canal.

Uma das inovações deste protocolo consistiu no mecanismo de *Clear Channel Assessment* (CCA). No CCA verifica-se a ocupação do canal através da amostragem da potência do sinal, sendo o resultado obtido comparado com um limiar predefinido acima do qual se conclui se o canal está ocupado. Caso o canal se encontre ocupado com determinada transmissão, o nó recorre a um mecanismo de *backoff*, através do qual espera um determinado intervalo de tempo (definido pela interface), antecedendo a repetição do mecanismo de CCA.

Estes mecanismos são usados no protocolo B-MAC, para definir um processo de escuta designado por *Low Power Listening* (LPL), caracterizado por possibilitar a transição dos nós entre os estados de adormecido e activo.

No que diz respeito à transmissão de dados, os dados são antecidos por um preâmbulo com duração superior ao período de adormecimento, com o intuito de o nó receptor ser informado da iminente transmissão de dados. Nas situações onde são enviados

dados, o nó receptor (tal como o emissor) permanece activo de modo a que a transferência de dados possa ser processada. Outro aspecto a realçar consiste no envio de uma mensagem de confirmação (ACK) após a recepção de pacotes *unicast*.

No entanto, o modo de funcionamento do protocolo B-MAC apresenta diversas desvantagens, enumeradas de seguida:

- Excessivo consumo de energia por parte do emissor e do receptor para envio frequente de dados: dada a longa duração do preâmbulo enviado aquando da necessidade de transmitir dados por parte de um nó, o emissor irá consumir um considerável valor de energia, sendo tal situação idêntica para o caso do receptor.
- Energia consumida devido à existência de *overhearing* (escuta por nós não receptores): caso um nó acorde durante a transmissão de um preâmbulo, independentemente de ser ou não o destino dos dados, irá manter-se acordado até ao final da duração do preâmbulo.
- Aumento da latência com o número de *hops* (saltos): à medida que o número de *hops* entre o emissor e o receptor aumenta, ocorre um incremento do tempo de espera no destino, já que mais preâmbulos de longa duração são transmitidos antes dos dados.

### 2.3.1.2. X-MAC

O protocolo X-MAC [Buettner 2006] surgiu sobretudo com o intuito de eliminar alguns dos problemas anteriormente referidos para o caso do B-MAC. De modo a diminuir o consumo de energia do emissor e receptor dos dados devido aos longos preâmbulos, foi introduzida a transmissão consecutiva de curtos preâmbulos de tamanho fixo.

Por outro lado, para resolver o problema relacionado com a ocorrência de *overhearing*, o X-MAC propõe a integração do endereço do destino dos dados, no preâmbulo. Entende-se por *overhearing* o facto de o nó receptor acordar e permanecer escutando um longo preâmbulo quando não é o receptor da mensagem a enviar após o preâmbulo.

A inserção do endereço de destino no preâmbulo possibilita ao receptor verificar se é o destino da mensagem, aquando do início da recepção do preâmbulo, e transitar para um estado de adormecido caso não o seja.

Relativamente ao elevado atraso em redes *multi-hop*, foi introduzido um espaçamento temporal entre os pequenos preâmbulos enviados, permitindo ao receptor o envio de uma confirmação (ACK) após a recepção do preâmbulo. Após receber o ACK, o emissor procede ao envio dos dados, verificando-se uma relevante diminuição da latência na transmissão de dados a 1 *hop*, a qual será repercutida nos restantes saltos.

Além dos aspectos acima referidos, o protocolo X-MAC, tem como parte integrante um algoritmo que permite adaptar o *duty-cycle* dos diversos nós, possibilitando uma adaptação contínua do protocolo ao tráfego na rede. Este aspecto reflecte-se num reduzido consumo de energia e latência, apesar da variação da carga na rede. No que diz respeito às desvantagens do protocolo X-MAC, saliente-se a ausência de um método destinado a evitar as colisões de preâmbulos aquando da existência de vários emissores simultâneos. Por não usar o mecanismo de CCA, um receptor pode não acordar se houver colisão de preâmbulos.

A figura seguinte ilustra os principais aspectos relacionados com os protocolos B-MAC e X-MAC, possibilitando a comparação das diferenças entre ambos.

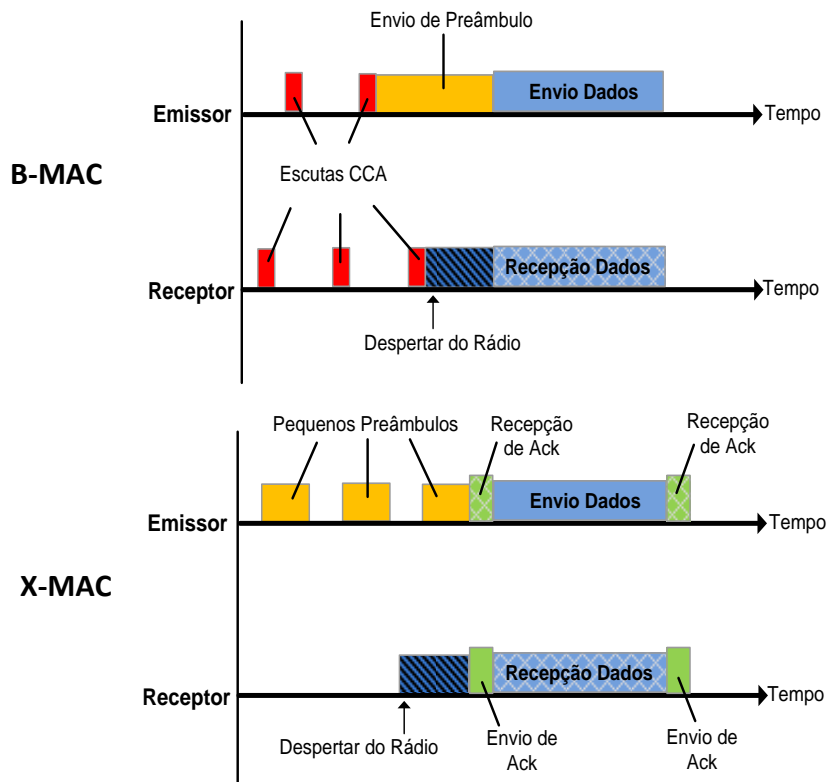


Figura 2. 1 - Comparação entre os protocolos B-MAC e X-MAC

Relativamente ao desempenho na presença de mobilidade, tanto o X-MAC como o B-MAC não apresentam degradação devido a serem intrinsecamente assíncronos, embora

tenham limitações a nível de *throughput* (especialmente para múltiplas transmissões concorrentes). Outro problema está relacionado com o aumento do número de retransmissões em *unicast* por parte de um nó emissor, já que caso o receptor se mova para o exterior da sua vizinhança, após o envio dos preâmbulos, o emissor não recebe o ACK, repetindo o envio dos preâmbulos (sucessivamente) mas de forma inconsequente já que o receptor não os receberá.

### 2.3.1.3. WiseMAC

O protocolo WiseMAC [El-Hoiydi 2003] baseia-se no mecanismo de CSMA (*Carrier Sense Multiple Access*) e recorre à amostragem do meio para minimizar o consumo de energia.

Uma das inovações deste protocolo face ao B-MAC consiste na memorização do instante de amostragem do meio nos vizinhos directos. Conhecendo-se o instante em que o vizinho acorda para realizar a amostragem do meio, o nó pode enviar o preâmbulo ligeiramente antes desse momento, possibilitando uma reduzida dimensão do mesmo. Este aspecto encontra-se ilustrado na Figura 2.2.

Esta abordagem reduz a energia consumida devido à transmissão e recepção de mensagens, assim como a causada pela existência de *overhearing*. Por outro lado, esquemas de *backoff* e reserva do meio são utilizados para melhorar o desempenho em termos de *fairness* (justiça no acesso ao meio) e redução de colisões.

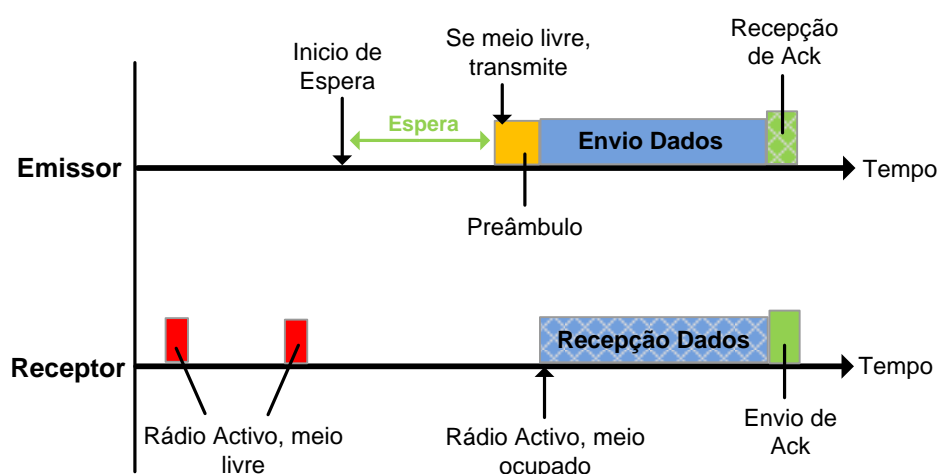


Figura 2. 2 – Transmissão de dados no protocolo WiseMAC.

Na presença de mobilidade dos nós verifica-se a ocorrência de uma considerável perda de eficiência devido à possibilidade de a vizinhança mudar frequentemente, tendendo o comportamento do WiseMAC para o B-MAC com mobilidade elevada.

### 2.3.2. SyncWUF

O protocolo SyncWUF [Shi 2007] caracteriza-se pela utilização alternativa de uma de duas metodologias baseadas num processo de amostragem do meio: um esquema de *Wake-Up-Frame* (WUF) ou o protocolo WiseMAC, aqui designado de *Wake-Up-Preamble* (WUP). Esta combinação realiza-se com o intuito de aumentar o tempo de vida das baterias, sem afectar parâmetros como a capacidade do canal ou a latência.

O processo de amostragem consiste num método baseado no algoritmo CSMA para enviar pacotes de dados para determinado destino. Isto é, antecedendo cada pacote, o emissor envia um preâmbulo com um tamanho de pelo menos um ciclo de *duty cycle*, de modo a acordar o receptor.

A potência de emissão do preâmbulo destinado a acordar o receptor com WUP (*wake-up-preamble*) é detectada através da amostragem efectuada pelo nó usando um detector do nível de RSSI (*Received Signal Strength Indication*). Por outro lado, num esquema WUF (*wake-up frame*), é utilizado um conjunto de várias *frames* (tramas) de dimensão reduzida SWUF (*short wake-up frame*), as quais incluem o endereço MAC do destino e um campo que indica a sua posição no interior da WUF.

Deste modo, caso um nó seja acordado devido à recepção de uma SWUF completa, este pode decidir se está perante uma situação de *overhearing*, através da comparação do seu endereço MAC com o presente na mensagem recebida, transitando para um estado de adormecido caso não se verifique a equidade dos mesmos. Pelo contrário, se for o destino da mensagem, este pode calcular o tempo de espera até ao início da transmissão dos dados, recorrendo à posição da SWUF indicada nessa mensagem e adormecendo no intervalo de tempo que decorre até a esse instante.

No protocolo SyncWUF, os emissores armazenam os esquemas temporais relativos ao funcionamento dos receptores (*schedules*) dos receptores e adaptam o esquema para acordar o receptor (WUF ou WUP) de acordo com esse valor. Considerando  $\theta$  a tolerância

do relógio do nó e  $T$  o tempo decorrido desde o último ACK recebido de um determinado nó, então o comprimento de um pacote WUP é dado por  $4\theta T$ .

Assim, se o intervalo de tempo decorrido desde a recepção do último *schedule* do receptor é reduzido (i.e.  $T$  é pequeno), então a dimensão do WUS  $4\theta T$  é suficientemente pequena, sendo usado um mecanismo WUP tal como se verifica no protocolo WiseMAC.

Pelo contrário, se é verificada a existência de um *schedule* desatualizado, então são usadas diversas tramas SWUF para reduzir o tempo de espera desnecessário (e consequentemente o consumo de bateria), recorrendo-se ao esquema WUF anteriormente analisado. Este aspecto permite uma redução do tempo de espera, em média, de 1.5 vezes o tempo de uma SWUF. Considerando que a duração de uma SWUF não ultrapassa um número reduzido de milissegundos e comparando com as centenas de milissegundos inerentes ao método WUP, conclui-se a existência de uma considerável diminuição do tempo de espera desnecessário.

Saliente-se o facto de os nós evitarem casos de acordarem devido a ruído, já que podem obter a potência do sinal recebido WUS (wake-up-signal), assim como o ritmo de transmissão do sinal, utilizando um método de *Data Rate Detection* (DRD).

Apesar de o esquema WUF reduzir a energia consumida, minimizando o tempo durante o qual um nó permanece activo à espera de mensagens, não é usado nenhum mecanismo para aumentar a capacidade do canal. Note-se o facto de a reduzida capacidade do canal se dever sobretudo ao elevado tamanho do preâmbulo WUS (*wake-up-signal*).

No protocolo WiseMAC ocorre uma diminuição da dimensão do preâmbulo WUS, o que implica um aumento da capacidade do canal. No entanto, este facto poderá ter consequências drásticas para um coerente funcionamento do protocolo, já que seria a mensagem ideal para transportar informações úteis para tratar aspectos como *drifts* de relógio, mobilidade dos dispositivos. Esta mensagem poderia, por exemplo, possuir a indicação da referência temporal do emissor, de modo ao receptor ajustar o seu relógio com o do emissor. Diminuindo a dimensão da mesma, este aspecto é inviabilizado.

### 2.3.3. Abordagem Síncrona

#### 2.3.3.1. S-MAC

No protocolo S-MAC (*Sensor-MAC*) [Ye 2004] os nós encontram-se síncronos em determinada referência temporal de modo a acordarem conjuntamente no início de cada *slot*, mantendo um *duty cycle* fixo em cada nó.

Sendo um protocolo com uma abordagem síncrona, torna-se indispensável a manutenção de um minucioso sincronismo entre os períodos em que um nó está adormecido ou activo, o qual poderá ser prejudicado pelo *drift* (desvios) de relógio existentes em cada nó. Com o intuito de tornar o protocolo robusto a esses *drifts*, enviam-se periodicamente pacotes SYNC (Figura 2.3) de reduzida dimensão, os quais incluem o endereço do emissor e o tempo até transitar para o estado de adormecido.

O protocolo S-MAC utiliza um mecanismo de contenção de modo a evitar colisões, semelhante ao definido na norma 802.11 [802.11 1999]. Assim, os pacotes enviados em *broadcast* são antecidos pela escuta do meio do algoritmo CSMA, enquanto as mensagens *unicast* combinam o mencionado algoritmo com uma troca de tramas RTS (*Request To Send*)-CTS (*Clear To Send*)-DADOS-ACK (confirmação) [Bharghavan 1994] entre o emissor e o receptor.

Tal como se verifica na Figura 2.3, o nó emissor envia um pacote RTS e o respectivo receptor responde com o pacote CTS. Se algum nó ouvir o pacote RTS e não o pacote CTS, é-lhe permitido enviar dados para outros nós. Caso um nó ouça o pacote CTS, não lhe é permitido enviar dados durante o tempo de alocação do canal especificado no pacote. Deste modo, o número de colisões diminui drasticamente, principalmente em redes *multi-hop*.

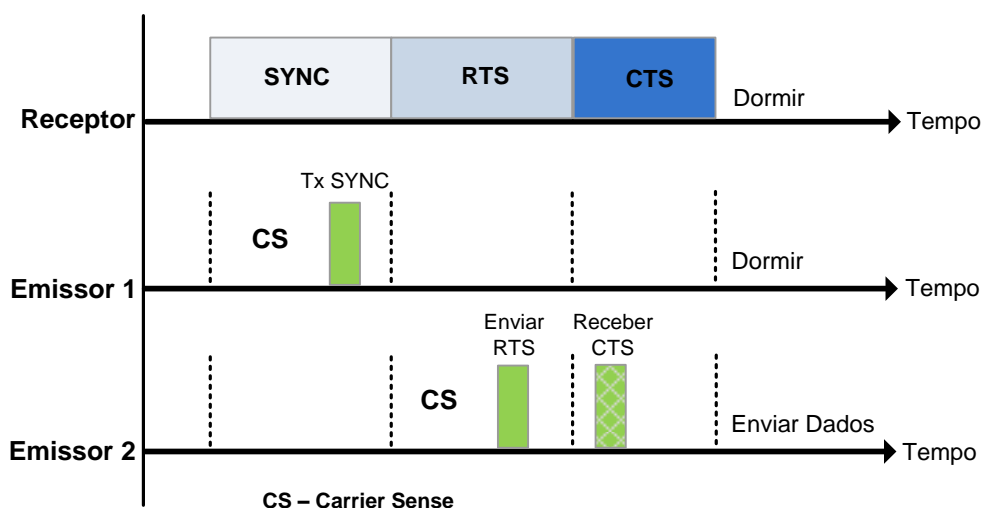


Figura 2. 3 - Descrição de um período de funcionamento do protocolo S-MAC.

Outro aspecto associado ao protocolo S-MAC consiste na inserção em cada pacote de um campo que indica a duração da transmissão em curso. Caso um nó receba um pacote destinado a outro nó, tem conhecimento da duração do intervalo de tempo que deve ficar adormecido devido à transmissão actual. Assim, ocorre uma redução dos gastos excessivos de energia devido ao fenómeno de *overhearing*.

Com o objectivo de evitar o elevado custo imposto pela necessidade de re-transmissão de mensagens longas quando apenas alguns bits são corrompidos, é utilizado um método de *message passing* caracterizado pela fragmentação da uma mensagem de dimensão superior em várias pequenas, as quais são posteriormente transmitidas em rajada. Na técnica acima mencionada, procede-se apenas a uma troca RTS/CTS para reservar o meio, sendo enviado pelo receptor um ACK após a recepção de cada fragmento.

De entre as desvantagens deste protocolo, realça-se o facto de os intervalos de tempo em que o nó permanece activo e adormecido serem constantes e não se adaptarem ao tráfego da rede.

No que diz respeito ao funcionamento do protocolo S-MAC na presença de mobilidade, caso um nó pretenda estabelecer uma ligação com um *cluster* diferente (grupo com outra fonte de sincronização) necessita de esperar por um novo período de sincronização, o qual, como mencionado anteriormente consiste num período de 10 segundos em cada 2 minutos. Este tempo de espera poderá ser crítico para o desempenho da maioria das aplicações, tornando a utilização do protocolo S-MAC numa rede de sensores sem fios inadequada na presença de mobilidade dos nós.

### 2.3.3.2. T-MAC

Apresentando como principal prioridade, colmatar algumas das lacunas do protocolo S-MAC, o protocolo T-MAC (*Timeout-MAC*) [van Dam 2003], introduziu a possibilidade de adaptar o *duty-cycle* com base no tráfego existente na rede. A técnica utilizada consiste em definir uma janela temporal bastante curta em que se está à espera de receber ou enviar pacotes RTS ou CTS, após o período de sincronização. Se não for detectada actividade, o nó regressa ao seu estado adormecido.

Como benefício directo da aplicação desta técnica, o protocolo T-MAC diminui consideravelmente os consumos energéticos quando o tráfego existente na rede é escasso. Este protocolo apresenta como principal desvantagem o facto de existir a possibilidade de um nó transitar para um estado de adormecido quando um outro nó possui mensagens para enviar.

### 2.3.3.3. SCP-MAC

O protocolo *Scheduled Channel Polling MAC* (SCP-MAC) [Ye 2006] combina um esquema de acesso ao meio onde os nós se encontram activos/adormecidos em intervalos predefinidos, com um mecanismo de LPL de modo a reduzir o consumo de energia existente.

Os protocolos síncronos permitem utilizações de *duty-cycles* bastante baixos, contudo pode haver falha na comunicação caso nenhum pacote seja recebido no curto período actividade, devido a colisão ou ruído no canal.

Por outro lado, os protocolos assíncronos permitem a obtenção de reduzidos consumos através do uso de LPL mas apresentam elevados gastos energéticos aquando da necessidade de transmissão de dados. Este aspecto deve-se à necessidade de transmitir um número considerável de preâmbulos antecedendo os dados.

O protocolo SCP-MAC caracteriza-se pela existência de um funcionamento misto, no qual se usa o sistema de LPL para efectuar escutas ao meio, mas recorre-se a métodos de sincronização de modo a alinhar temporalmente o momento de escuta do canal por todos os nós.

Neste protocolo, quando um nó necessita de enviar dados, procede ao envio de um pacote *wake up* de modo a provocar o despertar do receptor, o qual se prepara para a recepção dos dados. Esta situação encontra-se ilustrada na Figura 2.4.

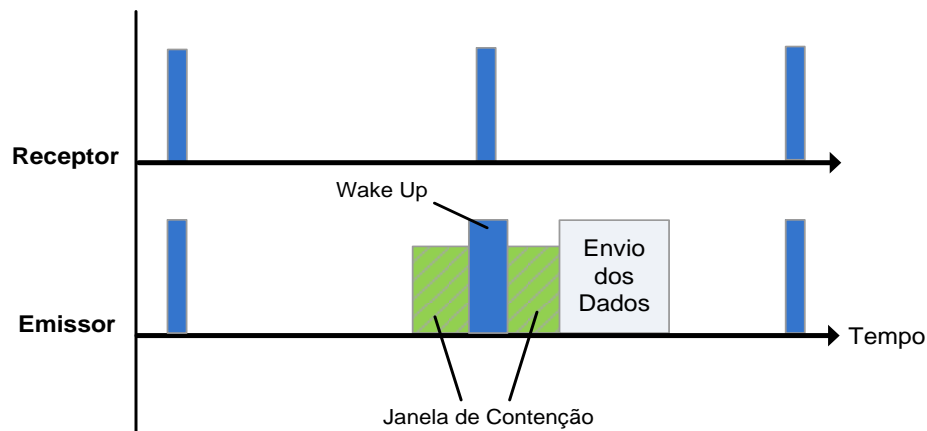


Figura 2. 4 - Transmissão de dados no protocolo SCP-MAC

O receptor apenas verifica o nível de energia no canal, sendo desta forma inconsequentes as colisões entre pacotes *wake up*. Antecedendo o envio do pacote *wake up* executa-se um período de contensão no qual ocorre um processo semelhante ao CSMA. Isto é, espera-se durante um curto intervalo de tempo aleatório, após o qual, caso não ocorra nenhuma transmissão, procede-se ao envio do sinal *wake up*. O valor reduzido deste intervalo de tempo deve-se ao facto de o intervalo de escuta por parte do receptor ser também bastante curto.

Posteriormente ao envio deste sinal que informa o receptor da iminente transmissão dos dados, procede-se a um outro período de contensão após o qual se enviam os dados. Desta forma, reduz-se a probabilidade de colisão no envio dos dados e melhora-se a eficácia do protocolo [Ye 2006], pois pode ter havido mais do que um nó a enviar o sinal *wake up*.

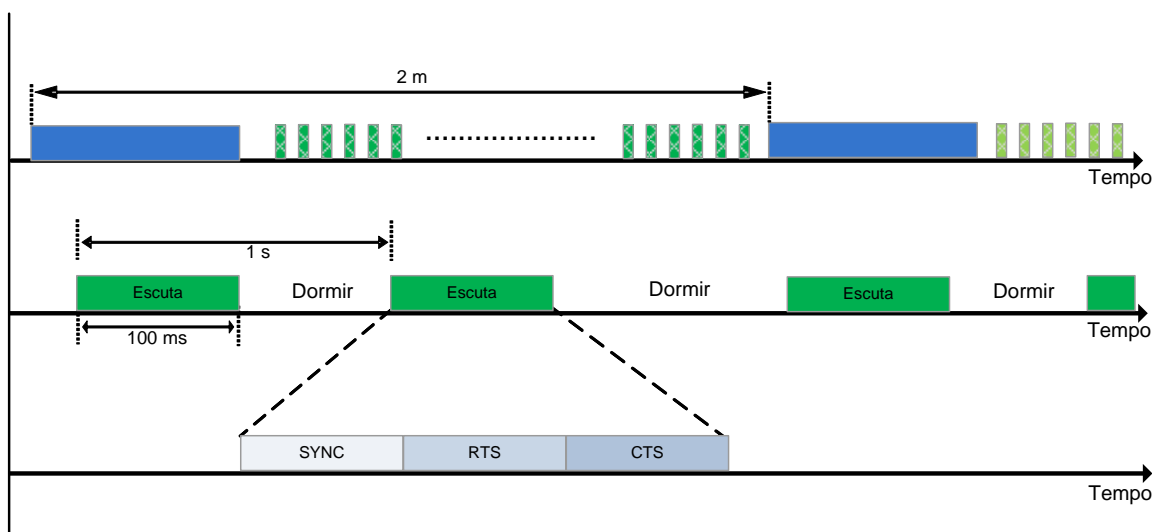
O processo de sincronização poderá ser realizado através da disseminação de pacotes SYNC isolados ou através da agregação dos mesmos com os dados transmitidos, sendo este método designado por *piggybacking*. Com o esquema utilizado neste protocolo verifica-se a existência de *duty-cycles* entre 0,01% a 0,1%, proporcionando consumos energéticos 2 a 2,5 vezes inferiores a protocolos de LPL, como por exemplo o B-MAC. Relativamente às desvantagens do SCP-MAC, destaca-se a elevada precisão necessária no sincronismo devido à curta duração dos períodos de escuta do meio e a existência de valores elevados no que respeita à latência, que tornam difíceis a sua utilização por nós móveis.

### 2.3.3.4. MS-MAC

O protocolo MS-MAC [Pham 2004] apresenta inúmeras semelhanças com o protocolo S-MAC [Ye 2004], na ausência de mobilidade dos nós. Por outro lado, num cenário de mobilidade dos nós da rede, o protocolo transita gradualmente para um modo de funcionamento contínuo, sem adormecimento do nó, com o aumento da velocidade.

O protocolo MS-MAC recorre ao envio periódico de mensagens SYNC para manutenção das ligações e do sincronismo entre vizinhos, assim como para a descoberta de novos vizinhos.

Com o protocolo S-MAC original, caso um nó procure estabelecer uma ligação com um novo vizinho, tem obrigatoriamente de esperar por um novo período de sincronização em que o nó permanece acordado durante o período de *duty cycle*, o qual ocorre durante 10s em cada 2 minutos. Isto é, o nó é forçado a esperar um intervalo de até 2 minutos para estabelecer a ligação e sincronizar com o novo vizinho, tal como se verifica na Figura 2.5.



**Figura 2. 5 - Protocolo MS-MAC, ciclos onde o nó permanece a dormir e ciclos de escuta**

Com o intuito de contornar a desvantagem anterior, o protocolo MS-MAC procede à análise da potência do sinal aquando da recepção de uma mensagem SYNC e com base na variação dessa potência entre várias mensagens consecutivas, conclui acerca da mobilidade do nó emissor e consequentemente, sobre a velocidade relativa de deslocação do emissor face ao receptor.

Caso ocorram alterações no valor das potências entre mensagens consecutivas, a mensagem SYNC passa a conter informação acerca da velocidade dos nós móveis presentes

na vizinhança. Caso existam vários nós móveis na vizinhança, o valor presente na mensagem SYNC refere-se à velocidade máxima entre esses vizinhos. Por outro lado, caso o valor da potência relativa a mensagens SYNC consecutivas se mantenha constante, o campo referente à velocidade dos nós móveis é omitido.

Tendo em conta as informações fornecidas pelo nó emissor do SYNC acerca da velocidade dos nós móveis, os seus vizinhos procedem à formação de uma zona activa em torno desse nó, aquando da sua deslocação entre dois *clusters*, ou seja, entre dois grupos distintos de nós. Esta zona activa caracteriza-se pelo facto de os nós integrantes repetirem o período de sincronização um maior número de vezes, permitindo a diminuição do tempo necessário para estabelecimento de uma nova sincronização no *cluster* de destino.

Assim, devido à criação desta zona activa aquando da deslocação entre *clusters* distintos, evita-se a necessidade de um nó esperar 2 minutos até estabelecer uma nova ligação. O estabelecimento de ligações dentro de um *cluster* demora apenas o intervalo de tempo entre as mensagens SYNC, ou seja, 10s. Daí, conclui-se que a utilidade da zona activa verifica-se apenas durante a deslocação do nó móvel entre dois *clusters* distintos.

Caso se verifique a ausência de mobilidade dos nós ou a sua deslocação no interior de um único *cluster*, este protocolo funciona de forma similar ao S-MAC.

O protocolo MS-MAC, colmatando uma das principais lacunas do protocolo S-MAC inerente ao suporte de mobilidade, apresenta ainda diversos problemas no que diz respeito à sua utilização em redes móveis. De entre esses problemas destaca-se o elevado consumo energético verificado no caso de um nó não possuir nós na sua vizinhança, já que o seu funcionamento transita para um modelo idêntico ao S-MAC mas mantém-se acordado 10s no início de cada período de 2 minutos. Assim, conclui-se que na presença de mobilidade este protocolo releva-se excessivamente rígido.

#### **2.3.3.5. LL-MAC**

O protocolo LL-MAC [Marín 2008] baseia-se num modelo TDMA, consistindo num protocolo de recolha de dados (*Collecting Data*) a partir dos diversos nós da rede até à raiz. Neste protocolo as redes encontram-se organizadas de uma forma hierárquica, em torno de uma raiz centrada numa estação base (*BS – Base Station*).

O esquema *Time Division Multiple Access* (TDMA), consiste num método de acesso ao meio para ambientes partilhados, que permite a diversos utilizadores partilharem o

mesmo canal de frequência através da divisão desse canal em intervalos de tempo, designados *slots*. Os *slots* consistem em pequenos períodos temporais sincronizados entre si. (Figura 2.6).

A transmissão dos dados até à raiz efectua-se através de diversos percursos *multi-hop* (vários saltos até à raiz). Além disso, devido à estruturação hierárquica, a contagem de número de *hops* (saltos) é efectuada tendo sempre como referência a raiz.

Neste protocolo, o período de trabalho  $T_c$  é dividido em dois períodos distintos: período activo (*AP – Active Period*) e o período em que o nó permanece adormecido (*SP – Sleep Period*). Além disso, o período activo é dividido em outros dois intervalos de tempo: período de controlo, onde é trocada informação acerca da topologia e um intervalo de tempo dedicado à transmissão de dados, durante o qual os dados são encaminhados até ao destino.

O intervalo de controlo apresenta significativas diferenças comparativamente com semelhante espaço de tempo em outros protocolos existentes em redes de sensores sem fios, já que não ocorre uma disputa pelo meio, embora os nós o escutem continuamente. O intervalo de controlo é dividido em três sub-intervalos:

- *Node advertisements* (NA): os nós anunciam informações indispensáveis para a obtenção de sincronismo entre si, conhecerem a vizinhança relativa à sua localização e escolherem um nó pai (*parent*) (nó com o qual se ligam na rede);
- *Adoption Request* (CAR): dedicado a novos nós que se pretendam integrar na rede existente;
- *Child Confirm* (CC): utilizado pelos nós mais antigos na rede para confirmar a integração dos novos elementos.

Relativamente ao segundo e terceiro intervalos, estes apenas são usados quando um nó necessita de alterar a relação com o seu pai na rede (por este desaparecer por exemplo) ou caso surja um novo nó na vizinhança.

O intervalo de tempo referente à transmissão de dados foi definido de modo a minimizar a latência com um baixo custo de computação. Desta forma, encontra-se dividido em  $M$  secções, as quais se encontram particionadas em  $N$  *sub-slots*, como ilustrado na Figura 2.6. Cada nó comunica com o seu *pai* no *sub-slot* alocado dentro da divisão correspondente ao *hop* onde está localizado.

No processo de transmissão da informação *hop-by-hop* desde o nó mais longínquo até à raiz, os primeiros nós comunicam através do número de *hop* mais elevado (divisão M-1), enquanto os nós seguintes utilizam a divisão M-2 e assim sucessivamente. Os nós pais escutam o canal em busca de informação dos seus nós filhos, em *slots* diferentes, armazenando a informação recebida. Na próxima divisão, cada nó pai envia os dados provenientes dos nós filhos na divisão anterior, para os nós com ao quais estabeleceram ligação aquando da sua entrada na rede (seus pais) e assim sucessivamente.

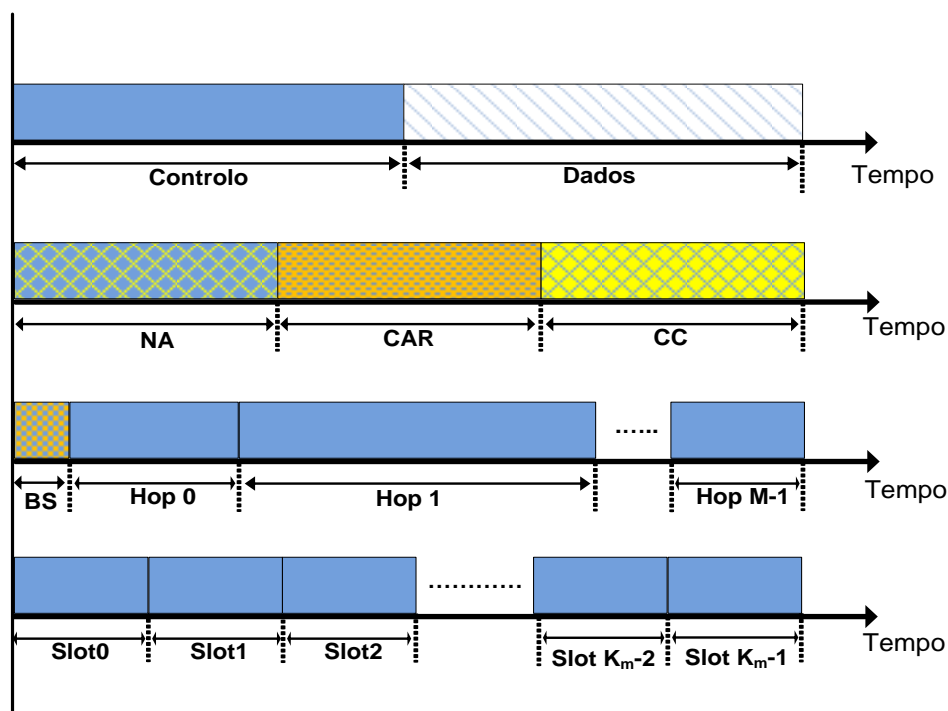


Figura 2. 6 - Intervalos de Controlo e de Transmissão de dados no protocolo LL-MAC

Neste protocolo, cada nó difunde os seus anúncios (*advertisements*), recebe informação dos nós filhos (*childs*) (nós recentes na rede) e retransmite esses mesmos dados para os pais. No que diz respeito aos nós órfãos (*orphan*), isto é, sem ligação com outro nó na rede, permanecem acordados continuamente, escutando o canal esperando por um anúncio.

Uma estação base (BS) consiste no nó encarregue de estabelecer o ciclo de trabalho  $T_c$ , assim como a distribuição temporal dos intervalos de tempo em que cada nó permanece activo ou adormecido.

Em cada ciclo de trabalho  $T_c$ , cada nó procede ao envio dos seus dados para o nó pai no *slot* dedicado. O protocolo LL-MAC estabelece como regra para os nós pais detectarem a mobilidade dos nós filhos e verificarem a saída destes da sua vizinhança, pelo facto de não

receberem no *slot* dedicado nenhuma mensagem do nó filho durante 3 ciclos consecutivos, eliminando posteriormente a totalidade das informações referentes a esse nó.

Tal como mencionado anteriormente, os anúncios possuem as informações necessárias para a manutenção de uma coerente sincronização entre os nós, permitindo a correcção de erros, por exemplo, devido aos *drifts* de relógio. Deste modo, os nós filhos recebem os anúncios do seu nó pai, retirando a respectiva informação relativa à sincronização. Posteriormente, as variáveis do receptor são actualizadas de acordo com os valores recebidos passando essas novas magnitudes a figurar nos anúncios por si enviados.

No que respeita ao desempenho do protocolo LL-MAC na presença de mobilidade, este apresenta como principal problema o facto de exigir uma minuciosa precisão no sincronismo exigido entre os nós, no mínimo ao nível dos milissegundos. Na presença de mobilidade dos nós este requisito torna-se extremamente difícil de ser cumprido.

#### 2.3.3.6. LEACH

O protocolo *Low-Energy Adaptive Clustering Hierarchy* (LEACH) [Heinzelman 2000] apresenta uma metodologia para organização dos nós em *clusters* definindo um desses membros como o *cluster head* (líder) do grupo. Com o intuito de não penalizar o nó líder do *cluster*, em termos do consumo de energia, utiliza-se um algoritmo que permite a mudança aleatória desse mesmo líder. Por outro lado, ao nó responsável pela liderança do *cluster*, cabe-lhe também a função de agregar os dados, de forma a reduzir o número de mensagens transmitidas e consequentemente diminuir os gastos energéticos, aumentando o tempo de vida da rede.

Finalmente, no protocolo LEACH, o número óptimo de *clusters* presentes numa rede pode ser determinado, *à priori*, pelo próprio sistema. A obtenção desse valor óptimo depende de um conjunto de parâmetros, tais como a topologia da rede e os custos de computação versus requisitos mínimos de comunicação.

Relativamente ao funcionamento do protocolo na presença de mobilidade dos nós, este revela-se excessivamente rígido já que a cada alteração em determinado *cluster*, corre o algoritmo para determinar o líder desse mesmo cluster. Tendo em conta que na presença de mobilidade as mudanças na rede ocorrem com uma considerável frequência, o número de vezes que é utilizado o protocolo LEACH pode ser bastante elevado.

## 2.3.4. Abordagem Híbrida

### 2.3.4.1. Z-MAC

O protocolo Z-MAC (*Zebra-MAC*) [Rhee 2005] propõe um modelo híbrido para redes de sensores sem fios, procurando conjugar as vantagens dos esquemas TDMA e CSMA. Em situações de carga elevada utiliza-se um esquema TDMA, enquanto em casos de baixa carga usa-se um comportamento semelhante ao mecanismo CSMA.

A descoberta de vizinhos efectua-se difundindo periodicamente uma mensagem PING, que inclui a lista de vizinhos a *1-hop* do nó emissor. Deste modo, ao receber esta mensagem e a consequente lista de vizinhos, um nó passa a conhecer os nós presentes a uma distância de *2 hops*, com um atraso máximo de dois períodos de PING.

No que diz respeito à alocação de *slots*, é utilizado o algoritmo DRAND [Rhee 2006] [Rhee 2004]. O DRAND consiste numa implementação distribuída do algoritmo RAND [Ramanathan 1997], assegurando uma distribuição aleatória dos *slots* atribuídos que garante a inexistência de nós a uma distância inferior ou igual a dois *hops* com o mesmo *slot* reservado para a transmissão de dados. Este método de alocação de *slots* possibilita a ausência de interferência entre os nós da vizinhança, situados a 1 ou 2 *hops*, embora possa existir interferência a mais *hops*.

Na presença de mobilidade dos nós, o protocolo Z-MAC é inadequado pois requer sucessivas invocações do algoritmo DRAND para alocação de *slots*. Durante este processo de alocação de *slots* os nós estão impossibilitados de transmitir mensagens. Além disso, o facto de a alocação ser aleatória leva a que os *slots* alocados possam não ser, de entre os disponíveis, os que minimizam o tempo de propagação de dados.

Além disso, em casos de existência de uma carga elevada utiliza-se um esquema TDMA, ao qual está associado a necessidade de existência de uma extrema precisão de sincronismo (ao nível dos milissegundos). Na presença de mobilidade a existência desta precisão torna-se consideravelmente difícil, consistindo noutra desvantagem deste protocolo.

### 2.3.4.2. MH-MAC

O MH-MAC (Multimode Hybrid MAC) [Bernardo 2007] consiste num protocolo híbrido para redes de sensores sem fios, no qual os nós se encontram por omissão no modo FULL-ON, sem dormir, podendo transitar para o modo assíncrono ou síncrono, através da configuração efectuada pela aplicação.

No que diz respeito ao modo de funcionamento assíncrono, é utilizado um algoritmo LPL semelhante ao utilizado no protocolo X-MAC [Buettner 2006]. Os dados são precedidos de uma sequência de pequenos preâmbulos com espaçamento entre si, durante um intervalo de tempo igual a 2 vezes o período do *duty-cycle*. Ou seja, duas vezes o intervalo de tempo decorrido entre duas transições consecutivas do estado activo para o adormecido

No caso da transmissão de dados em *broadcast*, de modo a diminuir os custos energéticos devidos à ocorrência do fenómeno de *overhearing*, o protocolo MH-MAC introduziu um campo adicional nos preâmbulos que indica a duração do intervalo até ao envio do pacote de dados. Deste modo, os nós podem transitar para um estado de adormecidos, acordando no momento exacto em que se prevê o envio do pacote.

A Figura 2.7 ilustra a transmissão de dados em *unicast*, a qual se inicia através do envio de um conjunto de pequenos preâmbulos por parte do futuro emissor dos dados. O receptor, ao ligar o rádio, confirma a recepção de um preâmbulo através de um PACK, seguindo-se o envio dos dados. Estes podem ser confirmados através do recurso a uma mensagem ACK.

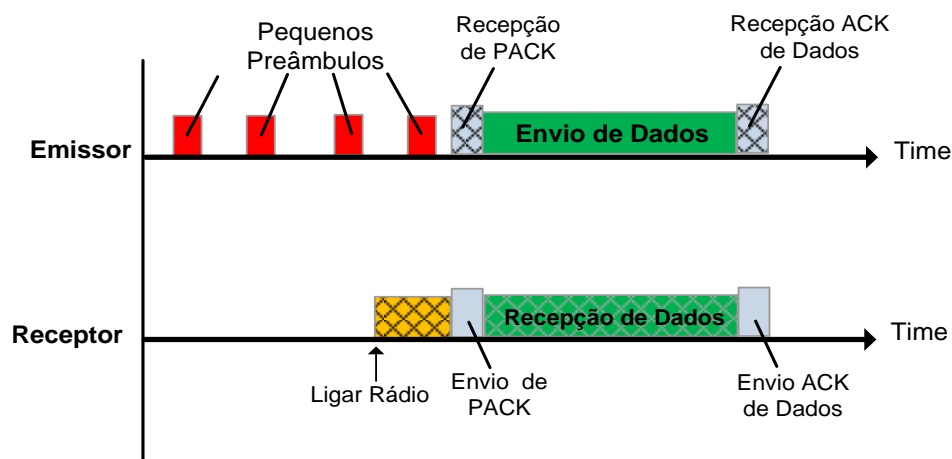


Figura 2.7 - Transmissão de dados no protocolo MH-MAC

O protocolo MH-MAC utiliza um pacote designado *Shut Up* para reduzir as colisões de preâmbulos. Caso um nó receptor escute preâmbulos provenientes de mais do que uma origem, envia um pacote *Shut Up* com probabilidade  $p$  estática, com o objectivo de terminar o envio de preâmbulos por parte de todos os emissores, excepto um.

No caso do modo de funcionamento síncrono, cada *macro-slot* é constituído por uma sequência de onze *slots*, cada um com 100ms de duração, existindo um ou mais *slots* públicos e zero ou mais *slots* dedicados. Por omissão é usado um *slot* público e dez *slots* dedicados. Os *slots* dedicados são destinados à comunicação *unicast* entre dois nós. Nos *slots* públicos todos os nós podem comunicar, sendo os *slots* onde se realizam os processos de manutenção do sincronismo entre nós. Em ambos os tipos de *slots* os nós recorrem a um protocolo de contenção semelhante ao verificado no T-MAC [van Dam 2003], de modo a reduzir os gastos energéticos.

De modo a manter a sincronização entre os diversos nós presentes na rede, são difundidos nos *slots* públicos pacotes SYNC, os quais indicam o início do *duty-cycle* do nó emissor, permitindo aos receptores ajustar o seu *duty-cycle* de acordo com os dados presentes na mensagem SYNC. Esta mensagem SYNC apresenta uma validade de 180 segundos. Note-se a importância da mensagem SYNC de modo a colmatar os erros na sincronização entre os nós, impostos, por exemplo, pelos *drifts* de relógio.

O protocolo MH-MAC permite a manutenção de nós em modo *full-on* ou assíncrono na vizinhança de nós síncronos, necessitando, os nós não síncronos, de manter uma tabela como a distribuição dos *slots* públicos dos nós síncronos e de iniciar a transmissão dos pacotes num *slot* público. Caso o destino das mensagens seja desconhecido, os nós síncronos enviam um preâmbulo antecedendo os dados, de modo a acordar o nó receptor, sendo a sua recepção confirmada através de um PACK (Figura 2.7). Assim, caso os instantes em que os nós acordam estejam alinhados, o *overhead* imposto pelo envio do preâmbulo é minimizado, já que a recepção do mesmo ocorre após um curto intervalo de tempo.

Na presença de mobilidade dos nós, torna-se indispensável uma extrema rapidez aquando da associação de um nó assíncrono a um nó síncrono. Isto é, o intervalo de tempo decorrido entre um nó assíncrono iniciar o processo de transição para síncrono e o início da transmissão dos dados deve ser reduzido. Neste protocolo a obtenção do *slot* dedicado e o início da transmissão demora pelo menos três vezes o período do *macro-slot* relativo ao

modo síncrono (1.1s). Este valor mínimo de 3.3s revela-se excessivo para um cenário de mobilidade em redes de sensores sem fios.

No protocolo MH-MAC o método para manter o sincronismo entre os nós e assim colmatar aspectos como os *drifts* de relógio existentes nos diversos nós (sensores), realiza-se com uma periodicidade reduzida. Este facto pode conduzir a consideráveis diferenças no sincronismo entre os nós, o que poderá ser dramático, sobretudo num cenário de mobilidade.

O facto de a probabilidade de envio da mensagem *Shut Up* ser fixa constitui outra desvantagem deste protocolo, já que este valor deveria variar consoante o número de possíveis emissores simultâneos do pacote *Shut Up*.

### 2.3.4.3. MobH-MAC

O protocolo MobH-MAC [Raja 2008] (*Mobility Adaptive Hybrid Protocol*) consiste num protocolo híbrido para redes de sensores sem fios móveis, baseado numa abordagem síncrona, no qual ocorre diferenciação entre *slots* destinados a nós móveis ou nós estáticos. Para nós estáticos, utiliza-se um mecanismo TDMA baseado em instantes programados para acordar/adormecer (*schedule-based*), enquanto nos *slots* destinados aos nós móveis é usado um modelo de contenção.

Neste protocolo verifica-se o recurso a um algoritmo destinado a estimar o nível de mobilidade baseado na potência do sinal recebido. Consoante o nível de mobilidade da rede, procede-se a um ajuste do número de *slots* para nós estáticos e móveis, em função do tempo da trama.

Cada nó envia periodicamente uma mensagem HELLO com o intuito de assinalar a sua presença na vizinhança e um *beacon* transmitido no primeiro *slot* de cada trama, com informação acerca do nível de mobilidade. Este *beacon* é enviado com a mesma potência da mensagem anterior. Comparando as sucessivas potências das mensagens recebidas, cada nó determina a sua posição relativamente aos vizinhos. Deste modo, um nó encontra-se habilitado a determinar o seu índice da mobilidade, sendo este enviado na próxima mensagem *broadcast* de controlo.

Com base no nível de mobilidade de cada nó, define-se um limiar que permite decidir se um nó é estático ou móvel, sendo esse aspecto decisivo para o processo de alocação de *slots* onde são diferenciados os dois tipo de nós.

Numa rede em que os nós estão em constante mobilidade, estes adaptam continuamente o tempo de trama, assim como a razão entre *slots* dedicados para nós móveis/estáticos. Consequentemente, este facto poderá originar discrepâncias entre os valores presentes nos nós quando são alterados os parâmetros acima mencionados. Com o intuito de evitar essas disparidades os novos valores são reunidos num *cluster head*, o qual calcula a média das diversas magnitudes e difunde esses valores para os diversos nós vizinhos, no final da trama. O líder do cluster (*cluster head*) é alterado segundo a abordagem proposta no protocolo LEACH.

Um nó móvel, após entrar na vizinhança, junta-se à rede existente num intervalo de tempo que depende do intervalo de espera até à recepção do *beacon* enviado pelos nós fixos. Aquando do envio dos dados, os nós concorrem pelo meio em duas fases, enviando na primeira um tom de modo a acordar os nós vizinhos e na segunda os dados.

Nos *slots* destinados aos nós estáticos, é usado um mecanismo idêntico ao utilizado no protocolo LMAC [van Hoesel 2004] para reserva de *slots*. A totalidade dos *slots* destinados aos nós estáticos divide-se em duas partes distintas:

- Controlo: uma secção do *slot* escutada por todos os nós estáticos, que se destina à troca de informação acerca do estado actual dos *slots* na vizinhança.
- Dados: mantêm-se apenas o remetente e o receptor acordados, existindo a possibilidade de cada nó ter diversos *slots* atribuídos, dependendo do tráfego, ao contrário do que se verifica no protocolo LMAC.

Por outro lado, verifica-se a existência de um mecanismo de *timeout*, com o objectivo de evitar a permanência do nó acordado em determinado *slot*, não existindo dados para transmissão.

No caso dos *slots* dedicados aos nós móveis, recorre-se ao protocolo SCP-MAC de modo a evitar o *overhead* de um esquema de TDMA e permitir o envio de dados com um reduzido custo energético, possível devido ao elevado nível de sincronização entre nós requerido pelo protocolo.

No que diz respeito à inicialização da rede, cada nó sincroniza-se com os seus vizinhos e determina o seu nível de mobilidade. De modo idêntico ao protocolo LMAC, o *gateway* difunde uma mensagem de controlo em *broadcast*, permitindo aos restantes nós sincronizarem-se de acordo com a referência temporal presente nessa mensagem. Cada nó envia um *beacon* com informação acerca do índice de mobilidade com o objectivo de

determinar o seu nível de mobilidade. Após estar sincronizado e com o nível de mobilidade determinado, um nó pode seleccionar *slots* destinados a nós móveis ou a estáticos para transmitir dados. A execução desta escolha é realizada com base num limiar referente ao nível de mobilidade.

#### 2.3.4.4. *Funneling-MAC*

Num cenário de recolha de dados as redes de sensores sem fios apresentam um efeito de funil, devido ao tráfego originado por diversos emissores para um único destino, seguindo uma *sink tree*. Este aspecto resulta, numa zona próxima do nó destino, num aumento da intensidade de tráfego, do número de colisões, da congestão, da perda de pacotes e de gastos energéticos.

O protocolo *funneling-MAC* [Ahn 2006] foi elaborado com intuito de diminuir este efeito de funil através de um modelo híbrido, onde os nós localizados na zona junto ao destino funcionam segundo um algoritmo de TDMA, enquanto nos restantes se verifica a existência do método de CSMA/CA.

O nó destino procede ao envio periódico de um *beacon*, definindo o limite da zona de maior intensidade de tráfego (*funneling*) através do valor da potência de emissão. Os restantes nós ao receberem o mencionado *beacon* tomam conhecimento da sua presença na zona de elevada intensidade de tráfego, passando a funcionar segundo um modelo TDMA. Os restantes nós usam um algoritmo de CSMA, sendo este o modo de funcionamento por omissão dos nós da rede.

No que diz respeito ao método de sincronização existente entre os nós presentes na zona de elevada intensidade de tráfego (designados *f-nodes*), tal processo é efectuado apenas com base na referência temporal existente no *beacon* enviado pelo nó destino. Isto é, cada nó ao receber a mencionada mensagem, procede ao ajuste do relógio com base no valor do relógio do nó destino. Este aspecto consiste numa das principais desvantagens deste protocolo, já que o tempo de processamento da mensagem pelas camadas abaixo do nível MAC e o tempo de propagação podem gerar erros na ordem dos milissegundos, os quais não são compensados pelo protocolo.

Outro aspecto a realçar prende-se com facto de o protocolo *funneling-MAC* procurar minimizar o número de *beacons* enviados pelo nó destino, através da ausência desse envio

caso a rede se encontre livre ou com um nível reduzido de tráfego. Além disso, a totalidade dos *f-nodes* inicia um relógio após a recepção de um *beacon*, transitando para um modo de funcionamento baseado no algoritmo de CSMA caso este expire. Ou seja, se um *f-node* não receber nenhum *beacon* durante um elevado intervalo de tempo significa que deixou de estar na zona de funil, e transita para o modo de funcionamento CSMA.

O protocolo funnelling-MAC apresenta diversos problemas na presença de mobilidade dos nós da rede, de entre os quais se destaca o facto de a transição do modo de funcionamento CSMA para o modo TDMA depender do período do *beacon*. Como apenas os nós que recebem o *beacon* funcionam em modo TDMA, um nó que se encontre a funcionar utilizando CSMA e que acabou de entrar na vizinhança, apenas efectua a transição quando receber o próximo *beacon*. Sendo o período de envio do *beacon* elevado (20s, o valor mínimo usado nos testes reportados), a espera é considerável. Este aspecto prejudica o desempenho do protocolo, sobretudo no que respeita ao *throughput*.

## 2.4. Protocolos de Encaminhamento

Nos protocolos realizados no domínio das MANETs (redes ad hoc móveis) são criadas soluções para lidar com a mobilidade dos nós. No entanto, nesses protocolos não são desenvolvidos métodos para diminuir o consumo energético dos nós, devido à maior facilidade em recarregar as baterias dos nós numa MANET.

Numa rede de sensores sem fios móvel torna-se indispensável o desenvolvimento de métodos para lidar com a mobilidade dos nós, cujos conceitos base podem ser herdados dos protocolos desenvolvidos para as MANET. Tendo em conta o reduzido tempo de duração das baterias dos nós numa rede de sensores sem fios e geralmente a dificuldade da sua substituição, a principal diferença entre este domínio e o das MANETs consiste na necessidade de minimizar os gastos energéticos, mantendo o suporte para mobilidade dos nós.

De seguida são descritos alguns protocolos desenvolvidos para MANETs que fornecem suporte à mobilidade dos nós.

### 2.4.1. ABR

O protocolo *Associativity-Based Routing* (ABR) [Toh 1996] baseia-se no conceito de estabilidade da ligação. Cada nó gera periodicamente um *beacon* (mensagem HELLO) de modo a assinalar a sua presença aos nós da vizinhança. Estes *beacons* são usados para actualizar a tabela de vizinhos mantida por cada nó, juntamente com o número de *beacons* recebidos de cada um desses vizinhos.

Este protocolo usa como métrica para uma determinada ligação, o grau de estabilidade da mesma. Este nível de estabilidade é calculado com base no número de sucessivos *beacons* trocados em determinado intervalo. Um número superior de *beacons* trocados significa ligações mais estáveis, as quais são as consideradas durante a selecção das rotas. Assim, não há tantas trocas periódicas das rotas existentes na rede, já que essa escolha é efectuada com base no nível de estabilidade.

### 2.4.2. OLSR

O protocolo *Optimized Link State Routing* (OLSR) [Clausen 2001] utiliza o conceito de *Multipoint Relays* (MPR), de modo a proceder à disseminação das alterações das diversas ligações existentes na rede, sendo estas geradas apenas pelo nós seleccionados como MPRs.

A definição de MRP tem como principal objectivo minimizar o custo total de disseminação das mensagens de controlo enviadas em *broadcast*, reduzindo as retransmissões redundantes. Cada nó na rede selecciona um conjunto de nós formado pelos seus vizinhos directos (1 *hop*), os quais são designados MPR e ficam responsáveis pela transmissão, em *broadcast*, de mensagens de actualizações recebidas pelo nó que os seleccionou.

Nos protocolos de estado de linha (por exemplo, OSPF), todos os nós inundam a rede com informação sobre o estado das ligações. O protocolo OLSR usa o mesmo princípio, mas com duas optimizações. Em primeira instância, apenas os nós associados aos MPR são declarados nas mensagens referentes ao estado das ligações, o que provoca uma diminuição do tamanho das mensagens. Além disso, apenas os nós MPR geram mensagens sobre o estado das ligações, levando a uma diminuição do número de nós emissores de mensagens.

No protocolo OLSR, cada nó selecciona um conjunto dos seus nós vizinhos a 1 *hop* de modo que estes consigam alcançar todos os nós vizinhos a 2 *hops* do nó que os seleccionou,

ou seja, cada nó do conjunto de MPR deve ter uma ligação directa com um dos vizinhos directos.

Quanto menor o conjunto de MPR, menor será o custo total de envio das mensagens geradas pela inundação. Cada nó mantém uma tabela com informações dos MPR, a qual é actualizada através de mensagens *Hello* trocadas com esses nós MPR.

A mensagem *Hello* é enviada periodicamente, com o objectivo de sinalizar a selecção de vizinhos MPR, detectar a vizinhança e verificar o alcance das ligações. Com base nas mensagens *Hello* entre vizinhos (1 hop), um nó pode obter informações acerca dos nós situados a 2 hops.

Outro tipo de mensagem de controlo utilizada neste protocolo consiste na TC (*Topology Control*), a qual permite proceder à actualização das informações sobre os vizinhos. Estas mensagens enviadas periodicamente contêm a lista de vizinhos que seleccionaram o emissor como seu MPR.

A utilização de MPR diminui o número de mensagens de encaminhamento TC enviadas. No entanto, o reduzido número destas mensagens pode tornar-se uma vulnerabilidade quando a qualidade das ligações é má. Como há menos mensagens redundantes, as suas eventuais perdas conduzem à existência de tabelas de encaminhamento desactualizadas.

Como estas tabelas podem estar dessincronizados nos diferentes nós, podem ocorrer ciclos no encaminhamento, levando a congestionamento e mais perdas de mensagens, repetindo-se este ciclo indefinidamente.

Assim, o protocolo OLSR, graças ao recurso aos MPRs reduz o custo de encaminhamento existente na presença de uma rede consideravelmente densa, comparativamente com a maioria dos protocolos de estado de linha. No caso da existência de uma rede dispersa, o desempenho deste protocolo aproxima-se do desempenho relativo a um protocolo de estado de linha tradicional.

Um dos problemas associados ao protocolo OLSR diz respeito à dificuldade em escolher um intervalo para o envio periódico do estado das ligações na rede que permita reflectir no mínimo intervalo de tempo possível às alterações existentes na rede.

### 2.4.3. Conclusões

Os dois protocolos de encaminhamento para MANET apresentados permitem verificar que os mecanismos usados para seguir nós móveis são comuns com os protocolos apresentados anteriormente para redes de sensores sem fios – o envio periódico de *beacons*, e o agrupamento de nós. São depois acrescentados mecanismos, como a medição da estabilidade, ou o funcionamento *on demand* (no AODV [Chakeres 2004]), para reduzir a utilização de recursos de rede e melhorar o desempenho na presença de mobilidade.

## 2.5. Aplicações das Redes de Sensores sem Fios

Actualmente, a investigação em redes de sensores sem fios estende-se por diversas áreas, sendo este um aspecto que revela a imensa aplicabilidade potencial desta tecnologia, ainda em constante evolução. No sector empresarial as redes de sensores sem fios começam, presentemente, a dar os primeiros passos.

De entre as áreas de aplicação das redes de sensores sem fios destacam-se:

- Medicina - monitorização de pacientes, controlo na admissão de medicamentos, localização dos médicos nos hospitais;
- Domótica - aquecimento, ventilação, ar condicionado, detectores de intrusão, detectores de fumo, iluminação, sistemas de portas automáticas;
- Ambiental - detecção de incêndios, detecção de cheias, monitorização de movimentos dos animais, mapeamento da biodiversidade;
- Electrónica - controlo remoto universal, brinquedos, turismo;
- Educação - localização de crianças em escolas e espaços públicos, bombeiros, animais;
- Militar - localização de soldados, reconhecimento de inimigos, sistemas de mira, controlo de equipamentos e munição;
- Outras: controlo de tráfego, aviação, entretenimento

## Capítulo 3.

# Arquitectura Do Protocolo

---

Este capítulo apresenta a arquitectura do protocolo de controlo de acesso ao meio para redes de sensores sem fios móveis, proposto na presente dissertação.

A primeira secção apresenta uma visão geral dos principais objectivos do sistema, o cenário considerado e uma sucinta descrição do trabalho desenvolvido. Na segunda secção efectua-se uma descrição da estrutura do protocolo desenvolvido. A terceira secção apresenta o funcionamento do protocolo, focando aspectos como a transição entre os diversos modos de funcionamento e a manutenção de sincronismo entre os nós. Na última secção descreve-se a *interface cross layer* destinada ao controlo do protocolo MMH-MAC.

### 3.1. Introdução

Na presente dissertação é proposto o protocolo MMH-MAC (*Mobile Multimode Hybrid MAC*), o qual visa através de um modelo híbrido de funcionamento, efectuar a gestão de redes de sensores sem fios móveis.

O cenário idealizado caracteriza-se pela existência de uma rede constituída simultaneamente por nós estáticos e por nós móveis. Este é o caso verificado numa área onde determinados sensores efectuem a monitorização de certas grandezas e outros deslocam-se com o intuito de recolher, no menor intervalo de tempo, os dados derivados da monitorização.

Deste modo, uma das possíveis aplicações deste protocolo seria um sistema de monitorização de incêndios constituído por nós estáticos, onde se verifica também a existência de nós móveis, os quais recolhem e armazenam informação acerca do estado actual do ambiente envolvente. Os mencionados nós móveis poderão ser sensores instalados no vestuário de bombeiros, em veículos terrestres (jipes, motas, etc.) ou em veículos aéreos (helicópteros).

Assim, o objectivo principal consiste no desenvolvimento de um protocolo de nível MAC apto a lidar com redes onde coexistem nós estáticos e móveis, desempenhando funções como monitorização e recolha de dados, e permitindo a comunicação eficiente entre nós independentemente da classe (móveis ou estáticos) a que pertencem.

Nas redes de sensores sem fios estáticas, além da existência de um tráfego periódico, poderão verificar-se rajadas de tráfego às quais o presente protocolo responde mantendo o desempenho, já que pode transitar entre um modo de funcionamento assíncrono e um modo de funcionamento síncrono. Assim, no caso de se verificar uma elevada carga, os nós funcionam num modo síncrono permitindo a maximização do *throughput* e salvaguardando a existência de reduzidos consumos energéticos.

Adicionalmente, a coexistência de nós móveis aumenta consideravelmente o desafio inerente à implementação de um protocolo de nível MAC. Os nós móveis podem agregar-se a *clusters* (grupos) estáticos existentes ou criar ligações entre si formando *clusters* móveis. Mas tal só pode ser realizado eficientemente pressupondo a existência de uma rápida agregação desses nós aos *clusters* estáticos, de forma a manter o *throughput* elevado em situações de tráfego periódico ou em rajadas e a conservação de uma elevada eficiência energética. Outro aspecto a ter em conta diz respeito à necessidade de se verificar a existência de um método que permita a colmatação dos *drifts* de relógio existentes nos nós, suportando um rápido alinhamento do relógio dos nós móveis face aos membros de um *cluster*.

Deste modo, além de manter as principais virtudes relativas a redes estáticas, inerentes ao modelo híbrido proposto anteriormente no protocolo MH-MAC [Bernardo 2007], o protocolo MMH-MAC procura integrar e otimizar a presença de mobilidade numa rede de sensores sem fios, focando-se sobretudo nos aspectos mencionados no parágrafo anterior.

Tal como foi apresentado no Capítulo 2, diversos protocolos como o MS-MAC, MA-MAC ou MobH-MAC visaram tratar as principais questões relacionadas com a mobilidade dos nós em redes de sensores sem fios. No entanto, nenhum dos casos analisados apresenta um elevado débito, na presença de mobilidade dos nós.

Nessa perspectiva o protocolo MMH-MAC proposto nesta dissertação foi desenvolvido com o intuito de colmatar aspectos relacionados com a mobilidade em redes de sensores sem fios, cujas implementações anteriores não apresentam desempenhos otimizados. O mencionado protocolo poderá funcionar em modo assíncrono ou síncrono e ser configurado através de uma *interface cross-layer*. Por outro lado, manteve-se as virtudes do protocolo MH-MAC relativas à gestão de tráfego não periódico em redes estáticas.

O trabalho desenvolvido durante esta dissertação foi apresentado no 8º Seminário da Rede Temática de Comunicações Móveis (RTCM) [Água 2008], o qual decorreu na Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa e no mês de Junho do ano de 2008. Está em preparação um artigo científico que irá ser submetido a uma conferência internacional ou a uma revista com os resultados desta dissertação acrescidos de dados obtidos com o protótipo a correr em Telos B, ainda em fase de teste.

### **3.2. Estrutura do protocolo MMH-MAC**

Dada a variedade de situações patentes numa rede de sensores sem fios na presença de mobilidade, torna-se indispensável a criação de um protocolo cujo modo de funcionamento seja ajustado através da escolha efectuada pelas aplicações, através de uma *interface cross-layer*.

Nos casos onde se verifica a ausência de dados para transmissão, o aspecto fulcral a considerar numa rede de sensores sem fios consiste na poupança energética, levando a um funcionamento em modo assíncrono. Pelo contrário, em situações onde se verifique a existência de uma elevada quantidade de tráfego para transmissão e a necessidade de minimizar o atraso, torna-se adequado o funcionamento do protocolo no modo síncrono. Assim, na medida em que se torna essencial a gestão do modo de funcionamento do protocolo consoante a situação da rede, o protocolo MMH-MAC apresenta a possibilidade deste modo ser definido pela aplicação.

Além disso, a necessidade de lidar com a mobilidade nas redes de sensores sem fios tornou indispensável o desenvolvimento de métodos para descoberta e detecção de perda de vizinhança. A variação da vizinhança pode automaticamente levar à transição autónoma entre os dois modos de funcionamento, devendo esta transição ocorrer no menor intervalo de tempo possível. Nas duas secções seguintes procede-se à análise dos dois modos de funcionamento do protocolo MMH-MAC.

### 3.2.1. Modo Assíncrono

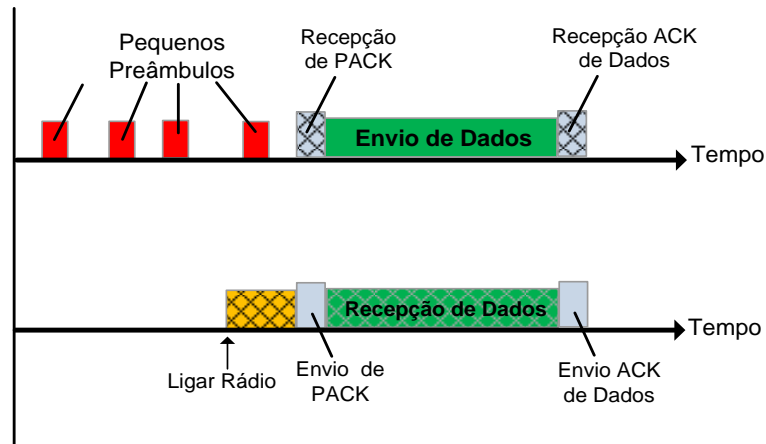
Durante os intervalos de tempo nos quais se verifica apenas a existência de tráfego esporádico para transmissão, o principal objectivo de um protocolo de nível MAC concebido para redes de sensores sem fios prende-se com a redução do consumo energético. Neste caso os nós devem funcionar em modo assíncrono, apresentando um valor de *duty-cycle* reduzido. Tal como analisado no capítulo anterior, no que diz respeito aos conceitos gerais, os protocolos adequados para casos onde se verifica a transmissão esporádica de dados com mobilidade são o X-MAC [Buettner 2006] e o B-MAC [Polastre 2004].

Assim, tendo em conta o facto de protocolo X-MAC utilizar um algoritmo de LPL (*Low Power Listening*), anulando as principais lacunas existentes no B-MAC e descritas no Capítulo 2, optou-se por tomar como base as características gerais desse protocolo. Além disso, manteve-se ainda as principais optimizações realizadas no protocolo MH-MAC para o funcionamento assíncrono dos diversos nós constituintes de uma rede.

Antecedendo o envio dos dados, um nó emissor procede ao envio de uma sequência de pequenos preâmbulos espaçados entre si por um intervalo de tempo aleatório, de modo a reduzir a probabilidade de colisões entre os preâmbulos. Este envio decorre durante um intervalo de tempo máximo de duas vezes o período do *duty-cycle*, anulando os problemas derivados da existência de um único preâmbulo com dimensões elevadas, os quais estão sobretudo relacionados com a ocorrência de *overhearing*. O envio dos preâmbulos ocorre de modo a permitir aos nós vizinhos acordarem e verificarem a existência de um nó apto para efectuar a transmissão de dados.

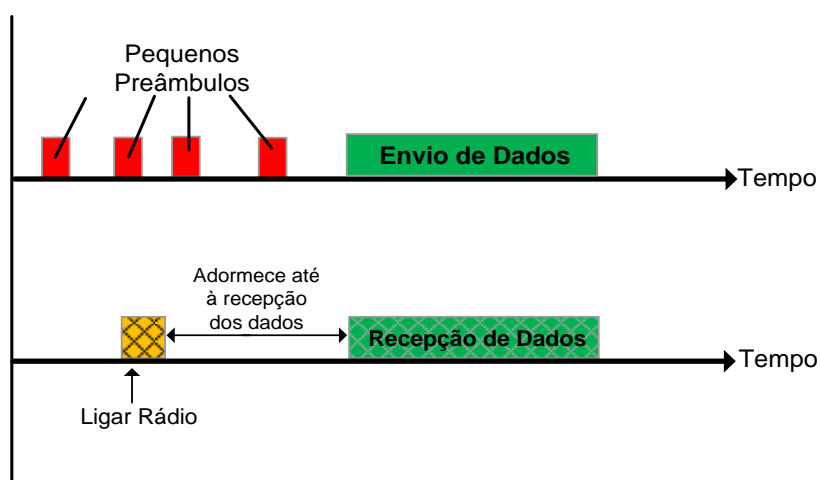
No que diz respeito à transmissão de dados em *unicast*, cada preâmbulo possui informação acerca do seu emissor e receptor. Baseando-se nesta informação, o receptor, após receber um preâmbulo envia uma confirmação de preâmbulo (PACK) ao emissor. Tal

como ilustra a Figura 3.1, após a recepção do PACK, o nó emissor procede de imediato ao envio dos dados. Além da informação relativa ao emissor e ao destinatário, os pequenos preâmbulos contêm ainda o tempo a decorrer até ao início do envio dos dados.



**Figura 3. 1 - Transmissão unicast no protocolo MMH-MAC**

No que diz respeito à transmissão de dados em *broadcast*, a principal diferença prende-se com o facto de não se encurtar o tempo de transmissão dos preâmbulos, procurando-se assim garantir que todos os nós vizinhos e possíveis destinos dos dados acordem com o envio dos mesmos. Deste modo, no caso de uma transmissão em *broadcast* o tempo de envio dos preâmbulos é quase sempre igual a dois períodos de funcionamento como se pode observar na Figura 3.2. A excepção, apresentada adiante, ocorre quando existem nós a funcionar em modo síncrono na vizinhança, ou quando são detectadas colisões entre preâmbulos.



**Figura 3. 2 - Transmissão broadcast no protocolo MMH-MAC**

Tal como acontece no caso da transmissão em *unicast*, os preâmbulos possuem informação acerca do intervalo de tempo até ao início do envio dos dados. No caso dos nós que não correspondem ao destino dos dados, estes determinam o tempo decorrente até ao envio dos dados e permanecem num estado de adormecidos com o objectivo de não interferirem com outra transmissão em curso.

Por outro lado, os nós que são destino da mensagem utilizam o tempo presente nos preâmbulos, relativo ao intervalo decorrente até ao envio dos dados para determinar com precisão esse instante e iniciar um período de adormecimento, acordando ligeiramente antes da transmissão dos dados. O intervalo de tempo decorrido entre o despertar do nó e a transmissão dos dados é dado por um valor aleatório entre 100 e 300ms.

Esta técnica, ao ser utilizada em casos onde se verifique apenas a interacção entre nós assíncronos, permite um considerável aumento das poupanças energéticas, já que os nós adormecem entre a recepção do preâmbulo e o início da transmissão dos dados. No entanto, a incerteza introduzida com o mecanismo de *Shut Up*, apresentado de seguida, reduz a eficiência do tempo médio de adormecimento.

Outro aspecto a ter em conta está relacionado com as colisões ocorridas durante o período relativo ao envio de preâmbulos. Devido ao conhecido problema do nó escondido [Bharghavan 1994], um nó pode escutar o canal, concluir que não existem transmissões e continuar uma transmissão que irá provocar colisões com pacotes com origem em outros nós.

Com o intuito de anular este problema foi introduzido um método probabilístico que permite parar a transmissão de preâmbulos, através do envio de uma mensagem designada por *Shut Up*. Caso um nó receptor receba num intervalo de 15ms, um número de preâmbulos superior a um e com origem em mais do que um emissor, poderá requerer o término da transmissão de um dos emissores através do envio de uma mensagem *Shut Up*.

Na Figura 3.3 apresenta-se um exemplo de utilização do pacote *Shut Up*, existindo dois emissores de preâmbulos e um único receptor. Após receber preâmbulos de ambos os emissores, o receptor sinaliza qual o emissor activo. O nó que permanecerá activo será o emissor do qual o nó recebeu um maior número de preâmbulos, sendo escolhido o nó de menor ID em caso de empate. O pacote *Shut Up* inclui o endereço do nó escolhido para continuar, desencadeando a paragem dos restantes nós. Assim caso existam  $N$  nós a colidir,

apenas é necessária uma trama, ao contrário da alternativa de se indicar quais os nós a terminar (na qual seriam necessárias  $N-1$  tramas).

O emissor, ao receber uma mensagem *Shut Up*, espera durante 15ms por mais mensagens deste tipo. Se não ocorrerem mais recepções e a mensagem recebida indicar que o nó deve permanecer activo, então continua a enviar preâmbulos. Caso contrário, termina o envio dos mesmos. Caso receba várias mensagens *Shut Up*, identifica qual o nó a permanecer activo indicado por estas, que apresenta menor ID. Apenas se esse identificador corresponder ao seu, o nó continua a enviar preâmbulos; caso contrário, termina o envio dos mesmos.

Desde que o nó permaneça activo, após terminado este processo que impede a existência de colisões aquando da transmissão dos dados, o nó procede ao envio dos dados.

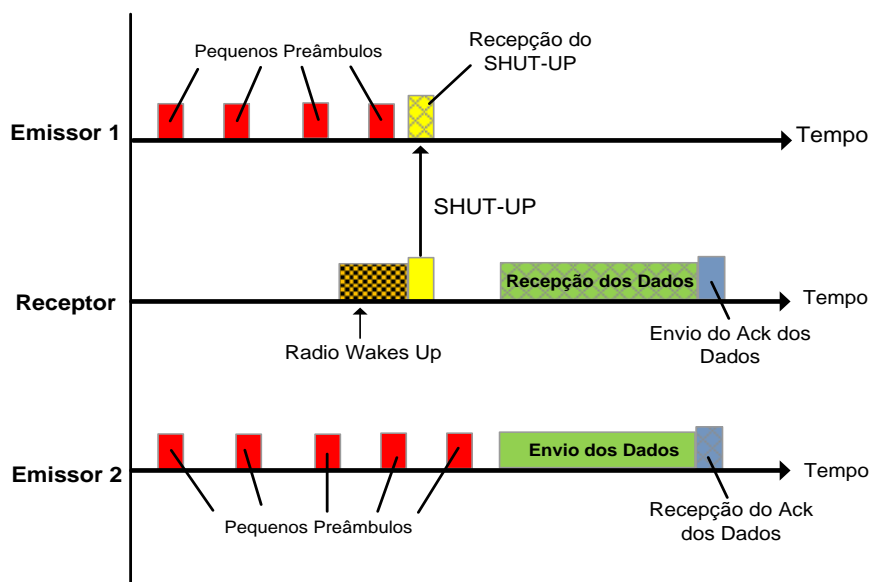


Figura 3. 3 - Utilização do pacote Shut Up

Por outro lado, esta mensagem *Shut Up* também pode ser enviada por um nó síncrono ao receber um preâmbulo, desencadeando o processo de transição entre o modo assíncrono e o síncrono por parte do emissor dos preâmbulos. Este processo de transição é analisado na secção seguinte.

A mensagem *Shut Up* possui um *bit* destinado a indicar se o seu envio se deve a uma colisão ou se foi enviado por um nó síncrono. A recepção do pacote *Shut Up* pode falhar se mais do que um destes pacotes for enviado simultaneamente ou devido a colisões com possíveis preâmbulos, caso existam emissores de preâmbulos activos.

O intervalo aleatório entre os preâmbulos aumenta a probabilidade de todos os emissores receberem o pacote *Shut Up*. No entanto, neste caso a perda de um pacote *Shut Up* poderá dever-se não só à colisão entre estes mas também à emissão simultânea de preâmbulos durante o envio do pacote *Shut Up*.

O protocolo MMH-MAC usa uma aproximação probabilística de modo a proceder ao controle do envio do pacote *Shut Up*. Um nó envia um *Shut Up* com uma probabilidade  $p_{shutup}$  sempre que uma das duas condições apresentadas acima que conduzem ao procedimento de envio deste pacote se verificar.

A probabilidade de ser transmitido com sucesso um pacote *Shut Up* existindo  $N$  nós em condições de proceder ao envio do mesmo é dada por

$$p_s = (1 - p_{coll}) N p_{shutup} (1 - p_{shutup})^{N-1}, \quad (3.1)$$

onde  $p_{coll}$  representa a probabilidade de colisão com um preâmbulo. Há sucesso se não houver colisão e apenas um dos nós transmitir. O número esperado de preâmbulos recebidos até ao envio do pacote *Shut Up* ( $N_{stop}$ ) com sucesso é dado por (3.2), sendo  $M_p$  o número total de preâmbulos enviados antes da transmissão dos dados.

Se a totalidade dos pacotes *Shut Up* enviados falhar, os preâmbulos param em  $M_p + 1$ . Pode observa-se que quando  $M_p$  é grande,  $N_{stop}$  apresenta um mínimo para  $p_{shutup} \approx \frac{1}{N}$ .

$$N_{stop} = (M_p + 1)(1 - p_s)^{M_p} + \sum_{i=1}^{M_p} i p_s (1 - p_s)^{i-1} = \frac{1}{p_s} + \left(1 - \frac{1}{p_s}\right) (1 - p_s)^{M_p} \quad (3.2)$$

O protocolo MMH-MAC recorre à informação local acerca do número de nós vizinhos para calcular  $p_{shutup} = \min\left(\frac{1}{1+neig}; 0.7\right)$ . No entanto, o número de vizinhos conhecidos pode ser diferente do número real de emissores da mensagem *Shut Up* (por exemplo, para uma distribuição espacial de nós não uniforme, ou para vizinhos ainda não conhecidos). Desta forma  $p_{shutup}$  nunca assume o valor 1, porque poderia originar colisões contínuas, sacrificando-se alguma eficiência no valor final de  $N_{stop}$ .

Uma implicação da utilização do pacote *Shut Up* prende-se com o facto de anular a as vantagens de, em transmissões *broadcast*, incluir de um campo nos preâmbulos que indica o tempo até ao envio dos dados. Assim, caso ocorra o envio de um pacote *Shut Up* que “cale” o nó emissor de preâmbulos, ao qual o receptor tinha ajustado o tempo de adormecimento até ao início do envio dos dados, esse envio dos dados não ocorre.

De seguida apresentam-se duas máquinas de estados que ilustram os procedimentos decorridos na emissão e recepção, aquando da realização do envio de dados entre dois nós a funcionar em modo assíncrono.

O emissor procede ao envio de preâmbulos durante, no máximo, 2.2s. Caso essa emissão não seja interrompida, após terminado esse intervalo de tempo, o nó procede ao envio dos dados. Por outro lado, caso essa emissão seja *unicast* e o nó receba um PACK, procede igualmente ao envio dos dados, terminando o envio dos preâmbulos.

Por sua vez, um nó ao receber preâmbulos, contabiliza durante um intervalo de 15ms, o número de emissores existentes. Caso esse número de emissores seja superior a um, então o nó procede ao envio de uma mensagem *Shut Up* destinada a sinalizar qual o nó que deve continuar a enviar preâmbulos, calando-se os restantes. Caso um nó receba várias mensagens *Shut Up*, o nó verifica qual o nó que permanece activo, seleccionando o que possui um menor ID.

Note-se o facto de a referência efectuada na segunda máquina de estados ao algoritmo para efectuar a decisão acerca do envio da mensagem *Shut Up*, *corresponder à aproximação probabilística acima apresentada para proceder ao controlo do envio da mensagem Shut Up.*

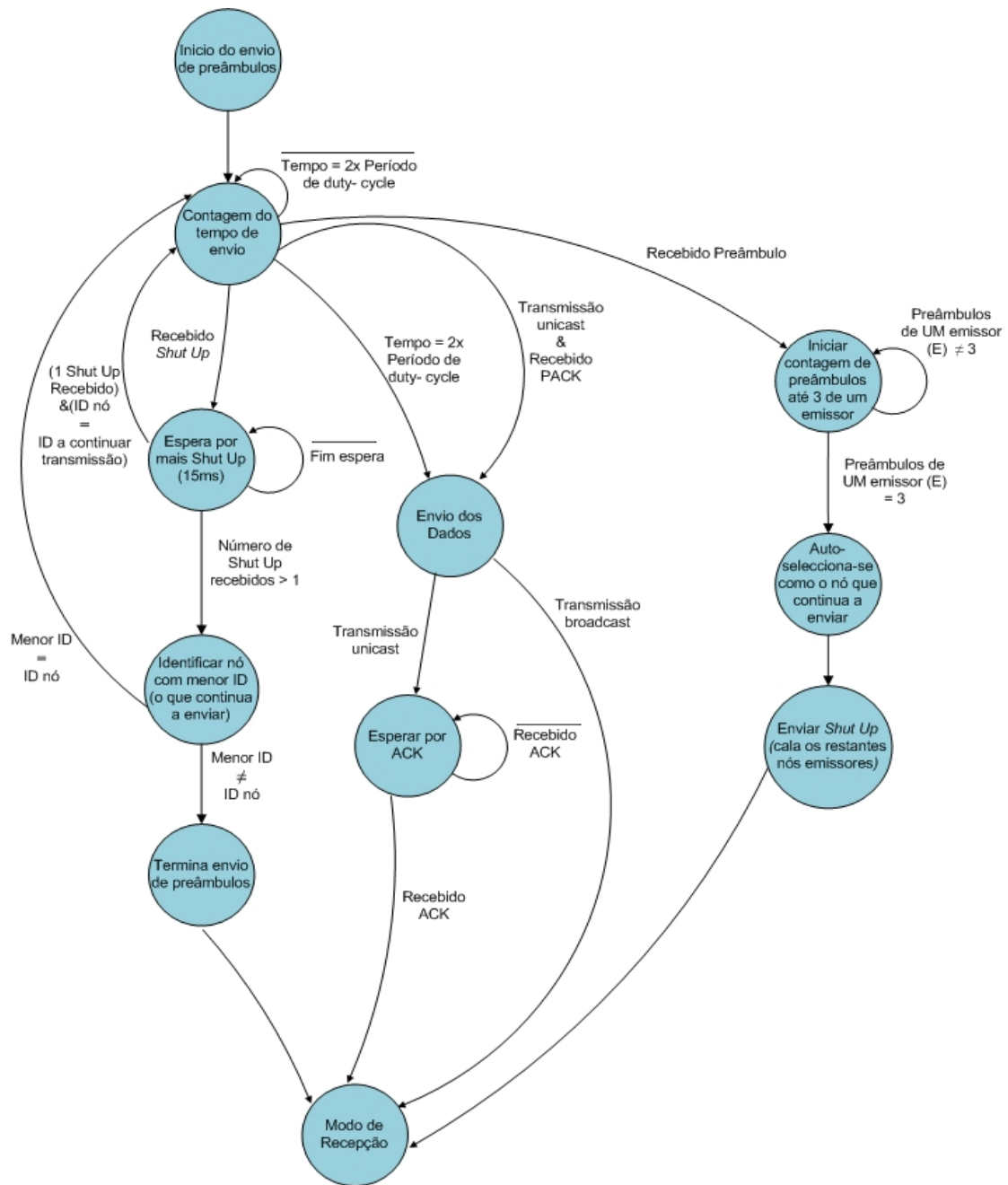


Figura 3. 4 - Transmissão de dados entre nós assíncronos – procedimentos relativos ao emissor

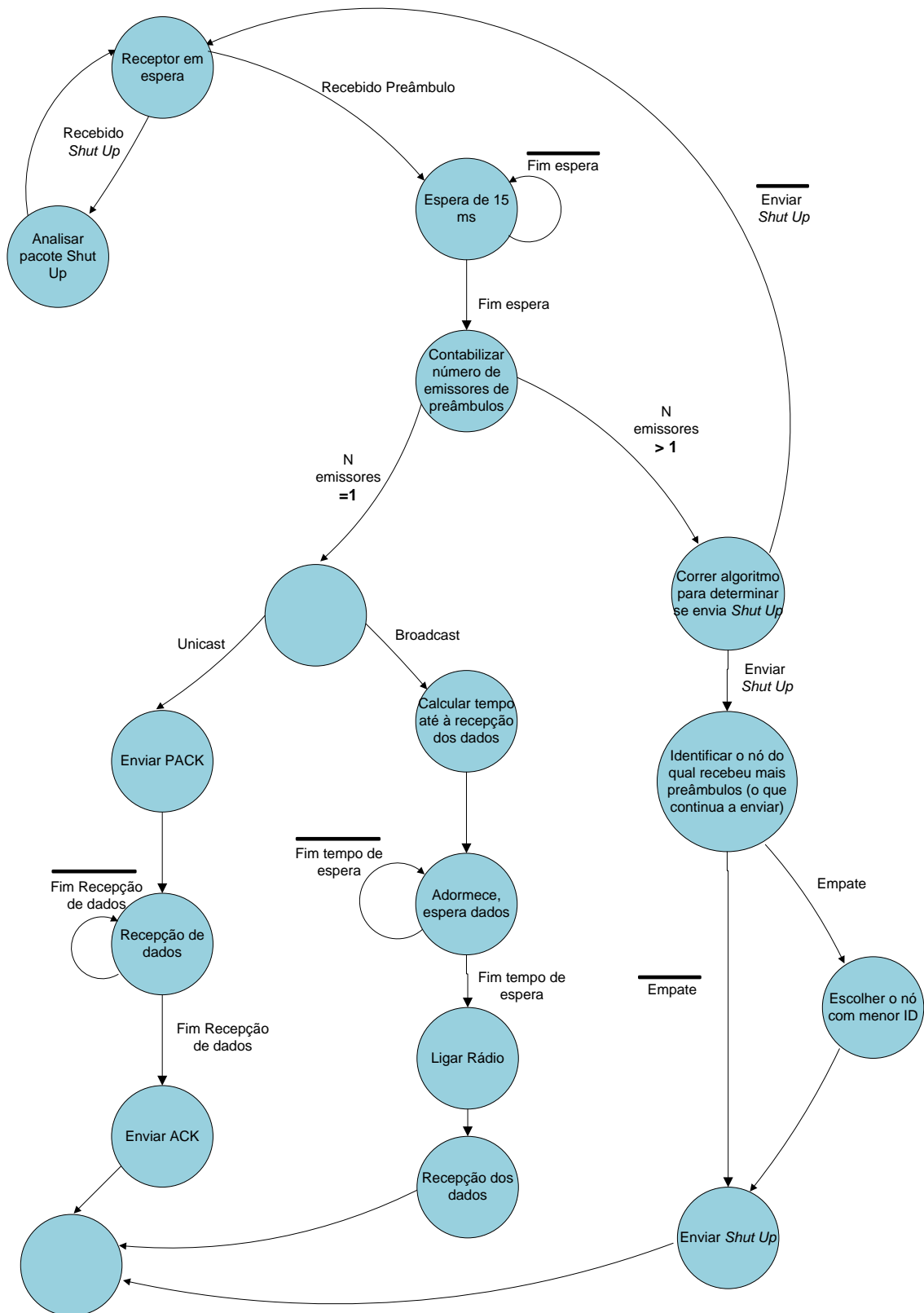


Figura 3. 5 - Transmissão de dados entre nós assíncronos – procedimentos relativos ao receptor

### 3.2.2. Modo Síncrono

No que respeita ao modo síncrono, em contraste com o analisado acima, apresenta como principal objectivo a criação de condições que permitam, de um modo optimizado, a transmissão de um elevado volume de dados entre diversos nós dispersos pela rede, no menor intervalo de tempo possível.

A estrutura considerada consiste num esquema de *slots* baseado numa arquitectura TDMA, permitindo a existência de modelos com ou sem contenção, consoante as definições da aplicação.

Tendo em conta o suporte para mobilidade dos nós existente no protocolo MMH-MAC, o modo síncrono apresenta métodos para descoberta de vizinhos, detecção de ruptura de uma ligação, gestão da sincronização em diversos *clusters* com a possível entrada/saída de nós. Por outro lado, garante-se a existência de um atraso mínimo na agregação de um nó assíncrono a um *cluster* síncrono, ou seja, ajuste do relógio e obtenção de pelo menos um *slot* dedicado. A agregação de um nó assíncrono, à qual está inerente uma transição no modo de funcionamento, é analisada posteriormente.

#### 3.2.2.1. Comunicação

No modo síncrono, os nós utilizam a sequências de *slots*, *macro-slots*, com um período idêntico ao do *duty-cycle* do modo assíncrono (1.1s). Cada *macro-slot* encontra-se dividido em *slots* com uma duração de 100 milissegundos (ms).

Na arquitectura desenvolvida, cada nó possui no mínimo um *slot* público na estrutura de *slots* definida, existindo a possibilidade de alocar *slots* privados com os vizinhos, de modo a proceder à transmissão de dados isenta de colisões.

Relativamente aos *slots* públicos existe a possibilidade de um nó possuir mais de um *slot* deste tipo, já que numa estrutura escalável onde coexistem diversos *clusters*, poderá verificar-se a existência de diversas estruturas de *slots* distintas. Este aspecto conduz à possibilidade de existirem *slots* públicos em diferentes posições na sequência de *slots*, consoante o *cluster*. Os nós que interligam os diferentes *clusters* devem possuir a capacidade de gerir este aspecto, gerindo em paralelo os vários *slots* públicos.

Durante o trabalho desenvolvido considerou-se uma estrutura de *slots* constituída por 11 *slots*, onde se constata a existência de um *slot* público e dez *slots* privados.

### 3.2.2.2. Modelo

O modelo do modo síncrono do protocolo MMH-MAC segue as linhas gerais do referente ao MH-MAC.

Nos *slots* dedicados, destinados ao tráfego entre dois nós, o MMH-MAC usa o método de contenção CSMA, que efectua a escuta do canal durante um intervalo de espera, antes do envio das tramas. Este método permite a diminuição das colisões existentes a 1 *hop*, as quais podem ocorrer, por exemplo, com outros nós na periferia onde se verificam outros modos de funcionamento. Por outro lado, a totalidade dos dados enviados são confirmados com o envio de uma mensagem de confirmação ACK, excluindo-se a necessidade de recorrer ao mecanismo RTS/CTS, já que o mecanismo de alocação de *slots* garante a ausência de problemas relacionadas com a existência da estação escondida.

Com o objectivo de diminuir os consumos energéticos utiliza-se um modelo de contenção semelhante ao existente no protocolo T-MAC [van Dam 2003] nos *slots* dedicados. Os nós usam CCA para monitorizar o canal durante os primeiros 25ms. Caso não se verifique a existência de transmissões o nó transita para um estado de adormecido, o que permite uma poupança energética.

Nos *slots* públicos, um emissor sempre que necessita de enviar alguma mensagem executa um processo CSMA, no qual espera um período temporal aleatório e escuta o canal. Caso o canal se encontre livre, o nó procede ao envio dos dados.

Por outro lado, caso se trate de um *slot* público partilhado por diversos nós, existe a possibilidade de se verificarem dificuldades relacionadas com o problema do nó escondido. Este problema é evitado procedendo ao envio de uma sequência de mensagens *unicast* RTS/CTS antecedendo o envio dos dados, existindo confirmações através de pacotes ACK.

No que diz respeito à estrutura do *slot* público, ilustrada na Figura 3.6, os primeiros 50ms do *slot* público são dedicados a sinalização MAC ou da aplicação, enquanto os segundos 50ms são reservados para todo o tipo de tráfego. Nesta segunda parte do *slot* público, o nó transita para um estado de adormecido caso não exista tráfego nos primeiros 10ms.

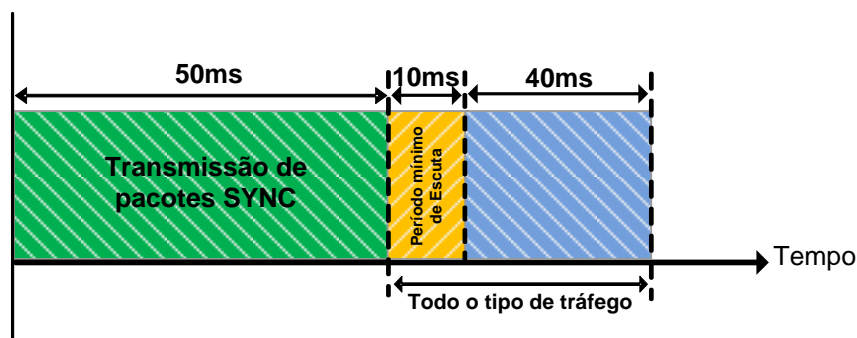


Figura 3. 6 - Estrutura do slot público no protocolo MMH-MAC

Os nós síncronos podem trocar pequenas mensagens usando o mecanismo de CSMA na segunda parte do *slot* público ou podem alocar mais slots dedicados para comunicação *unicast* com os nós vizinhos.

Estes nós síncronos mantêm a sincronização entre si através do envio periódico de uma mensagem SYNC no *slot* público, tal como é ilustrado na Figura 3.6. Com o objectivo de permitir a sincronização de um vizinho assíncrono fora do *slot* público, o protocolo MMH-MAC permite que os pacotes SYNC sejam enviados num slot dedicado. O pacote SYNC identifica o emissor, o período de envio do SYNC, o estado actual da alocação de *slots* e a posição do pacote SYNC relativamente ao início do período de *duty cycle*.

O protocolo MMH-MAC usa as mensagens SYNC para detectar as mudanças na vizinhança, já que cada nó mantêm uma tabela com os SYNC recebidos (tabela de SYNC). Uma nova entrada nesta tabela é adicionada quando uma nova mensagem SYNC é recebido com um valor de RSSI superior ao valor de limiar ( $-87\text{dBm}$  por defeito).

A escolha deste valor deve-se ao estudo realizado em [Srinivasan 2006], o qual refere que para valores de RSSI até  $-87\text{ dBm}$ , os dispositivos TelosB possuem um *Packet Receive Rate* (PRR)  $\geq 85\%$ , descendo significativamente para valores mais baixos

Pelo contrário, uma entrada é apagada se não for recebida nenhuma mensagem SYNC durante 2.5 períodos de envio de SYNC. O período de SYNC é definido na API do protocolo MMH-MAC como um múltiplo do período de *duty cycle* (por defeito 3), permitindo às aplicações adaptar-se às velocidades relativas dos nós.

Um dos aspectos a salientar relativamente ao envio dos pacotes SYNC consiste no facto de o instante de envio do pacote não ser fixo dentro do intervalo de 50ms. Com o intuito de evitar colisões entre pacotes SYNC, o emissor define aleatoriamente um de cinco intervalos possíveis dentro do intervalo inicial de 50ms. Os intervalos são contabilizados a

partir dos 0ms, com um espaçamento de 10ms, situando-se o último nos 40ms. Os intervalos de envio são identificados por um valor inteiro entre zero e quatro (ou entre zero e oito, num *slot* privado), sendo representado no pacote SYNC na forma de três bits. A principal vantagem do envio em um de cinco intervalos fixos, escolhido aleatoriamente, consiste na possibilidade de existirem vários nós síncronos a enviar até um máximo de 5 SYNC em paralelo, por *slot* público. Assim, leva à redução da probabilidade de colisão dos SYNC, a qual está sempre presente numa rede de sensores sem fios na presença de nós móveis.

A alocação de *slots* é controlada pela API do MMH-MAC, recorrendo à troca de duas mensagens: *Request* e *OK*. A mensagem *Request* inclui os *slots* alocados no emissor, os *slots* preferidos para a alocação e o número de *slots* requeridos. O receptor selecciona os *slots* disponíveis dentro dos *slots* preferidos ou próximos destes, de modo a minimizar o atraso na recepção. De seguida procede ao envio de uma mensagem *OK* indicando os *slots* alocados. Os *slots* dedicados podem ser libertados usando a API, ou quando a entrada da tabela de SYNC é apagada devido ao *timeout* por não receber mensagens SYNC.

O vector de *slots* dedicados presente na mensagem SYNC permite a detecção de incoerências na alocação de *slots* entre vizinhos, as quais podem resultar de alocações concorrentes. Os pacotes *Shut Up* são usados para sinalizar a interferência com outros nós nos *slots* alocados.

### 3.3. Funcionamento do Protocolo

Analisados os dois modos de funcionamento do protocolo, torna-se essencial proceder à sua análise, sobretudo em aspectos relacionados com o estabelecimento inicial de sincronismo entre os nós, as transições entre os modos de funcionamento, manutenção de sincronismo entre os nós e gestão dos diversos *clusters* existentes.

A aplicação é responsável por definir os aspectos referentes à definição das características dos modos de controlo, possuindo também a possibilidade de seleccionar o modo de funcionamento. Além disso, a transição entre modos de funcionamento poderá também ocorrer por definição do protocolo MAC.

### 3.3.1. Estabelecimento inicial de sincronismo entre nós Assíncronos

O estabelecimento de sincronismo entre os nós consiste num dos aspectos de maior importância de um modelo híbrido, tal como é o caso do protocolo MMH-MAC. No caso concreto de os nós que estabelecem sincronismo estarem a funcionar no modo assíncrono, foi utilizada uma abordagem que permite o ajuste dos relógios entre os nós e a alocação de *slots* dedicados através de um mecanismo de *handshake*. No entanto, caso existam nós síncronos na vizinhança, o intervalo de tempo necessário para se proceder à sincronização reduz-se consideravelmente como se analisará na subsecção 3.3.2.

#### 3.3.1.1. Protocolo de sincronização

O método de seguida analisado, diz respeito apenas ao caso em que os nós intervenientes na sincronização se encontram a funcionar em modo assíncrono. A situação em que um dos nós se encontra síncrono é abordada na subsecção seguinte.

O protocolo de sincronização inicia-se com envio de uma sequência de preâmbulos com uma duração de duas vezes o período, com o objectivo de acordar os nós vizinhos. De seguida, envia-se um pacote *Hello*, existindo a possibilidade de este ser realizado em *broadcast* ou *unicast*. Isto é, se o nó pretende estabelecer a sincronização com a totalidade da sua vizinhança, envia a mensagem em *broadcast*, mas se o destino for apenas um nó com o qual pretende sincronizar, então envia a mensagem em *unicast*.

No que diz respeito à definição do tipo de sincronização a efectuar pelo nó, esta é determinada pela aplicação.

O pacote *Hello* é constituído por um campo designado por *slot reserve bit* (SRBit), o qual permite indicar se além do estabelecimento de sincronismo, o nó pretende alocar *slots* reservados para futuras transmissões de dados.

No caso do envio do pacote *Hello* em *broadcast*, pretende-se que este seja recebido por todos os nós vizinhos, independentemente do seu modo de funcionamento. Logo, o modo considerado no envio desse pacote é o assíncrono.

No caso de uma transmissão em *unicast*, caso o nó tenha conhecimento que o nó receptor se encontra a funcionar no modo síncrono, usa essa informação ajustando o modo de transmissão do pacote *Hello*. Não possuindo conhecimento de qual o modo de

funcionamento, o nó que desencadeia o processo de sincronização admite o modo assíncrono para o receptor.

Posteriormente à recepção do pacote *Hello*, os nós mantêm o rádio ligado durante o intervalo de tempo máximo de 1.1 segundos, de modo a se proceder à troca das mensagens *Request* e *OK*, descritas abaixo, destinadas a descobrir a vizinhança e alocar *slots* dedicados.

De modo a clarificar a abordagem acima efectuada relativamente ao processo destinado a estabelecer o sincronismo entre os nós, apresenta-se de seguida um exemplo, ilustrado na Figura 3.7, e a consequente análise do mesmo. O nó A com o intuito de adquirir sincronismo com os seus vizinhos, após proceder ao envio de uma sequência de preâmbulos com uma duração de dois períodos para acordar os nós vizinhos, envia uma mensagem *Hello* em *broadcast*. Quando a camada MAC dos nós vizinhos, B e C, recebe o pacote, sinaliza a sua recepção para a aplicação através da API, gerando o evento *SynchronizeReq*. Desta forma, cabe à aplicação a responsabilidade de decidir sobre a aceitação ou rejeição do estabelecimento de sincronismo com o nó. Este evento contém o parâmetro *SRBit* e o endereço do emissor.

Supondo que a aplicação aceita o estabelecimento de sincronismo, esta tem a hipótese de reservar um ou mais *slots* dedicados com o nó. Saliente-se a versatilidade associada ao facto de o nó possuir a capacidade de decidir qual o número de *slots* a reservar, desde que existam recursos disponíveis para satisfazer as suas exigências. Por outro lado, se a aplicação decidir estabelecer sincronismo com o nó mas optar por não reservar qualquer *slot* privado, a comunicação pode ser realizada no slot público.

No caso dual, ou seja, se a aplicação rejeitar o pedido de sincronismo, não se verifica o envio da mensagem *Request*, ficando o nó emissor sem conhecimento da presença do nó receptor da mensagem *Hello*, o que implica o cancelamento do processo de sincronismo.

Caso o sincronismo seja aceite pela aplicação, os nós vizinhos B e C enviam uma mensagem *Request* após esperarem um tempo aleatório. Sendo usado com o intuito de diminuir a probabilidade de ocorrerem colisões entre diversas mensagens *Request*, este intervalo de tempo pode variar entre 0 e 1 segundo.

Após a recepção da mensagem *Request*, ocorre o envio de uma mensagem *OK*, a qual termina o processo de sincronismo, realizado com recurso a um método *three-way-handshake* [Tomlinson 1975].

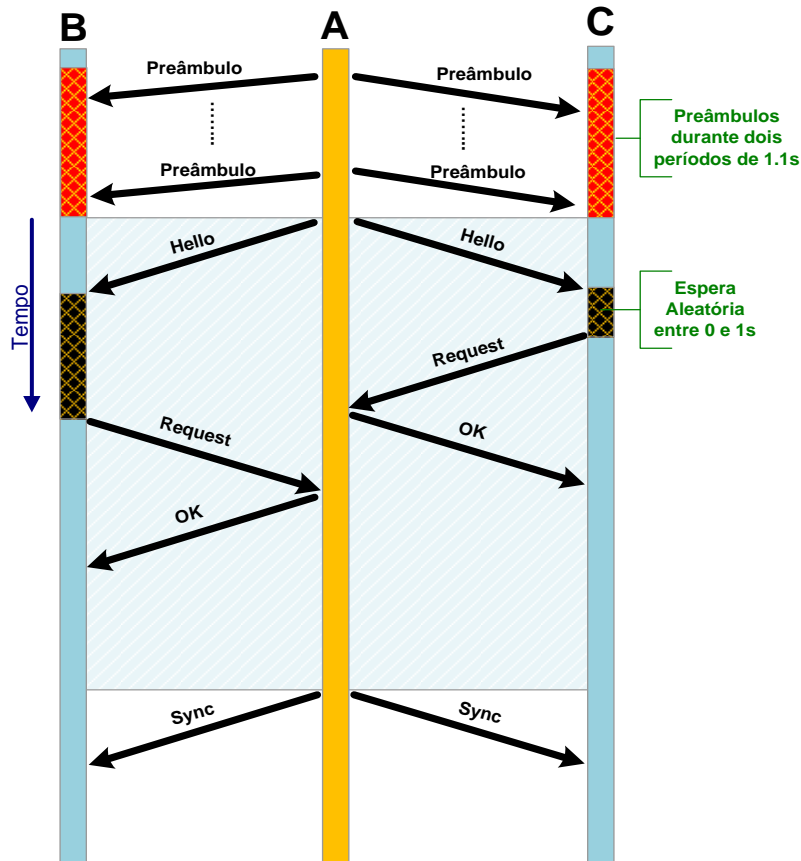


Figura 3. 7 - Exemplo de estabelecimento inicial de Sincronismo entre nós assíncronos

No entanto, caso a mensagem OK não seja recebida pelo destinatário, o nó emissor da mensagem *Request* espera um determinado tempo aleatório e procede ao reenvio da mensagem *Request*. Os limites deste intervalo de espera são 0s e 1s, tal como mencionado anteriormente para o primeiro envio da mensagem *Request*. O tempo máximo permitido para o processo de sincronização, desde a recepção da mensagem *Hello* até ao envio do último OK, é no máximo um período de *duty cycle* (1.1s).

Caso neste intervalo de tempo não se complete a troca *Request/Ok*, os nós (ou o nó), com o quais se estava a processar o estabelecimento de sincronismo mantêm a sincronização de relógios, adiando a reserva de *slots*. De modo a evitar interferências com outras transmissões, utiliza-se o próximo *slot* público para proceder à requisição de um *slot* dedicado, mantendo o método acima mencionado, ou seja, uma troca de mensagens *Request/Ok*. De modo a fornecer ao receptor informação acerca da necessidade de alocar *slots*, o pacote *Request* contém o parâmetro *SRBit*, bem como a lista de *slots* ocupados por vizinhos do nó emissor.

A aplicação define um parâmetro da API que indica o facto de realizar a alocação de slots, ou deixar esse processo a cargo do protocolo de nível MAC. Após a recepção da mensagem *Request*, caso o valor desse parâmetro seja 0, o nó procede à alocação de *slots*, através de uma selecção aleatória de entre o grupo de slots disponível, garantindo-se que num raio de 2 *hops* não existem dois pares de nós a efectuar transmissões de mensagens no mesmo *slot*. Esta condição permite eliminar a ocorrência de interferências nas transmissões e colisões.

Pelo contrário, se o valor do parâmetro definido pela aplicação for 1, o protocolo MMH-MAC informa a aplicação, com o intuito de esta utilizar um algoritmo de alocação de *slots* que cumpra os seus critérios. O protocolo MAC realiza esta operação recorrendo ao evento “*allocateSlots*”, após o qual a aplicação gera um outro evento designado “*allocateSlotsDone*”, informando acerca dos *slots* alocados. Caso a aplicação decida remeter a alocação de *slots* para o nível MAC, atribui ao parâmetro *SlotsMAC*, presente no evento “*allocateSlotsDone*”, o valor 1. Caso contrário o valor desse parâmetro é 0. Estes dois eventos encontram-se descritos na Tabela 3.1.

Atingidos os 1.1s após a recepção ou envio (consoante o nó) da mensagem *Hello*, considera-se terminado o processo de sincronismo na totalidade dos nós envolvidos. No nó emissor, responsável pelo início do processo de estabelecimento de sincronismo, a aplicação toma conhecimento da lista de nós vizinhos que procederam ao sincronismo, bem como os respectivos *slots* dedicados alocados. Nos nós vizinhos a aplicação recebe informação acerca da finalização do processo e do sucesso do mesmo. Terminado este processo, o nó originador do processo de sincronização procede ao envio de uma mensagem SYNC de modo a permitir aos vizinhos a sincronização dos relógios, considerando o seu como a referência.

Nas Figuras 3.8 e 3.9 descrevem-se as máquinas de estados com os procedimentos efectuados pelo nó que desencadeia o processo de sincronização e pelos nós vizinhos.

Quando o nó procede ao envio dos preâmbulos, pode receber uma mensagem *Shut Up*, a qual indica para terminar o envio de preâmbulos. Caso o *bit* que indica o tipo do *Shut*

*Up* possua o valor 1, indica que após o término do envio de preâmbulos, ocorre um processo de sincronismo rápido descrito na subsecção 3.3.2.

Na máquina de estados da Figura 3.8, quando ocorre esta situação indica-se que o comportamento passa a ser modelado pela primeira das máquinas de estados presente nessa subsecção. Por sua vez, caso não se verifique a existência de um processo de sincronismo após a recepção do *Shut Up*, os procedimentos são descritos pela máquina de estados presente na figura 3.8.

No caso do receptor, admite-se na Figura 3.9 que apenas recebe preâmbulos de um emissor, de modo a não sobrecarregar a figura com a análise efectuada na subsecção 3.3.1 relativamente à utilização da mensagem *Shut Up*. Caso o nó esteja síncrono, o comportamento passa a ser modelado pela figura 3.9, referente ao comportamento de um vizinho de um nó que desencadeia a transição assíncrono-síncrono, existindo nós síncronos na vizinhança do assíncrono.

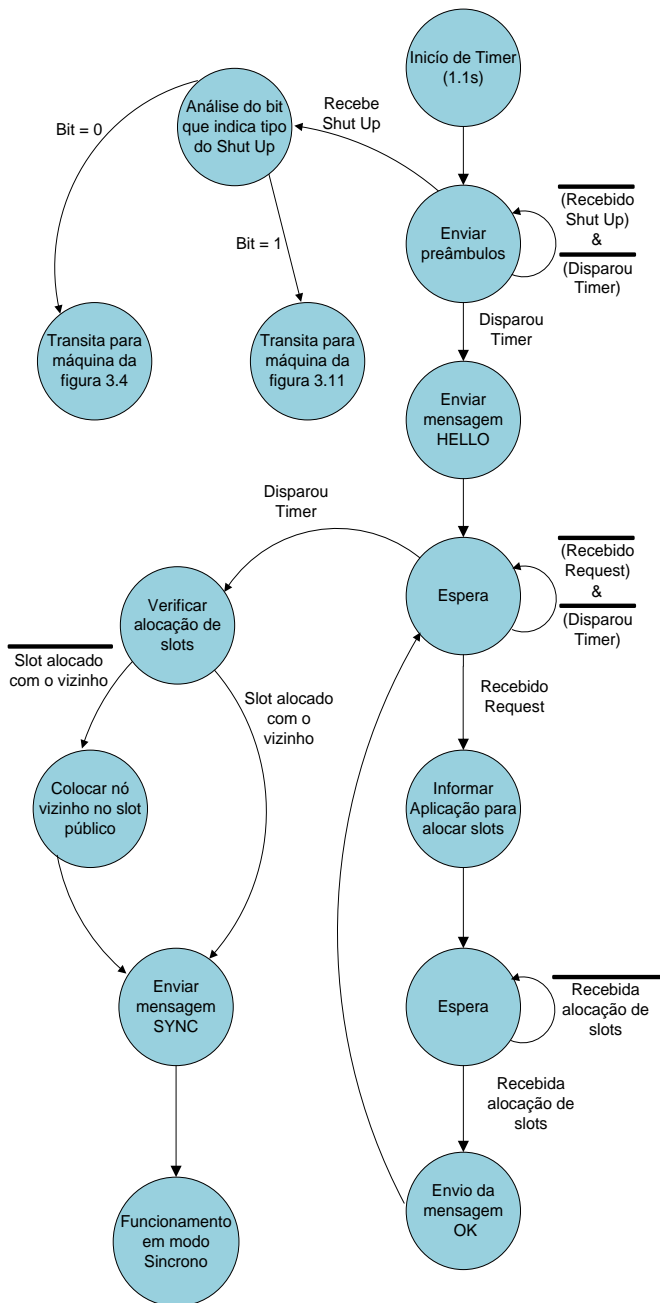


Figura 3. 8 – Procedimentos relativos ao nó que desencadeia o processo de sincronização.

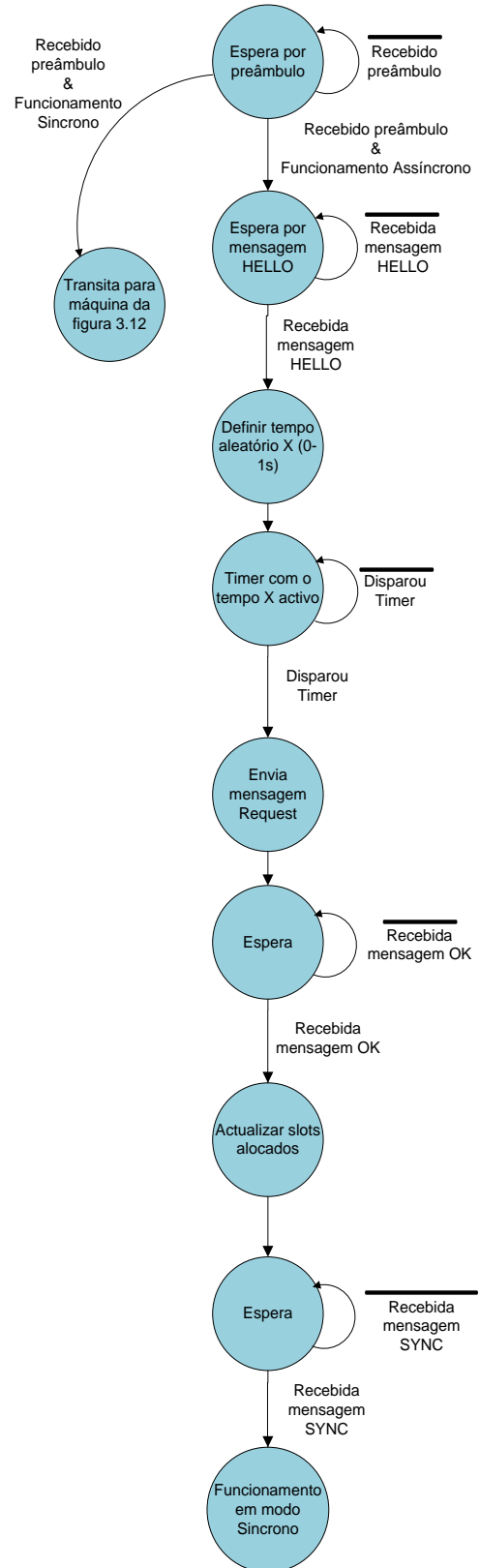


Figura 3. 9 – Procedimentos associados aos nós vizinhos do nó que origina o processo de sincronização

### 3.3.2. Transição Assíncrono-Síncrono

A existência de mobilidade dos nós numa rede de sensores sem fios leva à existência tanto de *grupos de nós* assíncronos como síncronos, na vizinhança de *clusters* síncronos. Deste modo, torna-se essencial a elaboração de métodos relativos à descoberta de vizinhos, diminuição da interferência e detecção de quebra nas ligações, com o intuito de aumentar a eficiência da comunicação. De forma a melhorar a eficiência da comunicação em modo síncrono, o protocolo MMH-MAC utiliza dois métodos para nós assíncronos descobrirem vizinhos síncronos: um método passivo e um outro activo.

O método passivo, caracteriza-se pelo facto de um nó em modo assíncrono aguardar passivamente pela recepção de um pacote SYNC, de forma a ficar sincronizado com o *cluster* síncrono. Por outro lado, o método activo caracteriza-se pelos nós em modo assíncrono usarem continuamente o protocolo de sincronização descrito na secção acima para detectar activamente *clusters* síncronos (ou outros nós em modo assíncrono). O método activo baseia-se, sobretudo, no uso de pacotes *Shut Up* para reduzir a interferência. Os nós síncronos interrompem o envio através de pacotes *Shut Up*, e antecipam o envio de um pacote SYNC. Deste modo, os danos causados na comunicação síncrona devido à transmissão dos preâmbulos são reduzidos, assegurando-se o alinhamento dos períodos em que os nós assíncronos e síncronos estão activos.

Outro dos cenários possíveis consiste na existência de nós síncronos na vizinhança de nós assíncronos, mantendo-se o modo de funcionamento de ambos os tipos de nós. Neste caso, tanto os nós assíncronos como os síncronos mantêm guardados numa tabela de SYNC os *duty cycles* e alocações de *slots* referentes aos últimos nós com os quais estiveram sincronizados. Cada entrada da tabela referente a esses valores é mantida durante um intervalo de tempo  $T_{\text{PERSIST}}$  (por omissão 4 minutos) após a recepção do último pacote SYNC, sendo usada para proceder à sincronização das transmissões de dados com os diversos nós. Após este tempo, os *drifts* de relógio podem tornar esta informação inútil.

Quando o estado do nó de destino é desconhecido, um nó síncrono envia as mensagens utilizando os procedimentos do modo assíncrono: os dados são antecidos de um preâmbulo. A transmissão do preâmbulo inicia-se no *slot* público, de forma à totalidade dos receptores síncronos detectarem que o canal permanece ocupado após o fim do *slot*

público e enviarem um evento de canal bloqueado para a aplicação através da API, o qual permite evitar a perda de dados.

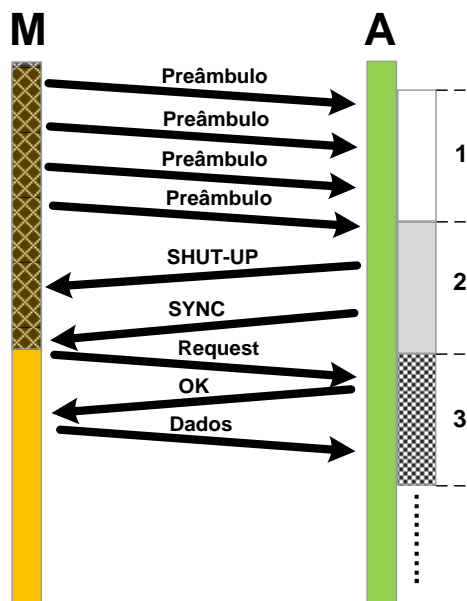
A maioria dos protocolos MAC para redes de sensores sem fios, impõe uma abordagem passiva: os nós móveis aguardam por uma mensagem SYNC e só depois ficam habilitados a iniciar o processo de sincronização. No entanto a abordagem activa permite ao protocolo MMH-MAC reduzir o tempo para proceder à sincronização entre os nós, ou seja, efectuar-se a associação de um nó assíncrono a um *cluster* síncrono, repetindo o protocolo de sincronização continuamente.

Neste protocolo, ocorre uma considerável redução do tempo de associação de um nó móvel assíncrono (M) a um nó síncrono (A), através da interrupção da sequência de preâmbulos recorrendo a uma mensagem *Shut Up*, seguida por um pacote SYNC que permite a rápida sincronização do nó móvel. A Figura 3.10 ilustra o conjunto de mensagens trocadas até à conclusão da transição entre os modos assíncrono e síncrono.

Quando o nó móvel M se move até à vizinhança do nó A, os preâmbulos enviados pelo primeiro são recebidos pelo nó A, podendo causar alguma perda de pacotes nas transmissões de A. No entanto, o facto de a recepção do preâmbulo desencadear o envio da mensagem *Shut Up* e conseqüentemente o nó M terminar o envio dos preâmbulos, leva a uma minimização da interferência causada pelos preâmbulos.

Após a recepção do pacote *Shut Up*, o nó M inicializa um temporizador com uma duração equivalente a duas vezes o período de *duty cycle* (T), à espera do pacote SYNC. Caso o SYNC não seja recebido, é reiniciado o envio dos preâmbulos que antecedem a mensagem *Hello*. O nó A envia um pacote SYNC em *broadcast* no próximo *slot* livre ou no próximo *slot* público, dependendo do que ocorra primeiro. A recepção do pacote SYNC permite o estabelecimento de uma sincronização inicial entre o nó M e o nó A, que pode apresentar um erro elevado pois o atraso da contenção no acesso ao meio não é conhecido.

Após a conclusão deste processo de sincronização inicial, ocorre uma troca *Request/Ok*, a qual permite uma considerável redução do erro na sincronização de relógios, como é analisado posteriormente. Assim, concluído o processo de sincronização, os nós M e A encontram-se aptos a proceder à transmissão de dados nos *slots* dedicados.



**Figura 3. 10 - Processo de Sincronização quando o nó fixo se encontra síncrono**

O intervalo de tempo relativo à associação M-A está condicionado pelo tempo necessário até atingir um *slot* activo e de seguida o intervalo até ao próximo slot dedicado. Relativamente à API do MMH-MAC, esta permite a configuração de *slots* dedicados de escuta do canal, durante os quais os nós ligam o rádio durante os primeiros 10ms com o intuito de detectar transmissões em curso, mesmo que estes estejam livres.

Cada *slot* de escuta significa um custo de aproximadamente 1% do *duty cycle* quando não há *slots* dedicados alocados. No entanto a existência de pelo menos um *slot* de escuta permite a redução do tempo de recepção do primeiro preâmbulo, num cenário onde se constata baixos níveis de carga.

Nas figuras 3.11 e 3.12 apresentam-se duas máquinas de estado que ilustram o comportamento do nó que desencadeia o início do processo de transição assíncrono-síncrono (procurador activo) e de um vizinho desse nó.

O nó procurador procede ao envio de preâmbulos durante um tempo máximo de 2.2s. Durante esse envio, o receptor síncrono, ao receber preâmbulos, procede ao envio de uma mensagem *Shut Up* com o bit que indica o tipo a 1, sinalizando o facto de essa mensagem ser relativa ao início de um processo de sincronismo em que o nó receptor dos preâmbulos se encontra síncrono. A esta mensagem segue-se o envio de uma mensagem SYNC, que ao ser recebida pelo nó que desencadeou o processo de sincronização, permite o ajuste do seu relógio consoante o nó que se encontrava síncrono inicialmente. Após esta troca de mensagens, decorre a alocação de *slots* através de uma troca Request/OK.

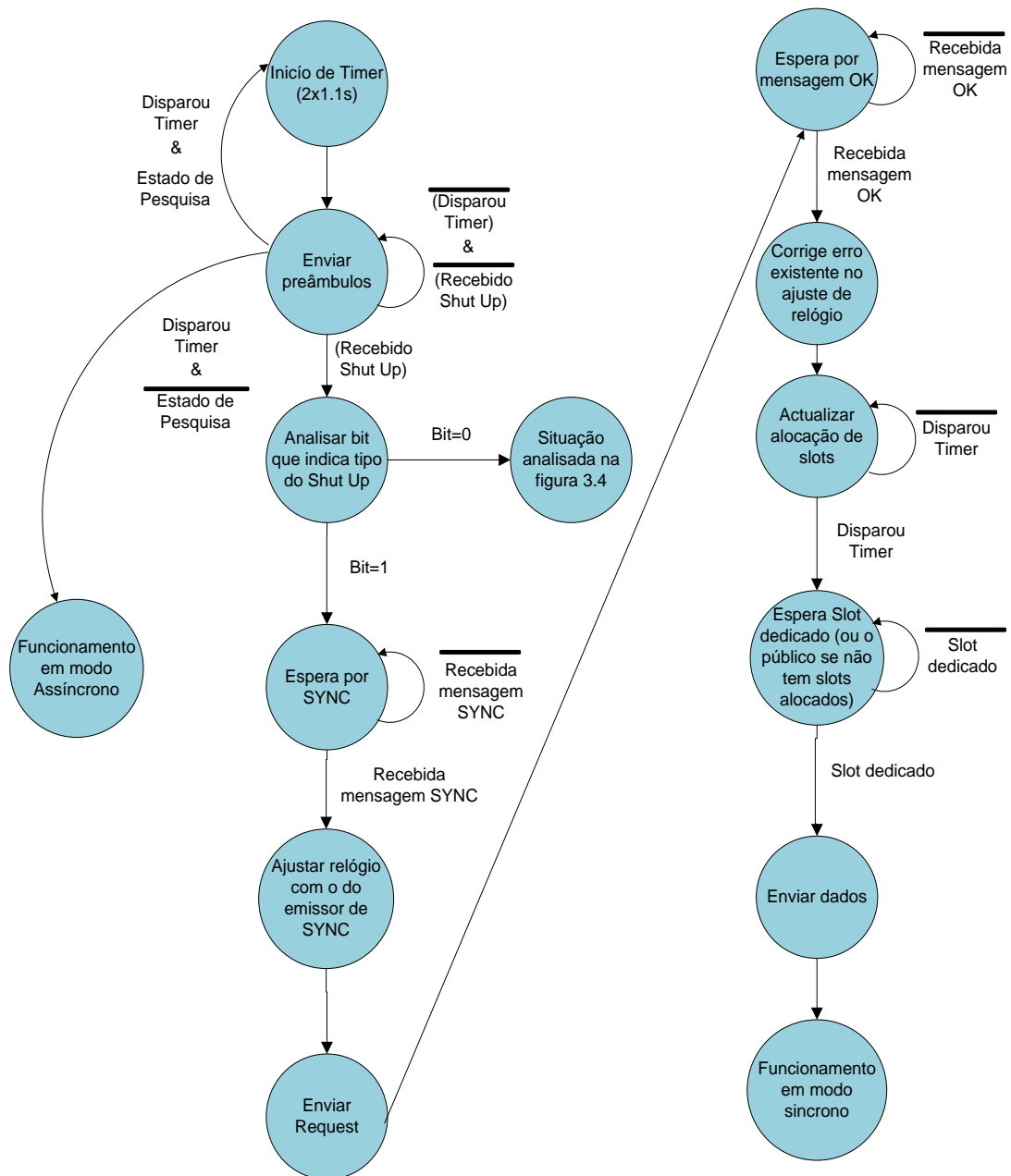


Figura 3. 11 – Transição Assíncrono-Síncrono caso o nó fixo seja síncrono, procedimentos realizados pelo nó assíncrono que desencadeia a sincronização.

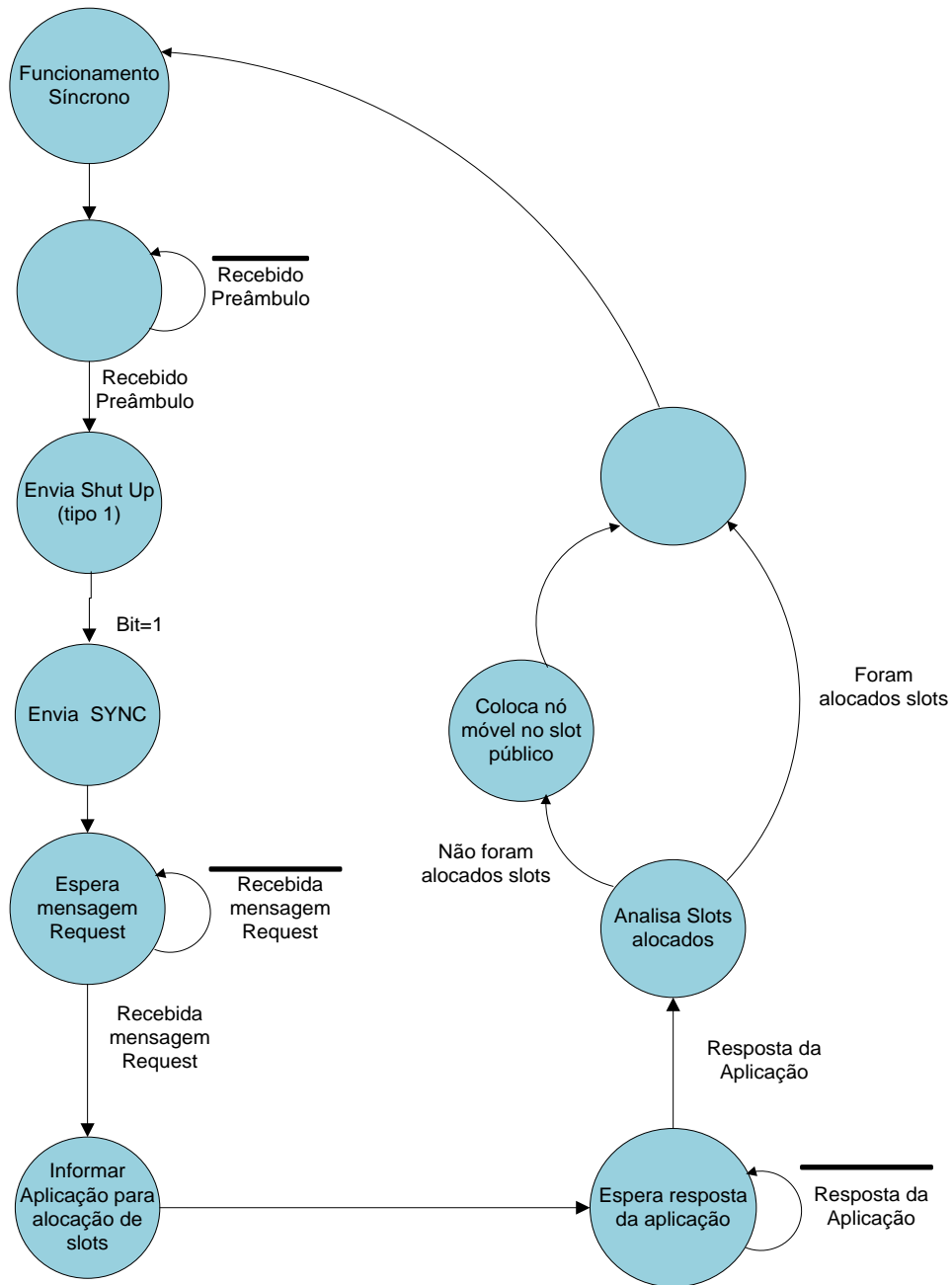


Figura 3. 12 - Transição Assíncrono-Síncrono caso o nó fixo seja síncrono, comportamento de um nó receptor dos preâmbulos enviados pelo nó que desencadeia o processo de sincronização

### 3.3.2.1. Modelo Analítico para o Tempo de Associação

O tempo necessário para um nó assíncrono transitar para o modo síncrono, sincronizando com um *cluster* de nós anteriormente síncronos (tempo total de associação),  $E[T_{sync}]$  depende do *slot* ( $m$ ) onde ocorre a escuta do preâmbulo pela primeira vez e do número de *slots* dedicados activos ( $S$ ). Aplicando o teorema de Bayes para cada *slot* ( $m$ ) e

considerando a probabilidade de um slot estar activo, dada por  $(p_m^{ACT})$ , toma a seguinte forma:

$$E[T_{sync}] = \sum_{m=0}^{10} \left( E[T_{sync}^{ACT} | m] p_m^{ACT} + E[T_{sync}^{IDLE} | m] (1 - p_m^{ACT}) \right) P[slot_m] \quad (3.3)$$

, onde  $E[T_{sync}^{ACT} | m]$  e  $E[T_{sync}^{IDLE} | m]$  são respectivamente o tempo esperado para o início do envio dos dados num *slot* activo ou livre, iniciando a procura no *slot*  $m$ .

Se os nós M e A não se conhecerem, A pode começar a recepção de preâmbulos em qualquer um dos 11 *slots*. Deste modo, a probabilidade relativa ao *slot* onde inicia a recepção é  $P[slot_m] = \frac{1}{11}$ . A probabilidade  $p_m^{ACT}$  é dada por (3.4) e depende do número de *slots* activos ( $S$ ).

$$p_m^{ACT} = \begin{cases} 1 & , m = 0 \text{ (slot público)} \\ \frac{S}{10} & , m = 1 \dots 10 \text{ (slot dedicado)} \end{cases} \quad (3.4)$$

De seguida  $p$  define-se como a probabilidade de um slot dedicado estar activo  $\left(\frac{S}{10}\right)$ . Se a recepção do primeiro preâmbulo ocorrer durante um *slot* activo, o tempo até à ocorrência de uma troca de mensagens de dados num *slot* dedicado é dado por:

$$E[T_{sync}^{ACT} | m] = E[T_{slot}] + \Gamma_m^{IDLE}, \quad (3.5)$$

onde  $E[T_{slot}] = \frac{T}{2}$  consiste no tempo médio até atingir o final do *slot* ( $T$  é o período do *duty cycle*), e o tempo médio até a um *slot* livre, iniciando a análise no *slot*  $m$ , é dado por:

$$\Gamma_m^{IDLE} = (S + \sigma_m(S))p^S + T \sum_{i=0}^S (i - 1 + \sigma_m(i)) (1 - p)p^{i-1} \quad (3.6)$$

$\sigma_m(i)$  tem em conta a passagem pelo *slot* público (0):

$$\sigma_m(i) = \begin{cases} 1, & i + m > 10 \\ 0, & i + m \leq 10 \end{cases} \quad (3.7)$$

Se o primeiro preâmbulo for recebido num *slot* livre, então o tempo de associação inclui o tempo até atingir um *slot* activo.  $\mu_m$  corresponde ao número máximo de slots até alcançar o *slot* 0 ou um *slot* de escuta, iniciando a procura no *slot*  $m$ .  $\mu_m$  é  $10 - m$  para  $m \neq 0$  quando não existem *slots* para escuta do canal. Se o *slot* de escuta se situa na posição  $L$  então  $\mu_m$  é dado por  $L - 1 - m$  para  $0 < m < L$  e aproximadamente  $10 - m$  para  $m \geq L$ . O intervalo de tempo esperado para encontrar um *slot* livre, após um *slot* activo  $i$ , iniciando a procura no *slot* livre  $m$ , é dado por:

$$\Gamma_{m,i}^{ACT} = (1 - (1 - p_{n_{m,i}}) \gamma_{m,i}) \Gamma_{n_{m,i}}^{IDLE} \quad (3.8)$$

Onde  $n_{m,i}$  é o *slot* dedicado onde se fica após avançar  $i$  slots a partir do *slot*  $m$  e  $\gamma_{m,i}$  é a probabilidade de existir um *slot* de escuta imediatamente a seguir a  $n_{m,i}$ . Se o *slot* de escuta existe e não foi alocado, então este pode ser usado para proceder à transmissão de dados. O tempo de associação efectivo depende da probabilidade de encontrar o *slot*  $i$ -ésimo activo, dada por  $p(1 - p)^{i-1}$ , ou de encontrar um *slot* de escuta ou o *slot* público, igual a,  $(1 - p)^{\mu_m}$  quando a procura se inicia no *slot*  $m$ :

$$E[T_{sync}^{IDLE} | m] = E[T_{slot}] + (\mu_m T + \Gamma_{m,\mu_m}^{ACT}) (1 - p)^{\mu_m} + \sum_{i=1}^{max\ m} ((i - 1)T + \Gamma_{m,n_{m,i}}^{ACT}) p(1 - p)^{i-1} \quad (3.9)$$

O modelo apresentado não tem em conta o tempo necessário para a transmissão dos pacotes. Além disso, é extremamente influenciado pela probabilidade  $P[slot_m]$ .

Se todos os vizinhos possuem o seu *duty cycle* alinhado, com um erro menor do que 60ms, então todas as transmissões iniciam-se no *slot* 0 e  $E[T_{sync}] = E[T_{sync}^{IDLE} | 0]$ , reduzindo-se o atraso até ao primeiro *slot* dedicado para transmissão de dados.

Se um nó móvel se move dentro de um *cluster*, o atraso relativo à sua associação é também afectado pelo atraso no *slot* dedicado libertado, gerando uma probabilidade  $p_m^{ACT}$  mais elevada do que a considerada no modelo.

### 3.3.3. Transição Síncrono - Assíncrono

Associada à transição de um nó do modo síncrono para o modo assíncrono está a libertação de todos os recursos associados a este, mantidos pelos restantes nós da vizinhança. De entre esses recursos assumem especial relevância os *slots* dedicados, permitindo assim que outros nós síncronos os reservem com o intuito de proceder à transmissão de dados.

Assim, antes de transitar para o modo assíncrono, o nó envia uma mensagem DSYNC em *broadcast*, com o intuito de informar os nós vizinhos do seu novo modo de funcionamento e da consequente necessidade de libertar recursos.

Esta transição poderá ocorrer devido à acção da aplicação usando o comando reservado para o efeito. Por outro lado, caso não receba uma mensagem SYNC de um determinado nó durante 2.5 períodos de envio de SYNC, liberta a totalidade dos recursos relativos a esse nó. Esta situação apenas acarreta problemas relacionados com o atraso na libertação dos recursos. Após transitar para o modo assíncrono o nó termina o envio das mensagens SYNC. Logo após 2.5 períodos de envio do último SYNC os nós da vizinhança apagam esse nó da tabela de SYNC e libertam os recursos associados à ligação.

O nó procede ao envio da mensagem de dessincronização no primeiro *slot* dedicado que possui com cada nó vizinho, ou caso ocorra primeiro no *slot* público, para acelerar a relocação de recursos. Isto é, caso um nó possua mais do que um *slot* dedicado, o envio da mensagem DSYNC ocorre no primeiro elemento desse grupo.

Por outro lado, o nó procede sempre ao envio da mensagem para dessincronizar, no início dos segundos 50ms do próximo *slot* público, de modo a permitir aos nós que não possuem *slots* alocados com este, conhecer a sua mudança para o modo assíncrono e libertar os recursos associados ao nó.

No que respeita ao intervalo de tempo decorrido até ao nó vizinho receber a mensagem de dessincronização, proveniente de um nó que se encontra no *slot*  $k$ , dois cenários distintos devem ser considerados: o nó receptor não possui *slots* alocados com o emissor, ou tem pelo menos um *slot* alocado.

No primeiro caso, o tempo decorrido até à recepção da mensagem DSYNC é dado por:

$$E[T] = \sum_{k=0}^{10} E[T|k] P(slot_k) \quad (3.10)$$

onde  $E[T|k]$  corresponde ao envio da mensagem no *slot* público e  $P(slot_k)$  diz respeito à probabilidade do *slot* onde se efectua a recepção.

Considerando  $T$ , a duração de cada *slot*, o tempo a decorrer até se chegar ao *slot* público é dado por:

$$E[T|k] = (11 - k) T \quad (3.11)$$

Note-se o facto de nas expressões apresentadas se desprezar o tempo de propagação da mensagem.

Finalmente, substituindo 3.11 em 3.10 e sabendo que  $P(slot_k) = \frac{1}{11}$ , o tempo decorrido até à recepção da mensagem de dessincronização vem:

$$E_p[T] = \frac{T}{11} \sum_{k=0}^{10} 11 - k = 6 T \quad (3.12)$$

Relativamente ao segundo caso, em que o nó vizinho possui pelos menos um *slot* alocado com o nó a transitar para o modo assíncrono, designando por  $s$  o primeiro *slot* alocado com esse vizinho, o tempo decorrido até à recepção da mensagem de dessincronização é novamente dado pela equação 3.10. As diferenças existentes, prendem-se com os valores relativos a  $E[T|k]$ , cuja forma é apresentada na equação seguinte.

$$E[T|k] = \begin{cases} (s - k) T - \frac{T}{2} & , k < s \\ 0 & , k = s \\ (11 - k) T & , k > s \\ 0 & , k = 0 \end{cases} \quad (3.13)$$

Assim, no caso do modelo considerado no protocolo MMH-MAC, o tempo máximo de espera pela mensagem de dessincronização DSYNC é de 1050ms. Isto é, se a decisão de transição para o modo assíncrono ocorrer no início do primeiro *slot* da sequência e o nó receptor não possuir *slots* dedicados com o emissor, o nó informa acerca da transição apenas após os restantes 10 *slots* dedicados e metade do público. Esta situação encontra-se ilustrada na Figura 3.13, onde um nó T inicia no *slot*  $k$  o processo de transição para o modo assíncrono. No *slot* dedicado  $s$ , procede-se ao primeiro envio da mensagem DSYNC de modo a acelerar o processo de relocação de recursos.

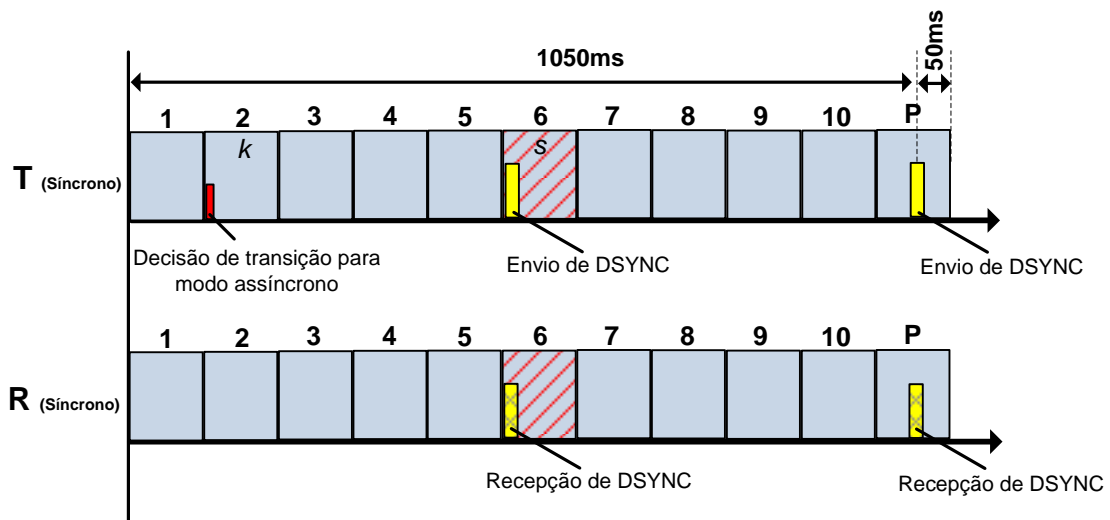


Figura 3. 13 – Transição Síncrono-Assíncrono

Após o nó transitar para o modo assíncrono é gerado um evento que informa a aplicação acerca do novo modo de funcionamento.

Se o objectivo for uma transição global na rede do modo síncrono para o assíncrono, este processo pode realizar-se *hop a hop*, com o atraso temporal máximo considerado acima.

### 3.3.4. Gestão de Clusters

O protocolo MMH-MAC procura estabelecer métodos para lidar com a gestão de *clusters*, ou seja, grupos de nós com uma referência temporal comum.

Foram definidos métodos de sincronização entre os nós quando todos se encontram assíncronos, apresentados na subsecção 3.3.1. Além disso, como a existência de mobilidade aumenta a probabilidade de coexistirem nós assíncronos na vizinhança de *clusters* síncronos, desenvolveu-se um método para reduzir o tempo de associação desses nós ao *cluster* síncrono, analisado em 3.3.2.

No que se refere aos *clusters* síncronos propriamente ditos, os nós situados na borda destes, acordam em ambos os instantes durante uma fase inicial, ou seja, seguem ambos os *schedules*. Este método é idêntico ao utilizado no protocolo S-MAC [Ye 2004].

No entanto, quando a ligação se torna estável o protocolo MMH-MAC procede à junção dos dois *clusters*, sendo adoptado o *duty cycle* do nó cujo número utilizado como seu identificador exclusivo ( $Q$ ) é maior. Uma ligação é considerada estável quando o número de

SYNC recebidos ultrapassa o limiar SYNC\_STAB\_THRESH. O valor deste parâmetro é por defeito 15.

Analogamente ao protocolo ABR [Toh 1996], existe um contador de *beacons* (SYNC) recebidos, sendo o valor desse contador guardado na tabela de SYNC.

Cada nó admite como referência temporal o nó com o qual mantém uma ligação com maior estabilidade, ou seja, durante a qual ocorreu a recepção de um maior número de mensagens SYNC. Caso um nó não receba nenhuma mensagem SYNC durante 2.5 períodos de envio de SYNC, apaga a entrada desse nó da tabela de SYNC e liberta todos os recursos associados a ao nó (*slots* alocados, etc).

Tanto os nós assíncronos como os síncronos, guardam na tabela de SYNC, os *duty cycles* e alocações de *slots* referentes aos últimos nós com os quais estiveram sincronizados, mantendo cada entrada da tabela durante  $T_{\text{PERSIST}}$  (por omissão 4 minutos) após a recepção do último pacote SYNC. Após este tempo, os *drifts* de relógio podem tornar a informação inútil.

### 3.3.5. Manutenção do Sincronismo entre os nós

Durante o funcionamento síncrono dos nós, os seus *duty cycles* devem estar alinhados. No entanto, a existência de *drifts* de relógio conduz à existência de discrepâncias entre os diversos *duty cycles* e à conseqüente diminuição do *throughput*. Assim, torna-se indispensável a definição de um método destinado a compensar os desvios verificados nos diversos nós.

No modo de funcionamento síncrono os nós enviam periodicamente nos primeiros 50ms do *slot* público uma mensagem SYNC que possui informação acerca do instante em que se procedeu ao seu envio (em múltiplos de 10 ms). Deste modo, os receptores desta mensagem, com base nesta referência temporal, ajustam os seus relógios garantindo o alinhamento entre os *duty cycles* dos nós. No entanto, o tempo de acesso ao meio depende da carga na rede, e vai variar para diferentes pacotes.

Cada nó admite como referência temporal o nó com o qual mantém uma ligação com maior estabilidade. Caso a referência temporal do nó seja eliminada pelas razões analisadas na subsecção anterior, o nó passa a assumir como referência o nó cuja entrada

na sua tabela de SYNC apresenta um maior número de SYNC recebidos, ou seja, uma ligação mais estável.

A compensação do tempo de acesso ao meio durante o envio do pacote SYNC, é realizada incluindo o valor do tempo de acesso do pacote SYNC anterior no pacote seguinte. O tempo é medido imediatamente após o envio do pacote [Rentel 2007]. O pacote SYNC possui um identificador ID\_SYNC incrementado a cada pacote enviado, que permite ao receptor identificar se o tempo de acesso contido no pacote recebido corresponde ao último recebido em termos sequenciais. Isto é, caso uma mensagem SYNC se perca, o nó ao receber o próximo SYNC descarta a referência temporal, já que o tempo de transmissão presente neste pacote diz respeito ao SYNC perdido.

Após a recepção de um pacote SYNC, o nó guarda a referência temporal do emissor. Ao receber o próximo SYNC, o nó com base na informação do tempo de transmissão do anterior e da referência temporal guardada, ajusta o relógio. Durante o estabelecimento de sincronismo inicial, a mensagem OK também contém o tempo de acesso do último SYNC, garantindo a correcção rápida dos erros existentes.

O algoritmo acima descrito permite a correcção dos *drifts* de relógio nos nós, utilizando apenas tipos de pacotes que já eram utilizados para outros fins, não introduzindo nenhum tipo novo. Além disso, garante robustez perante a perda de mensagens SYNC e a saída da referência temporal da vizinhança do nó.

O fluxograma presente na Figura 3.14 descreve o algoritmo utilizado para corrigir os *drifts* de relógio existentes. Se o nó recebe uma mensagem SYNC proveniente do nó considerado sua referência temporal, ou seja, o nó com o qual mantém a ligação de maior estabilidade, guarda o estado do relógio do emissor. Após proceder à verificação do ID dessa mensagem SYNC, caso conclua que a mensagem é a esperada de acordo com essa mesma sequência, então o nó corrige o erro inerente ao tempo de transmissão da mensagem SYNC anterior, com base no valor desse parâmetro presente na actual.

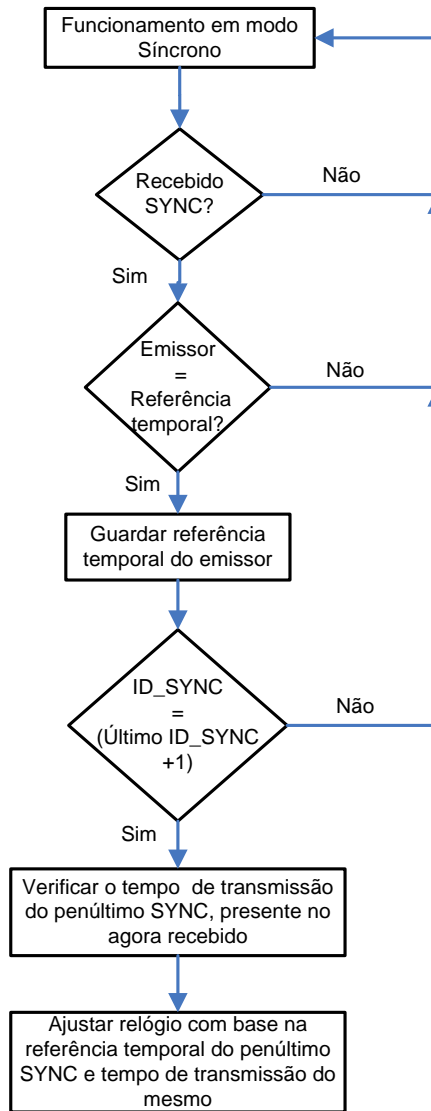


Figura 3.14 – Algoritmo utilizado para correcção dos *drifts* de relógio existentes nos nós

### 3.4. Interface de programação da aplicação

Nesta dissertação, foi desenvolvida uma interface de programação que permite a optimização entre camadas (*cross-layer*), entre o protocolo MMH-MAC e as aplicações. Desta forma, a existência de comando e eventos (alguns deles herdados do protocolo MH-MAC) conhecidos por ambas as camadas permite um considerável aumento da versatilidade inerente ao protocolo, já que pode ser configurado pela aplicação consoante as necessidades do momento.

Note-se o facto de os comandos consistirem em ordens dadas pela aplicação ao nível MAC de modo a que este proceda à execução de determinadas tarefas. Pelo contrário, os

eventos destinam-se à sinalização de determinada ocorrência, tal como a terminação da tarefa requerida anteriormente pela aplicação através do uso de um comando.

Os processos de realização de tarefas desencadeadas pelos comandos encontram-se embutidos no próprio protocolo, enquanto as funções de tratamento dos eventos são programadas e definidas nas aplicações.

Na Tabela seguinte são apresentados os diversos comandos e eventos, relativos à interface de programação da aplicação (API) do protocolo MMH-MAC.

| <b>Tipo</b>    | <b>Designação</b>                 | <b>Descrição</b>  |
|----------------|-----------------------------------|---|
| <i>Comando</i> | <b><i>synchronize</i></b>         | Permite à aplicação desencadear o estabelecimento de sincronismo com um ou mais nós vizinhos.   |
| <i>Evento</i>  | <b><i>synchronizeDone</i></b>     | Indica a conclusão do processo de sincronização com um ou mais vizinhos foi e conseqüente início do modo de funcionamento síncrono.                   |
| <i>Comando</i> | <b><i>assynchronize</i></b>       | Utilizado para iniciar a transição para o modo de funcionamento síncrono.   |
| <i>Evento</i>  | <b><i>assynchronizeDone</i></b>   | Indica a conclusão da transição para o modo de funcionamento assíncrono.  |
| <i>Evento</i>  | <b><i>synchronizeReq</i></b>      | Ocorrência da sinalização de pedido de sincronização por parte de um vizinho.   |
| <i>Comando</i> | <b><i>continueReq</i></b>         | Comando com a decisão da aplicação referente ao pedido de sincronismo por parte de um vizinho. Esta poderá aceitar a sincronização ou então rejeitar. |
| <i>Evento</i>  | <b><i>synchronizeReqDone</i></b>  | Sinal de conclusão da aquisição de sincronismo com um nó pertencente à vizinhança.  |
| <i>Comando</i> | <b><i>stopSynchronize</i></b>     | Comando para inicializar a dessincronização com um nó específico.   |
| <i>Evento</i>  | <b><i>stopSynchronizeDone</i></b> | Evento a sinalizar a finalização da dessincronização com um determinado nó.   |
| <i>Evento</i>  | <b><i>synchronizeKilled</i></b>   | Permite informar a aplicação, que o sincronismo com um determinado nó foi interrompido.   |
| <i>Comando</i> | <b><i>isSynchronous</i></b>       | Comando destinado a questionar a camada MAC se esta se encontra sincronizada com determinado nó.  |

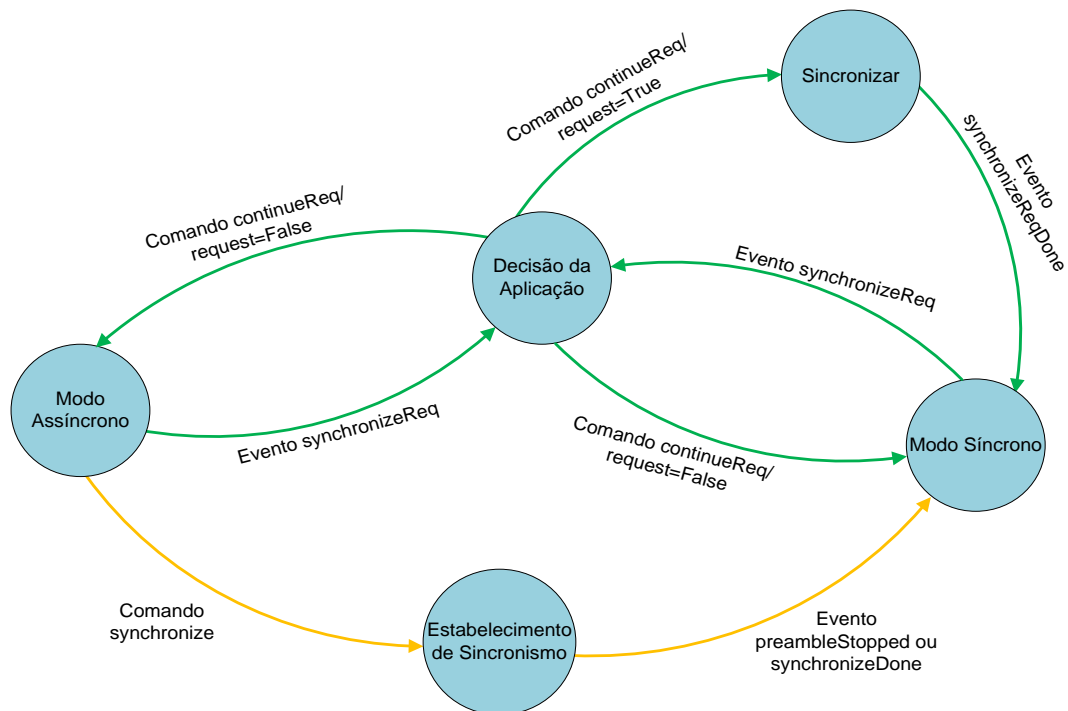
|                |                                      |  |
|----------------|--------------------------------------|--|
| <i>Evento</i>  | <b><i>preambleStopped</i></b>        | Indica a interrupção da aquisição de sincronismo.  |
| <i>Comando</i> | <b><i>send</i></b>                   | Permite enviar uma mensagem em modo assíncrono.  |
| <i>Evento</i>  | <b><i>sendDone</i></b>               | Indica a conclusão do envio de uma determinada mensagem.   |
| <i>Comando</i> | <b><i>sendLBcontrol</i></b>          | Comando para envio de uma mensagem de controlo no modo de funcionamento síncrono.  |
| <i>Comando</i> | <b><i>getSlotsSynchronize</i></b>    | Requisita o estado actual da alocação de <i>slots</i> estabelecida.  |
| <i>Evento</i>  | <b><i>slotsSynchronize</i></b>       | Evento com a lista pedida sobre a atribuição de <i>slots</i> por parte da aplicação.   |
| <i>Comando</i> | <b><i>whichSlotsRunning</i></b>      | Fornece informação acerca do <i>slot</i> actual, caso se verifique o funcionamento síncrono do nível MAC.  |
| <i>Comando</i> | <b><i>getMissingTimeNextSlot</i></b> | Indica o intervalo de tempo decorrente até ao início do próximo <i>slot</i> .  |
| <i>Evento</i>  | <b><i>allocateSlots</i></b>          | Requer à aplicação que proceda à alocação de <i>slots</i> .  |
| <i>Evento</i>  | <b><i>allocateSlotsDone</i></b>      | Evento que indica o término do processo de alocação de <i>slots</i> e quais os <i>slots</i> alocados (se existem). Parâmetro SlotsMAC=1 indica alocação de <i>slots</i> a ser efectuada pelo nível MAC; SlotsMAC=0, alocação realizada pela aplicação. |
| <i>Evento</i>  | <b><i>syncState</i></b>              | Informa a aplicação acerca do estado actual das mensagens SYNC recebidas dos diferentes nós.   |

**Tabela 3. 1 – Comandos e eventos relativos à API do protocolo MMH-MAC**

Os comandos e eventos podem ser conjugados entre si de modo a proceder à realização de determinadas acções, como por exemplo o estabelecimento de sincronismo. Na figura seguinte ilustra-se a sequência de comandos e eventos usados para proceder ao estabelecimento de sincronismo entre nós, de um ponto de vista do nó que desencadeia esse processo ou de um nó que recebe um pedido para estabelecer sincronismo.

Com cor laranja encontra-se representada a sequência relativa ao nó que desencadeia o estabelecimento de sincronismo. Inicialmente utiliza o comando “synchronize” para iniciar o processo de sincronismo. Quando se conclui o processo de sincronismo, activa o evento “synchronizeDone”. Após este ponto, o nó passa a funcionar em modo síncrono.

Por outro lado, um nó em modo assíncrono ou síncrono poderá receber um pedido de sincronismo, questionando de seguida a aplicação sobre a continuidade ou não do processo, recorrendo ao comando “synchronizeReq”. Caso a aplicação rejeite o pedido de sincronismo, o nó mantém-se no estado anterior. Pelo contrário, se for aceite, desencadeia-se o processo de sincronização, sendo emitido após o seu término, o evento “synchronizeReqDone”. Esta sequência de acções relativas ao nó receptor do pedido de sincronismo, apresenta-se com cor verde na figura seguinte.



**Figura 3. 14 - Comandos e Eventos utilizados no estabelecimento de sincronismo entre os nós**



## Capítulo 4.

# Análise Do Desempenho

---

Realizada a descrição do protocolo proposto nesta dissertação, neste capítulo avalia-se o seu desempenho através da realização de diversos testes. São focados aspectos relacionados com o ambiente de teste e os diversos parâmetros analisados.

O capítulo encontra-se dividido em três secções. Na primeira secção apresenta-se o ambiente de teste utilizado, focando-se sobretudo no simulador. A segunda faz referência aos diversos pontos de análise do desempenho do protocolo MMH-MAC. Na última secção são apresentados os resultados obtidos, referentes aos vários pontos de análise.

### 4.1. Ambiente de Teste

Na elaboração deste trabalho foi utilizado o ambiente de desenvolvimento Linux, Fedora Core 7. Nesta dissertação recorreu-se ao sistema operativo TinyOS [Hill 2000] [Levis 2003b] [Levis 2007], optando-se por trabalhar na versão TinyOS 2.0 *stable*. A versão usada foi a TinyOS 2.0.2.

No entanto, constatou-se que o simulador existente, TOSSIM [TOSSIM 2007], não suportava a simulação de mobilidade em redes de sensores sem fios.

Além deste aspecto, outras funcionalidades disponíveis para o *hardware* TelosB não existiam no simulador TOSSIM. Entre as principais funcionalidades inexistentes destacavam-se [Levis 2003a] [Levis 2003b]:

- Modo de funcionamento LPL - permite o uso dos mecanismos de escuta do meio através do CCA;
- A utilização da *flash* de 1MB;
- A aquisição do valor de RSSI em cada mensagem recebida;
- Possibilidade de estimar o valor de energia consumida pelos nós durante a simulação efectuada;
- Suporte para mobilidade dos nós.

Tendo em conta a inexistência destas funcionalidades no simulador, foi desenvolvido o suporte para mobilidade durante a elaboração do trabalho. No caso das outras funcionalidades, foram utilizadas extensões ao simulador realizadas noutras teses de mestrado de 2006/2007, que foram concluídas já depois do arranque desta dissertação.

#### **4.1.1. Simulador TOSSIM**

Como mencionado, nesta dissertação foram usadas várias funcionalidades desenvolvidas anteriormente no âmbito de outras dissertações.

A implementação do modo de funcionamento LPL do dispositivo de rádio CC2420 foi realizada anteriormente pela dissertação responsável pelo desenvolvimento do protocolo MH-MAC [Pereira 2007]. De modo a implementar este mecanismo, não foi utilizado um método para optimização dos consumos através do recurso ao método CCA para escutar o canal. A opção recaiu em desligar e ligar o rádio nesse intervalo de tempo. O mecanismo de CCA apresenta uma elevada utilidade para “escutar” o canal e concluir se já existe uma transmissão por parte de outro nó, evitando os consumos desnecessários de se ligar o rádio. Assim, os consumos apresentados nesta dissertação possuem um valor acima do real, pois a aproximação adoptada, conduz a que se considere um consumo de energia acima do efectivamente verificado.

Relativamente à obtenção do valor do RSSI associado a cada mensagem recebida, definiu-se o valor do ganho associado à ligação entre cada par de nós como sendo o valor do RSSI. Esta aproximação permitiu estimar o valor desta grandeza, colmatando assim a ausência desta funcionalidade no simulador TOSSIM original.

Na obtenção de valores aleatórios, por omissão, o simulador usa sempre a mesma semente. Logo, o resultado obtido nas diferentes simulações para os diversos valores aleatórios é idêntico. Este aspecto leva a que os valores obtidos que supostamente seriam aleatórios, na realidade não o são. Com o objectivo de colmatar esta lacuna, procedeu-se à alteração da semente que está na base dos números aleatórios gerados, sendo cada nó dependente do valor dos milissegundos do relógio local. Esta opção vai de encontro ao verificado nos sensores TelosB onde o valor da semente é gerado a partir dos relógios disponíveis, sendo na quase totalidade dos casos diferente para os diversos nós.

O simulador TOSSIM não suportava medições de energia para os dispositivos TelosB. Duas soluções poderiam ter sido adoptadas para colmatar esta lacuna. Em primeira instância poderia ser usada uma ferramenta externa para proceder à análise dos ficheiros de simulações (*logs*) gerados para cada simulação e verificar quais os tempos de funcionamento do MCU, rádio, *flash* e sensores. Mas a elevada dimensão destes *logs* (ordem dos gigabytes) obrigaria a um considerável intervalo de tempo para uma ferramenta externa proceder à sua análise.

Assim, tendo em conta a desvantagem da solução anterior, a escolha recaiu sobre uma segunda opção. Foi desenvolvida uma aplicação que segue as linhas gerais do PowerTOSSIM existente para TinyOS-1.x [Shnayder 2004] e do PowerTOSSIM para TinyOS-2.x, o qual durante a elaboração deste trabalho apenas suportava o *hardware* Mica2. O simulador TOSSIM passou a incluir uma ferramenta de medição em tempo real do consumo efectuado por cada nó, seguindo o modelo de energia dos sensores TelosB. No final de cada simulação define-se um ficheiro para cada nó com as informações acerca do consumo de energia realizado pelo nó. Neste ficheiro é apresentado o valor total da energia consumida, assim como a descrição individual do consumo realizado pelo MCU, rádio, *flash*, sensores. Os valores das correntes para o cálculo da energia são os apresentados na secção 4.3.2.

#### 4.1.1.1. Suporte de Mobilidade no simulador TOSSIM

Nesta dissertação pretendeu-se desenvolver um protocolo destinado a redes de sensores sem fios móveis, logo seria indispensável proceder-se à análise do desempenho do mesmo na presença de mobilidade dos nós. No entanto, o simulador TOSSIM não disponha desta funcionalidade, o que inviabilizava a simulação do trabalho desenvolvido. Foi realizada na linguagem Python uma aplicação que permite realizar a simulação dos movimentos dos nós na rede. Esta aplicação efectua a leitura dos instantes em que ocorrem alterações das ligações e de quais os nós considerados nessas alterações, a partir de um ficheiro de texto destinado à especificação dos movimentos ocorridos.

Com base nos parâmetros definidos nesse ficheiro de texto, as alterações do estado da rede podem ocorrer no instante exacto definido pelo utilizador ou com uma oscilação aleatória relativamente a esse instante. Desta forma, permite-se a simulação de mobilidade em instantes fixos e definidos, ou de mobilidade em instantes aleatórios e desconhecidos *a priori*. A sintaxe do ficheiro de especificação de movimento é descrita no Apêndice A.

Por exemplo, no caso de um modo de funcionamento síncrono, caso o objectivo seja um nó mover-se num instante desconhecido no interior da sequência de *slots*, torna-se indispensável que esse movimento ocorra de uma forma aleatória. Caso contrário, o cenário existente seria idêntico para todas as simulações, inviabilizando o teste da totalidade das situações existentes na rede e, conseqüentemente, os resultados não seriam demonstrativos do desempenho real do sistema.

Após a leitura e processamento do ficheiro de texto são conhecidas as diversas alterações a efectuar à rede inicial e quais os instantes nos quais essas mudanças ocorrem. O restante algoritmo que implementa o suporte de mobilidade no simulador TOSSIM é bastante simples, pois compara a variável relativa ao tempo de simulação com os diversos instantes a realizar os eventos. Em determinado instante, caso esteja agendada uma mudança na rede, procede-se à alteração do valor relativo ao ganho e ao ruído nas ligações abrangidas.

Deste modo foi implementada uma aplicação que permite o suporte para mobilidade em redes de sensores sem fios no simulador TOSSIM, na qual está patente a versatilidade permitida graças à possibilidade de os dados serem introduzidos num ficheiro de texto.

Além disso, ao prever um modo de funcionamento onde os nós se movem em instantes aleatórios, garante uma melhor exploração do espaço de estados durante as simulações.

O fluxograma apresentado na Figura 4.1 ilustra o conjunto de procedimentos necessários para implementar o suporte para mobilidade dos nós no simulador TOSSIM. Note-se o facto de os identificadores A e B corresponderem aos nós membros da ligação a alterar o ganho, enquanto G diz respeito ao novo ganho dessa ligação.

Inicialmente efectua-se a leitura da informação relativa às alterações decorrentes na rede ao longo do tempo e presentes no ficheiro de texto. De seguida inicia-se a simulação, durante a qual, a cada instante, verifica-se se existe alguma alteração a efectuar na rede. Caso exista, procede-se às respectivas mudanças na estrutura da rede.

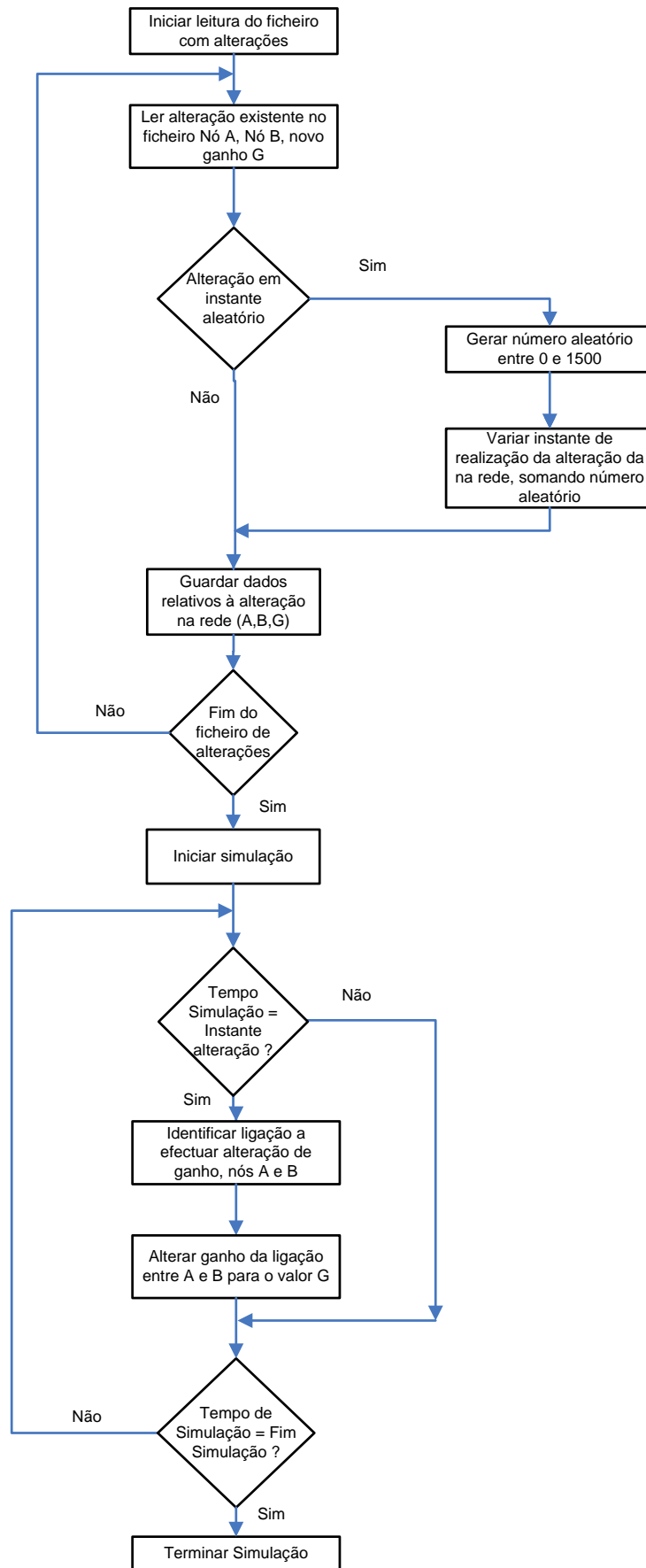


Figura 4. 1 - – Procedimentos relativos ao suporte para mobilidade dos nós no simulador TOSSIM

## 4.2. Pontos de Análise

Analisado o ambiente de testes do protocolo implementado e os componentes utilizados para proceder a esse processo, mencionam de seguida os pontos analisados, assim como as razões que estiveram na base da sua escolha. De entre os inúmeros testes efectuados, três aspectos revelaram-se cruciais para a análise do desempenho do protocolo: consumo de energia, tempo de transição assíncrono-síncrono e o débito.

Tendo em conta que o protocolo MAC foi desenvolvido para redes de sensores sem fios, um factor indispensável é o consumo energético, devido à reduzida durabilidade das baterias dos sensores. Assim, a minimização do consumo energético consiste num dos factores essenciais a analisar nesta dissertação. Nos protocolos híbridos desenvolvidos para suportar mobilidade em redes de sensores sem fios e analisados no Capítulo 2, verifica-se a existência de um intervalo de tempo elevado na transição entre o modo assíncrono e síncrono dos nós.

Estando o MMH-MAC perante esta transição entre os dois estados, a análise do tempo de transição assíncrono-síncrono apresenta-se como um factor essencial para a percepção do desempenho deste protocolo comparativamente com os existentes. Note-se a importância da minimização deste tempo de transição aquando da agregação de um nó móvel assíncrono a um *cluster* estático síncrono, a qual foi analisada teoricamente no Capítulo 3.

A capacidade de transmissão numa rede de sensores sem fios consistiu outro dos pontos a analisar na presente dissertação, de modo a fornecer informações acerca do débito disponível em situações de carga elevada na rede. O débito, também designado por *throughput*, foi obtido medindo a quantidade de dados que um nó consegue receber, em média, durante um determinado intervalo de tempo. Nas três secções seguintes são apresentados os três factores analisados. Além disso, em cada secção são abordados os testes realizados de modo a obter os resultados apresentados na secção seguinte.

### 4.2.1. Consumo de Energia

Devido à reduzida longevidade das fontes energéticas existentes nos nós (sensores) constituintes de uma rede de sensores sem fios, o consumo energético apresenta-se como

um dos aspectos que assumem maior relevo numa rede de sensores sem fios. No caso dos sensores TelosB a energia fornecida limita-se a duas pilhas AA.

No entanto, uma rigorosa estimação do tempo de vida de uma aplicação consoante o seu consumo energético apenas poderá ser efectuada considerando no modelo energético os consumos da totalidade dos elementos que constituem o sensor. Tendo em conta a escassez de informação existente acerca dos consumos energéticos dos elementos constituintes de cada sensor, numa outra tese de mestrado paralela [Silva 2008] foram efectuadas medidas dos consumos reais num dispositivo TelosB, apresentadas na secção 4.3.2. Este estudo caracterizou-se pela elevada exaustividade, tendo sido recolhidos os consumos do rádio a transmitir/receber, do MCU, sensores a ler e *flash* escrever/ler, em vários modos de funcionamento.

#### **4.2.2. Tempo de transição Assíncrono-Síncrono**

No cenário abordado nesta dissertação, analisado no Capítulo 3, considera-se a existência de *clusters* de nós estáticos e nós móveis que se deslocam entre esses *clusters*. Assim, estes nós móveis ao entrarem na vizinhança de um *cluster* estático pretendem num curto intervalo de tempo recolher um elevado volume de informação e maximizar o débito.

De modo a maximizar o débito os nós estáticos e fixos devem funcionar no modo síncrono, necessitando-se que a transição do móvel entre estes dois estados ocorra num intervalo de tempo mínimo. De modo a testar o desempenho do protocolo, relativamente ao tempo necessário para efectuar esta transição, considerou-se a existência de 6 *clusters* distintos contabilizando ao todo 21 nós. Na Figura 4.2 ilustra-se o percurso efectuado pelo nó móvel M ao longo dos 6 *clusters* estáticos.

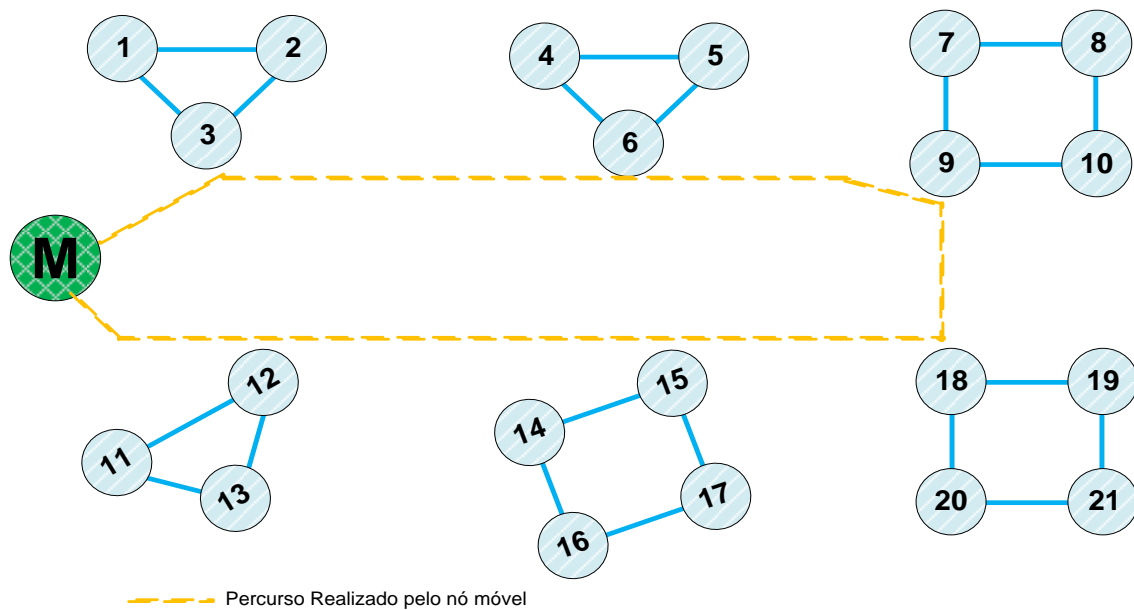


Figura 4. 2 - Percurso realizado pelo nó móvel M ao longo dos seis clusters

No total, o nó móvel efectua 120 sincronizações com os diversos *clusters* síncronos, entrando na vizinhança de cada um deles em intervalos de tempo aleatórios. A ocorrência de eventos em intervalos de tempo aleatórios com uma distribuição uniforme de largura 1100ms, garante que os resultados reflectem a possibilidade de o nó móvel chegar à vizinhança dos *clusters* assíncronos em qualquer instante na sequência de 11 *slots* com duração de 100ms cada.

### 4.2.3. Débito

Num protocolo de acesso ao meio para redes de sensores sem fios estáticas um dos factores a optimizar consiste no débito. Na presença de mobilidade na rede, os nós móveis vão ter ligações aos restantes nós com duração finita, onde efectuam a recolha de um elevado volume de dados. Ao maximizar o *throughput* garante-se a transmissão de um maior volume de dados no tempo de ligação disponível. O valor do débito refere-se à quantidade de mensagens/amostras recebidas pelo nó patrulha por período de tempo.

No protocolo MMH-MAC a medição do *throughput* foi efectuada considerando o intervalo de tempo desde que o nó móvel entra em contacto rádio com pelo menos um elemento do *cluster* estático, até sair da vizinhança do mesmo, permanecendo os nós do *cluster* estático continuamente a enviar mensagens. Assim, o resultado fornece informação acerca do *throughput* verificado consoante o tempo de permanência do nó móvel na

vizinhança, ou seja, de acordo com a velocidade de deslocamento desse nó móvel. Este valor contabiliza o tempo de sincronização, e o tempo de transmissão ao débito configurado, que depende do número de *slots* alocados.

Realizaram-se diversas simulações, nas quais se variou o tempo de permanência do nó móvel na vizinhança do *cluster* estático de modo a serem obtidos os resultados relativos à quantidade de dados recolhidos consoante a velocidade de deslocação do nó.

### 4.3. Resultados

Apresentados os pontos a abordar na análise de desempenho do protocolo, apresentam-se nesta secção os resultados obtidos. Esta secção encontra-se dividida em três subsecções. A primeira faz referência ao modelo de recolha de dados utilizado para análises estatísticas. Na segunda apresenta-se o modelo energético considerado nos dispositivos TelosB, de modo a estimar o tempo de vida das baterias dos sensores. Na última, são apresentados os resultados obtidos, relativos aos vários pontos em análise.

#### 4.3.1. Recolha de dados das simulações

De modo a proceder à medição do desempenho do protocolo MMH-MAC, foi usado o método das réplicas [Taha 1982]. Este método consiste em reunir um conjunto de observações independentes e de seguida realizar um tratamento estatístico sobre as mesmas. Com base nessas amostras ( $x_i$ ) obtém-se uma estimativa para o valor médio e variância de uma dada variável aleatória ( $X$ ). Estes valores são fornecidos respectivamente pelas equações (4.1) e (4.2).

Outro aspecto a salientar deve-se ao facto de a distribuição de cada variável ( $X$ ), não ser determinada. No entanto, sendo as  $N$  amostras independentes, pelo teorema do limite central, a distribuição de  $X$  vai tender para uma distribuição normal à medida que  $N$  aumenta.

$$\bar{X} = \frac{1}{N} \sum_{i=1}^N x_i \quad (4.1)$$

$$S^2 = \frac{1}{N-1} \sum_{i=1}^N (\bar{X} - x_i)^2 \quad (4.2)$$

Efectuada a recolha das estimativas dos vários parâmetros torna-se necessária determinar o seu intervalo de confiança, isto é, o seu erro de estimativa.

Caso a variável aleatória possua uma distribuição normal, o valor esperado dessa variável está compreendido no intervalo definido pela equação (4.4) [Mitrani 1982] com uma probabilidade  $1 - \varphi$  e um número  $n$  de amostras, onde o parâmetro é calculado com a equação (4.3).

$$P(t_n - 1 \leq z) = 1 - \frac{\varphi}{2} \quad (4.3)$$

$$P\left\{\bar{X} - \left(\frac{zS}{\sqrt{n}}\right) < \mu \leq \bar{X} + \left(\frac{zS}{\sqrt{n}}\right)\right\} = 1 - \varphi \quad (4.4)$$

Na análise de resultados apresentada nesta dissertação, procedeu-se ao cálculo dos intervalos de confiança para todas as simulações efectuadas. Para a totalidade das simulações foi utilizado um intervalo de confiança de 95% (*IC 95%*),  $\varphi = 5\%$ .

#### 4.3.2. Modelo de consumo energético nos Dispositivos TelosB utilizado

Nesta dissertação é utilizado um modelo de consumo energético obtido em [Silva 2008], onde foram estudados através de medições reais os consumos do dispositivo de rádio, do MCU, da escrita/leitura da *flash* e dos quatro sensores disponíveis.

O MCU caracteriza-se pela possibilidade de suportar vários modos de poupança de energia, *Low Power Modes* (LPM's), os quais possuem consumos de corrente distintos entre si. Esses modos que perfazem um total de seis, são apresentados na tabela 4.1.

Outro aspecto que assume uma crucial relevância acerca do consumo energético dos dispositivos TelosB, prende-se com o elevado consumo dos relógios, além do verificado para o CPU. Sucintamente, considerem-se as principais características dos relógios existentes:

- *Basic Clock (BC)*: caracterizado pelo seu baixo consumo, inclui o suporte ao relógio de cristal externo de 32768 Hz e um oscilador controlado internamente *Digitally Controlled Oscillator* (DCO). Contém o *Auxiliary Clock* (ACLK).
- *Auxiliary Clock* (ACLK): fornecido pelo relógio de cristal externo de 32768 Hz ou pelo DCO quando o MCU se encontra no modo activo.
- *Main Clock* (MCLK): usado pelo CPU.

- *Sub-Main Clock (SMCLK)*: usado pelos dispositivos periféricos.

O DCO consegue efectuar a transição do modo adormecido para modo ligado em cerca de 6  $\mu$ s.

| Modo de funcionamento | Estado do sistema   |
|-----------------------|---|
| Activo                | CPU e a totalidade dos relógios activos   |
| LPM0                  | CPU e MLCK desligados encontrando-se ACLK e SMCLK activos   |
| LPM1                  | CPU e MCLK desligado, ACLK e SMCLK activos, gerador dc DCO desligado apenas se o DCO não for usado no modo activo |
| LPM2                  | CPU, MCLK e SMCLK desligados, ACLK activo, gerador dc DCO permanece activo  |
| LPM3                  | CPU, MCLK, gerador dc DCO e SMCLK desligados, ACLK activo   |
| LPM4                  | CPU e todos os relógios desligados  |

**Tabela 4. 1 - Características dos modos LPM existentes nos sensores TelosB**

De seguida apresentam-se os valores usados para estimar o consumo energético dos diversos nós, efectuando-se uma distinção entre os componentes.

- **Consumo de corrente nos vários modos LPL do MCU**

Para os vários modos de funcionamento do MCU, os valores usados para a corrente consumida foram:

| Corrente Consumida MCU LPM |                         |
|----------------------------|-------------------------|
| Modo MCU                   | Corrente consumida (mA) |
| Activo                     | 1.8                     |
| LPM0                       | 0.178                   |
| LPM1                       | 0.179                   |
| LPM2                       | 0.036                   |
| LPM3                       | 0.014                   |
| LPM4                       | 0.009                   |

**Tabela 4. 2 - Consumo de corrente do TelosB nos vários modos de LPM, encontrando-se o dispositivo de rádio desligado**

Como se pode constatar pelos valores acima, o modo LPM3 é cerca de 129 vezes mais eficiente do que o modo activo. Logo, consiste no modo ideal para poupar energia quando o rádio se encontra desligado e quando não existe nenhuma tarefa pendente à excepção dos relógios. Nessa dissertação paralela, onde foram efectuadas as medições

acerca do consumo dos nós, verificou-se a transição do MCU para o modo LPM3 caso não esteja a efectuar nenhum processamento ou não possua tarefas pendentes.

- **Consumo de corrente do dispositivo de rádio**

O dispositivo de rádio poderá funcionar em um de três estados: Rádio Ligado a escutar canal (RX), Enviar mensagens (TX pacote) ou Receber mensagens (RX pacote). Note-se o facto de o dispositivo de rádio ser o maior consumidor de energia, como se pode verificar pelos valores apresentados na tabela seguinte.

| Corrente Consumida MCU + Rádio |                         |           |      |
|--------------------------------|-------------------------|-----------|------|
| Modo MCU                       | Corrente consumida (mA) |           |      |
|                                | TX pacote               | RX pacote | RX   |
| LPM3                           | 20.8                    | 21.5      | 19.1 |
| Activo                         | 20.8                    | 21.5      | 20.9 |

Tabela 4. 3 – Consumos médios do MCU, juntamente com o rádio, nos vários modos LPM

- **Consumo de corrente do dispositivo *flash***

Na tabela seguinte encontram-se os valores usados para a corrente consumida pela *flash* aquando das operações de leitura, escrita ou caso apague dados.

| Corrente Consumida MCU + Flash |                         |         |        |
|--------------------------------|-------------------------|---------|--------|
| Modo MCU                       | Corrente consumida (mA) |         |        |
|                                | Escrita                 | Leitura | Apagar |
| LPM3                           | 4                       | 4       | 5.5    |

Tabela 4. 4 - Resumo dos consumos médio do MCU mais o dispositivo flash

- **Consumo de corrente dos dispositivos sensores**

Os dispositivos TelosB são constituídos por quatro sensores, cujo consumo assume maior importância para longos períodos de monitorização do meio envolvente.

| Corrente Consumida MCU + Sensores |                         |          |      |      |
|-----------------------------------|-------------------------|----------|------|------|
| Modo MCU                          | Corrente consumida (mA) |          |      |      |
|                                   | Temperatura             | Humidade | TSAR | PAR  |
| LPM3                              | 0.55                    | 0.55     | 0.55 | 0.55 |

Tabela 4.5 - Consumos médios do MCU mais os sensores

### 4.3.3. Desempenho do Protocolo

#### 4.3.3.1. Tempo de Transição Assíncrono – Síncrono

Tal como foi mencionado nos capítulos anteriores, o tempo de transição do nó móvel do modo assíncrono para o síncrono consiste num dos aspectos de maior relevo numa rede de sensores sem fios móvel. De modo a avaliar o desempenho do protocolo MMH-MAC, recorreu-se há rede composta por seis *clusters* distintos apresentada na secção 4.2.2 e a um nó móvel que se desloca entre estes *clusters*.

Os nós estáticos encontram-se permanentemente no modo síncrono, usando o *slot* público (*slot* 0) e um número variável de *slots* dedicados. Em cada *slot* dedicado existe um tráfego periódico e constante de 10 pacotes de 35 bytes por *slot*. Os nós estáticos enviam um pacote SYNC por período, correspondendo à frequência máxima permitida no protocolo MMH-MAC. No caso dos nós móveis, estes permanecem temporariamente em contacto com os nós estáticos síncronos, mantendo essa conectividade durante um tempo aleatório descrito por uma distribuição uniforme com 2.2 segundos de largura, centrada no valor médio  $T_{CONNECT}$ . De modo a verificar o desempenho do protocolo foram testados dois tipos de nós móveis:

- Procuradores passivos, os quais esperam pelas mensagens SYNC enviadas pelos nós estáticos antes de proceder à sincronização com os mesmos;
- Procuradores activos que correm continuamente o protocolo de sincronização descrito na secção 3.3.1.

No segundo caso, considerou-se a possibilidade de os nós estáticos síncronos nunca acordarem durante os *slots* não alocados, ou a hipótese de acordarem nos últimos 10ms do slot 6 (*slot* de escuta), de modo a escutar um possível envio de preâmbulos por parte de um nó móvel.

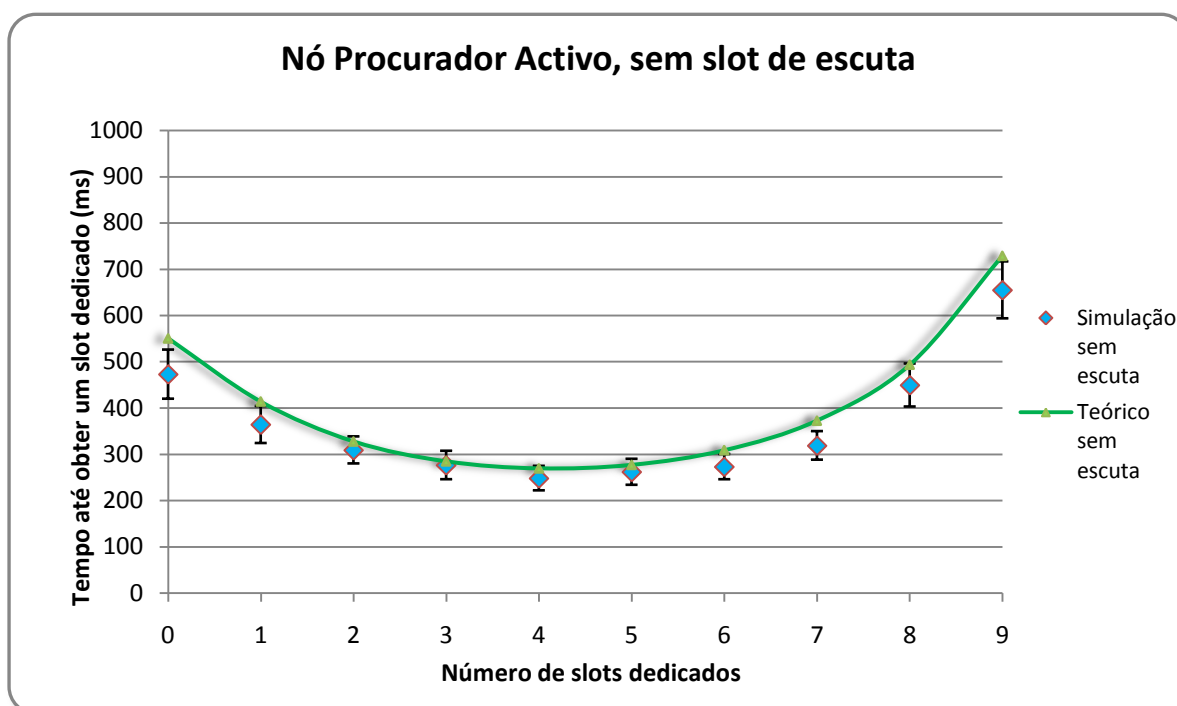
Durante o processo de sincronização, apenas um *slot* dedicado é alocado. Após a alocação, o nó móvel envia um pedido inicial, recebendo de seguida um fluxo contínuo de

pacotes de dados com 35 bytes, correspondendo à taxa máxima suportada num *slot* dedicado (cerca de 11.84 pacotes/*slot*). Durante os testes realizados procedeu-se à realização de 120 associações do nó móvel aos nós estáticos, medindo-se o tempo  $T_{sync}$ , definido como o tempo decorrido desde a entrada do nó móvel na vizinhança do estático até ao envio do pacote relativo à requisição da transmissão de dados.

Os resultados obtidos através das simulações efectuadas foram comparados com o modelo teórico apresentado no Capítulo 3, de modo a verificar a sua proximidade.

No que diz respeito aos resultados em que o nó móvel corre o algoritmo de sincronização continuamente (procurador activo), os resultados experimentais apresentados na Figura 4.3 mostram que o tempo mínimo até obter um *slot* dedicado ocorre aquando da existência de 4 slots alocados e o seu valor ronda os 280ms.

Além disso, no gráfico abaixo verifica-se que na quase totalidade dos pontos, o resultado teórico dado pela equação 3.9 encontra-se dentro do intervalo de confiança relativo aos resultados obtidos nas simulações efectuadas. Assim os resultados experimentais demonstram a validade do modelo teórico proposto no Capítulo 3.



**Figura 4.3 – Tempo de obtenção de um slot dedicado vs Número de slots alocados – sem slot de escuta**

Caso os nós síncronos procedam à escuta do canal nos últimos 10ms do *slot* 6, de modo a detectar possíveis transmissões de preâmbulos, o tempo de espera pelo *slot* dedicado diminui consideravelmente para poucos *slots* alocados, como mostra a Figura 4.4.

O tempo de espera é mínimo quando existem 3 *slots* alocados, apresentando uma magnitude de aproximadamente 200ms.

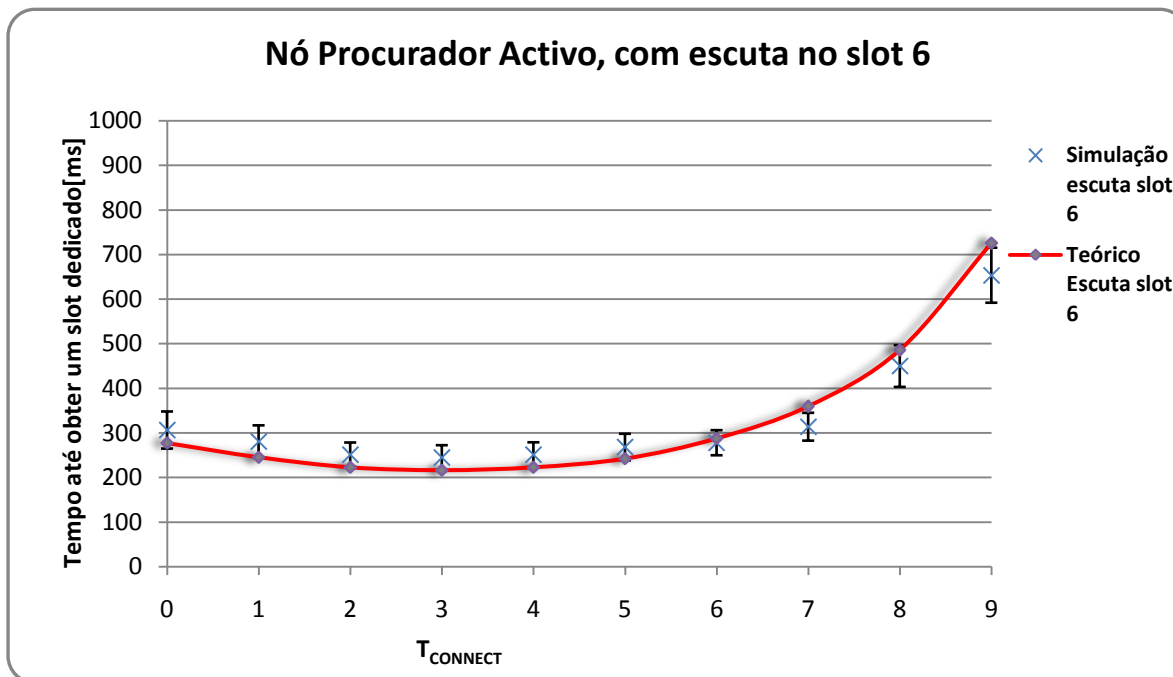


Figura 4. 4 – Tempo de obtenção de um slot dedicado vs Número de slots alocados – com slot de escuta

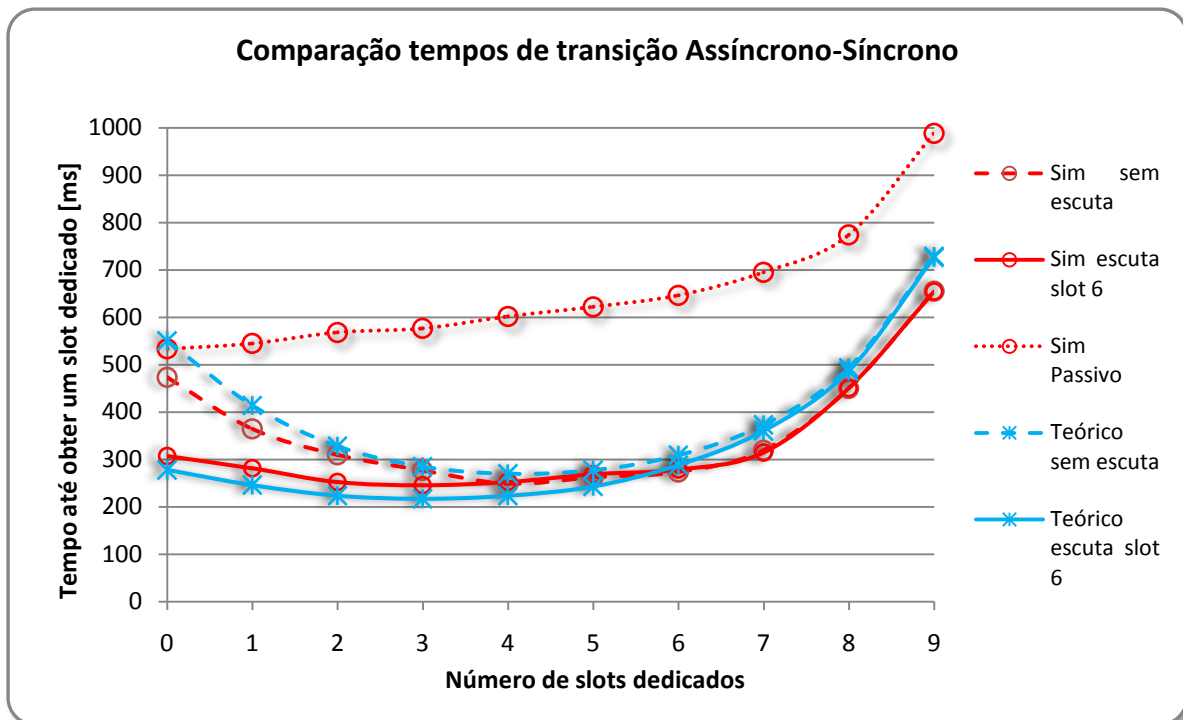
Constata-se também o facto de o resultado teórico dado pela equação 3.9 se encontrar dentro do intervalo de confiança do valor obtido experimentalmente.

Assim, apesar de a introdução da escuta no *slot* 6 aumentar o consumo energético em cerca de 1%, permite uma diminuição do tempo mínimo até à obtenção de um *slot* dedicado, em cerca de 29%.

Por último apresenta-se na Figura 4.5 uma comparação entre os diversos tempos teóricos e práticos, excluindo neste caso os intervalos de confiança anteriormente apresentados, com o objectivo de manter o gráfico legível. O resultado obtido através da abordagem clássica, isto é, onde o nó móvel espera a recepção da mensagem SYNC para proceder à sincronização apresenta valores consideravelmente mais elevados do que as restantes abordagens.

Um aspecto a salientar consiste na semelhança dos resultados para ambos os testes do nó procurador activo (com e sem escuta no slot 6), para um número de *slots* ocupados superior ou igual a 4. Logo a vantagem do recurso ao *slot* de escuta apenas se verifica no caso de o número de *slots* ocupados ser menor do que 4. Para valores superiores (carga elevadas na rede) a existência de um número considerável de *slots* alocados leva a que os nós recebam rapidamente os preâmbulos nesses slots dedicados, não existindo vantagem

em usar o *slot* de escuta. Observe-se também que neste cenário o custo energético da utilização de *slot* de escuta diminui, pois existe uma forte probabilidade de o *slot* de escuta estar alocado. Quando está alocado, o *slot* de escuta é igual a outro *slot* dedicado, tendo o tempo de escuta controlado pela actividade na rede.



**Figura 4. 5 – Comparação dos diversos tempos até à obtenção de um slot dedicado**

Comparativamente com o protocolo MH-MAC onde este tempo de transição apresentava pelo menos três períodos de 1.1s (mínimo 3.3s), verifica-se uma diminuição desse atraso em cerca de 94%, considerando o tempo mínimo relativo MMH-MAC, de aproximadamente 200ms. Esta considerável melhoria no desempenho do protocolo permite que um nó móvel assíncrono ao entrar na vizinhança de um *cluster* síncrono estabeleça sincronismo com um nó desse *cluster*, proceda à alocação de um *slot* e inicie a transmissão de dados, com atraso reduzido.

O custo relativo à utilização de um nó procurador activo é, sobretudo, a taxa de perda de pacotes induzida noutros nós devido à colisão de preâmbulos, já que o consumo de energia permanece praticamente inalterado. Os resultados da simulação mediram uma média de 0,12 pacotes perdidos num *slot* dedicado alocado devido à associação de um nó móvel e conseqüente envio contínuo de preâmbulos.

No entanto, o protocolo MMH-MAC encontra-se habilitado a recuperar essas perdas devido ao mecanismo de retransmissão dos pacotes perdidos. A taxa efectiva de perda de pacotes numa rede real provavelmente será maior, devido à interferência causada pelo envio contínuo dos preâmbulos quando o nó móvel está perto do limite do alcance rádio de comunicação. Embora, o simulador TOSSIM não suporte este modelo de interferência, os testes iniciais realizados nos sensores TelosB mostram que a aproximação é viável e que a taxa de pacotes perdidos num protótipo real é reduzida.

#### 4.3.3.2. Débito

Outro parâmetro cuja análise é indispensável para avaliar o desempenho do protocolo MMH-MAC é o *throughput*, ou seja, a quantidade de bytes de informação transmitida por segundo.

Realizaram-se 30 simulações onde se variou o tempo de permanência do nó móvel na vizinhança de um *cluster* estático e contabilizou-se o número de mensagens com 51 bytes, recebidas durante esse intervalo de tempo. No cenário considerado, o nó móvel efectua um percurso onde encontra dois *clusters* estáticos distintos. O percurso completo tem a duração de 10 minutos, mas o nó apenas permanece um determinado intervalo de tempo variável na vizinhança de cada *cluster*, como referido. O número de mensagens contabilizado diz respeito a cada um dos intervalos de permanência na vizinhança do *cluster* estático.

Na figura seguinte ilustra-se o percurso realizado pelo nó móvel ao longo dos 10 minutos.

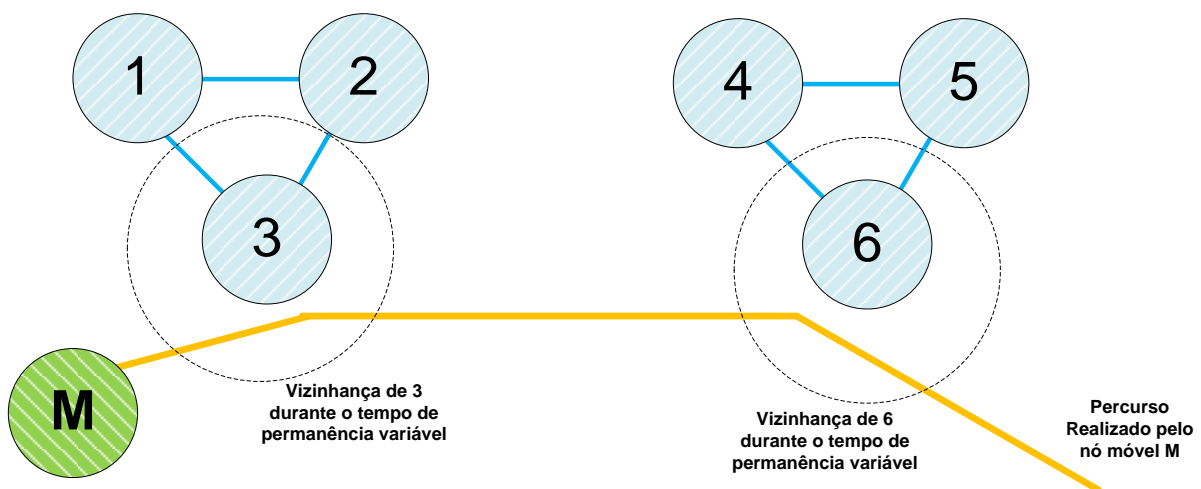


Figura 4. 6 – Percurso realizado pelo nó móvel no cenário considerado

Foram considerados três cenários distintos: ambos os nós assíncronos (assíncrono-assíncrono); o nó móvel inicialmente assíncrono e o fixo síncrono (assíncrono síncrono); e a existência de dois nós síncronos que permanecem estáticos a transmitir mensagens entre si (síncrono sem mobilidade), apenas como termo de comparação. Nos casos em que se verifica a existência de sincronismo entre os nós, a transmissão ocorreu recorrendo a três *slots* dedicados.

Das 30 simulações realizadas, foram excluídas a cinco melhores e as cinco piores. Na tabela seguinte apresentam-se os resultados obtidos, no que diz respeito ao número médio de mensagens recebidas, ao *throughput*, em função do tempo de permanência na vizinhança do cluster estático. Note-se a relação directa existente entre o tempo de permanência e a velocidade de deslocamento do nó, ou seja, à medida que esse intervalo de tempo aumenta, a velocidade do nó móvel diminui.

Por outro lado, procurou-se definir um modelo teórico relativo ao caso onde ocorre a transição assíncrono-síncrono, de modo a compararem-se os resultados experimentais e teóricos.

O tempo total é dado pela soma do intervalo de tempo onde se procede à transição entre o modo assíncrono e o síncrono ( $t_{setup}$ ) e do intervalo onde ocorre a transmissão de mensagens ( $T_{trans}$ ), isto é:

$$T_{total} = t_{setup} + T_{trans} \quad (4.5)$$

Além disso, sabe-se que o débito é dado por:

$$Débito = \frac{R \cdot T_{trans}}{T_{total}} \quad (4.6)$$

Onde R corresponde ao número de bytes transmitidos por período de 1.1s. Assim, em função do número médio de mensagens por slot  $N_{msg}$ , do número de bytes por mensagem  $Bytes_{msg}$  e do número de slots alocados  $N_{slots}$ , a expressão toma a seguinte forma:

$$R = N_{msg} \cdot Bytes_{msg} \cdot N_{slots} \quad (4.7)$$

Sabe-se ainda que o tempo de estabelecimento da ligação,  $t_{setup}$  foi anteriormente analisado no Capítulo 3, sendo obtido através da expressão 3.9.

Finalmente, a expressão relativa ao débito no caso de ocorrência da transição assíncrono-síncrono, toma a seguinte forma:

$$Débito = R \left( 1 - \frac{t_{setup}}{T_{total}} \right) \quad (4.8)$$

No gráfico da Figura 4.7 efectua-se a comparação entre os vários resultados obtidos. O débito relativo à existência de transmissão entre nós estáticos síncronos representa o valor máximo atingível, para um valor de  $t_{setup}$  nulo.

No caso do nó móvel assíncrono que comunica com um nó fixo síncrono, apenas apresenta um decréscimo no resultado obtido comparativamente com o anterior, nos primeiros 10s, aproximadamente. Este aspecto deve-se ao tempo de estabelecimento da ligação, o qual vê o seu efeito ser atenuado para tempos de permanência superiores. Isto é, para tempos de permanência mais elevados, o resultado obtido aproxima-se do referente ao caso em que os nós se mantêm síncronos a transmitir mensagens entre si. À medida que esse tempo aumenta, o efeito negativo do tempo de transição inicial atenua-se e o resultado tende para o relativo aos dois nós permanecerem estáticos.

Por outro lado, no caso do resultado teórico para a situação onde ocorre a transição assíncrono-síncrono, verifica-se que este sofre um ligeiro decréscimo para tempos de permanência reduzidos, passando depois a estabilizar no seu valor máximo, aproximadamente a partir dos 10s. Este aspecto é semelhante ao verificado no caso dos resultados experimentais.

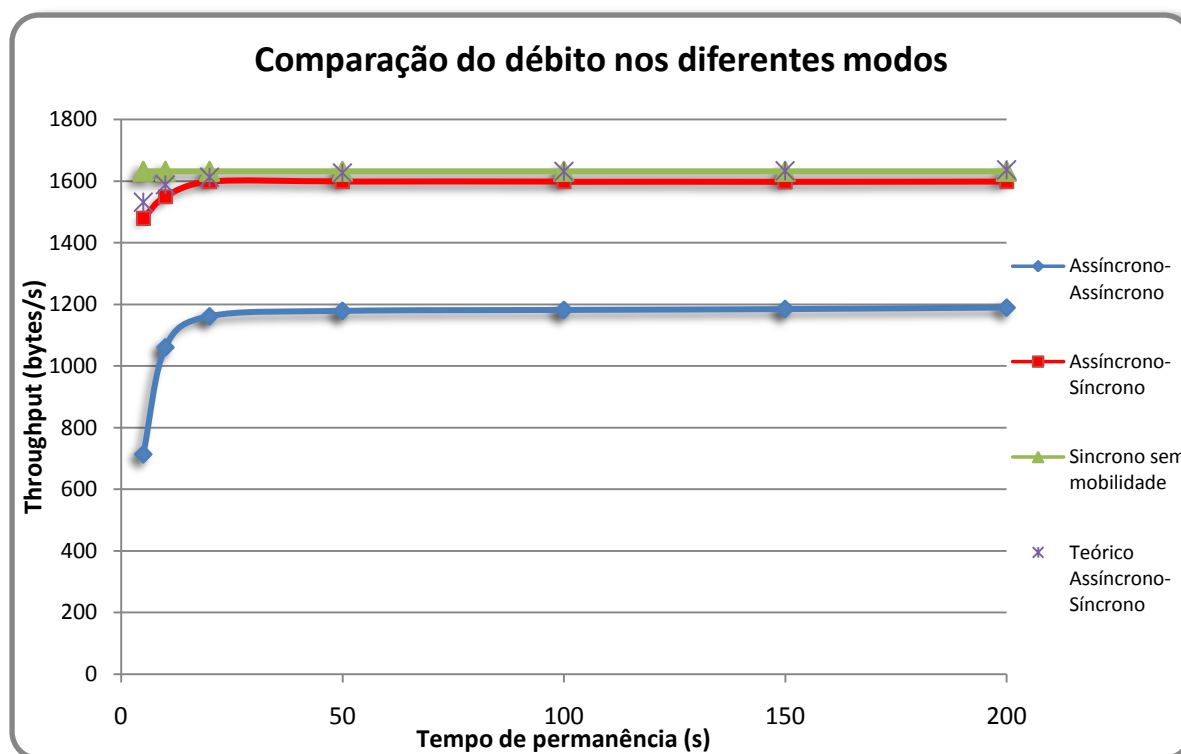


Figura 4. 7 – Débito vs Tempo de permanência para Assíncrono-Assíncrono e Assíncrono-Síncrono.

Relativamente ao caso em que ambos os nós permanecem assíncronos, obtiveram-se os piores resultados. Neste caso ocorre o envio de preâmbulos durante 2.2s, antecedendo o envio dos dados. Este aspecto provoca um consequente atraso inicial, o qual interfere de forma negativa no resultado referente ao débito. No entanto, o débito aumenta ligeiramente para valores de tempo de permanência acima dos 20 segundos, já que a partir desse valor diminui o efeito negativo provocado pelo atraso inicial devido ao envio dos preâmbulos que antecedem os dados.

Assim, conclui-se que o método de transição assíncrono-síncrono permite obter valores de débito superiores em mais de 33% ao resultados em que ambos os nós permanecem assíncronos, praticamente igualando o débito máximo possível para tempos de permanência superiores a 10 segundos. Numa transmissão em modo assíncrono, cada pacote de dados é enviado precedido de um preambulo. Caso o receptor nunca adormeça, envia rapidamente o PACK, que interrompe os preâmbulos, originando um débito relativamente elevado para um emissor, pois não há problemas de contenção no acesso ao meio.

Como a eficiência do modo assíncrono é penalizada pela concorrência no meio, procedeu-se ao estudo dos resultados obtidos em termos de débito para os dois cenários

considerados na Figura 4.7 mas na presença de vários emissores. Para cada emissor síncrono são alocados 3 *slots* dedicados. A limitação existente devido ao número máximo de *slots*, levou a que a análise fosse efectuada até a um máximo de 3 nós emissores, embora pudessem existir até um máximo de 10 emissores, desde que se reserva-se apenas um *slot* dedicado para cada emissor. Os resultados obtidos encontram-se no gráfico da figura seguinte.

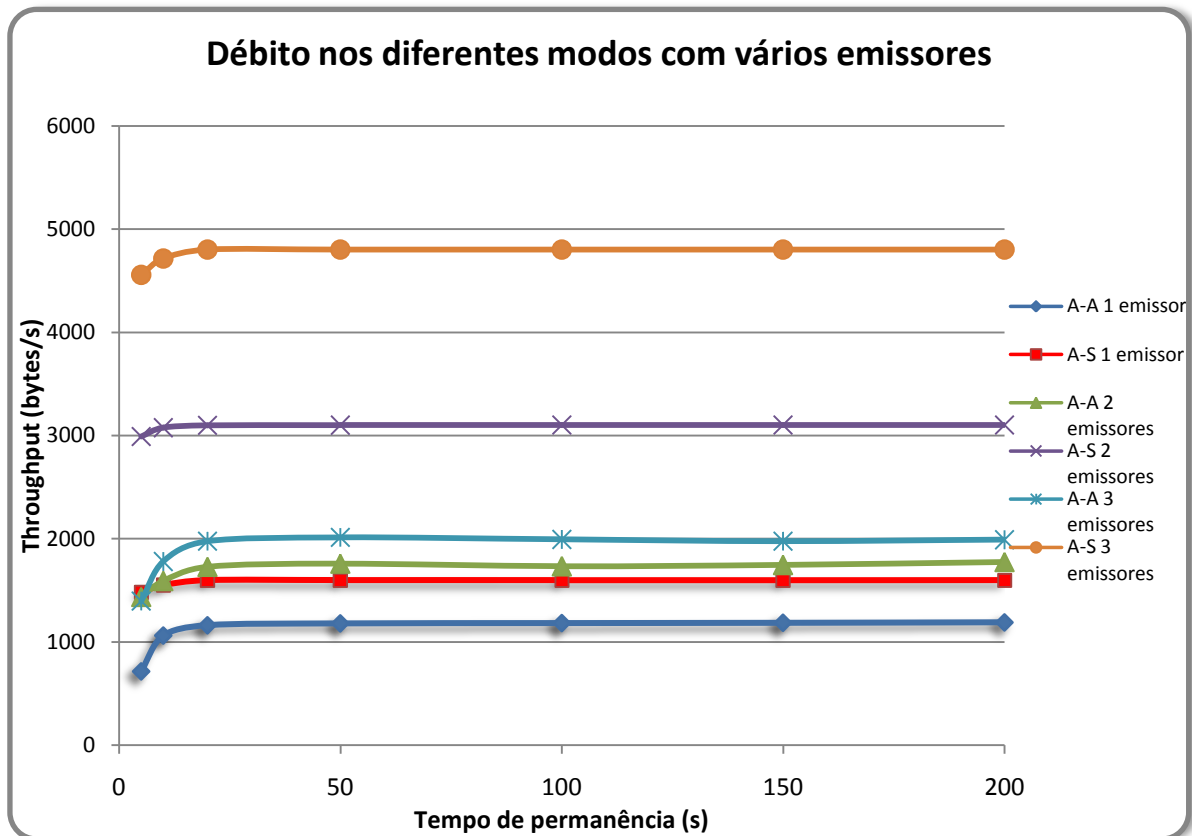


Figura 4. 8 - Débito Obtido na presença de vários emissores. A-A: Assíncrono-Assíncrono e A-S: Assíncrono-Síncrono.

Os resultados obtidos mostram que à medida que o número de emissores aumenta o incremento do débito verificado no caso assíncrono-assíncrono é bastante reduzido comparativamente com o verificado no caso assíncrono-síncrono, pois o efeito da contenção no acesso ao meio contribui para a redução do débito por utilizador. O reduzido aumento do débito ocorrido aquando do aumento de 2 para 3 emissores no caso assíncrono-assíncrono mostra que com este número de emissores, foi praticamente atingida a saturação em termos do débito quando tanto os nós emissores como o receptor se encontram no modo assíncrono.

Pelo contrário, o aumento do débito quando ocorre a transição do nó do modo assíncrono para o modo síncrono (assíncrono-síncrono), é directamente proporcional ao número de emissores. No caso de existirem três emissores o débito verificado em assíncrono-síncrono é cerca de 2,5 vezes o verificado no caso assíncrono-assíncrono.

#### 4.3.3.3. Consumo Energético

Numa rede de sensores sem fios o consumo energético consiste num dos factores de maior relevância, devido à curta durabilidade das baterias dos nós. No caso de uma rede onde coexistem nós móveis e estáticos, a principal preocupação com a redução do consumo energético é relativa aos nós estáticos, já que nos nós móveis, normalmente, é possível recarregar as suas baterias. Por exemplo, encontram-se localizados em veículos automóveis ou alojados na mala de bombeiros, os quais possuem pilhas suplentes para recarregar os sensores.

#### ❖ Comparação dos consumos energéticos em transmissões Assíncrono-Assíncrono e Síncrono-Assíncrono

Com o intuito de analisar o consumo energético num caso em que a transmissão ocorre entre nós assíncronos e um outro onde ocorre a existência de uma transição assíncrono-síncrono antecedendo a troca de dados, considerou-se o cenário anteriormente apresentado na secção anterior (4.3.3.2). Contabilizou-se a energia consumida pelos nós da rede ao longo dos 10 minutos de simulação, considerando-se 3 tipos de nós distintos:

- Nó móvel: a energia apresentada diz respeito ao consumo ao longo do percurso completo, ou seja, inclui as duas recolhas de dados;
- Nó fixo com dados: energia consumida por um nó fixo ao longo dos 10 minutos, o qual envia dados ao nó móvel durante o tempo de permanência deste na vizinhança;
- Nó fixo sem dados: considerou-se a energia consumida por um nó pertencente a um *cluster* da rede mas que não transmite dados ao nó móvel.

Note-se que no caso em que se estabelece sincronismo entre os nós, o nó fixo mantém os *slots* dedicados durante algum tempo após a saída do nó móvel da sua vizinhança.

Relativamente ao consumo de energia obtido para o caso em que os nós permanecem assíncronos, os resultados são apresentados no gráfico da Figura 4.9.

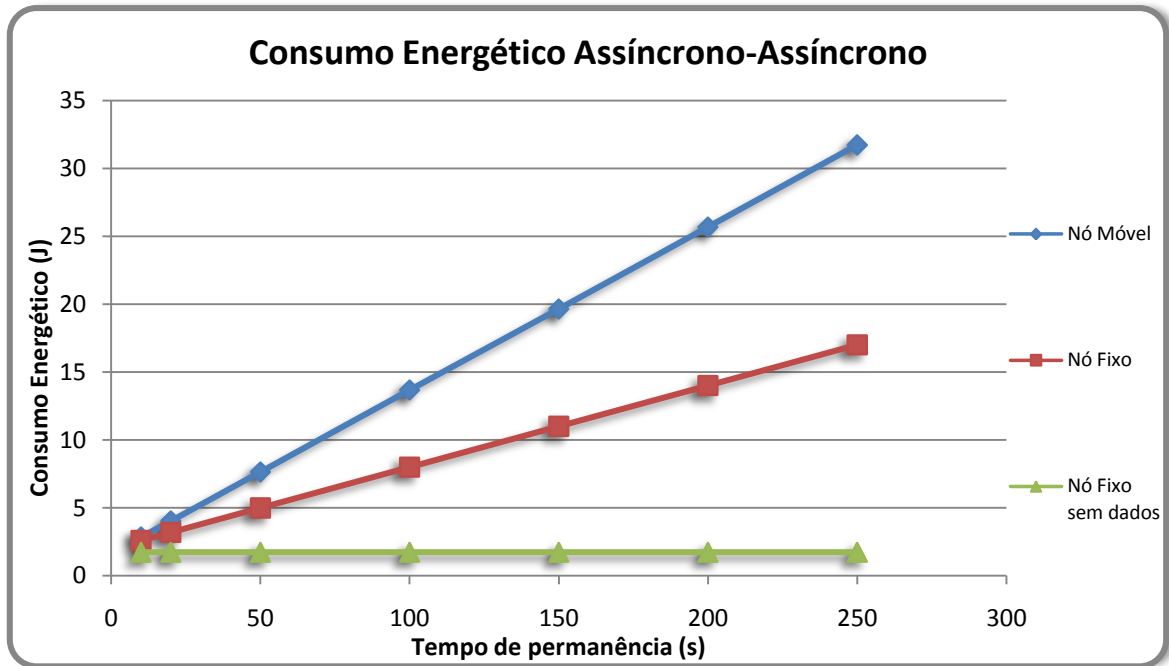


Figura 4.9 – Consumo energético no caso em que os nós permanecem assíncronos

No caso em que se estabelece sincronismo entre o nó móvel e o fixo, com o qual o nó procede ao envio de mensagens, os resultados obtidos encontram-se representados no gráfico seguinte.

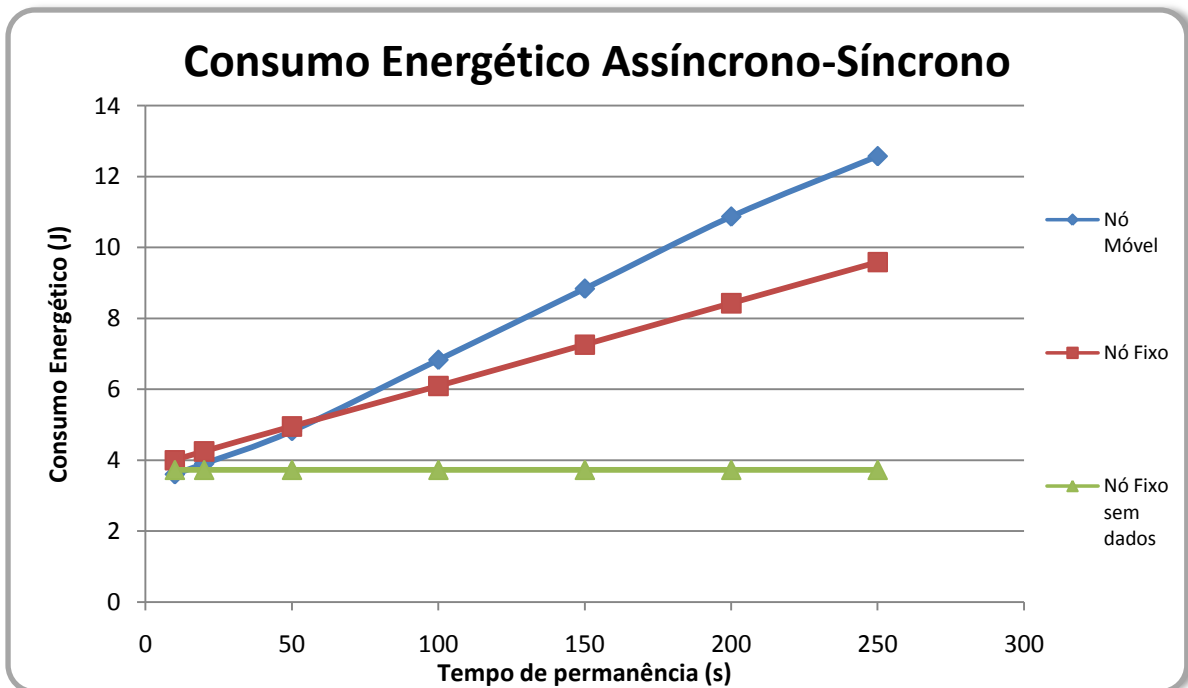


Figura 4.10 – Consumo energético, ocorrendo o estabelecimento de sincronismo entre o nó móvel e o fixo

Na situação assíncrono-síncrono o nó móvel apresenta um consumo energético inferior relativamente ao caso em que os nós permanecerem assíncronos, para tempos de permanência superiores a 50s. Abaixo desse valor o consumo mais elevado verifica-se no caso assíncrono-assíncrono.

Quanto ao nó fixo que transmite dados ao nó móvel durante o tempo de permanência, o consumo é inferior no caso assíncrono-síncrono, também para tempos de permanência superiores a 50s, onde é possível amortizar o custo energético da sincronização.

No que respeita ao nó fixo que não transmite dados ao nó móvel, o consumo energético revela-se inferior no caso assíncrono-assíncrono.

Assim, com base nos resultados acima, conclui-se que para tempos de permanência superiores a 50s torna-se vantajosa a situação assíncrono-síncrono, tanto que respeita ao nó móvel como ao fixo que transmite dados a esse nó móvel. No caso dos nós que se mantêm estáticos sem transmitir dados, o caso em que a totalidade dos nós se mantêm assíncronos assume maiores vantagens do ponto de vista energético.

#### ❖ **Consumo energético em função do número de *slots* alocado aquando da transição Assíncrono-Síncrono**

Neste caso, procedeu-se à análise da variação do consumo energético do nó móvel e nós estáticos síncronos da rede, em função do número de *slots* alocados aquando da sincronização inicial da rede. O algoritmo utilizado para realizar a transição assíncrono-síncrono, foi o descrito em 3.3.2.

Assim, considerou-se um cenário idêntico ao descrito na secção 4.3.3.1, com seis *clusters* distintos constituídos por nós síncronos, onde o nó móvel viaja entre eles contabilizando essa deslocação um total de 10 minutos. A energia foi contabilizada durante os 10 minutos que durou o percurso efectuado pelo nó móvel. Note-se o facto de o nó móvel efectuar 120 sincronizações com os nós dos clusters existentes e o tempo de permanência utilizado foi aproximadamente 1.5s.

Além da energia consumida pelo nó móvel, contabilizou-se também a energia consumida por cada um dos nós estáticos, os quais se mantêm síncronos durante o percurso

efectuado pelo nó móvel. Neste caso considera-se a frequência máxima para envio das mensagens SYNC, ou seja, uma por cada período 1.1s, nos primeiros 50ms do *slot* público.

O gráfico da Figura 4.11 efectua a comparação entre os consumos verificados nos nós estáticos e no nó móvel. No que respeita aos nós estáticos, efectuou-se uma média dos consumos verificados para esses nós, já que devido à sua semelhança, a representação individual apenas iria sobrecarregar o gráfico, não adicionando informação útil.

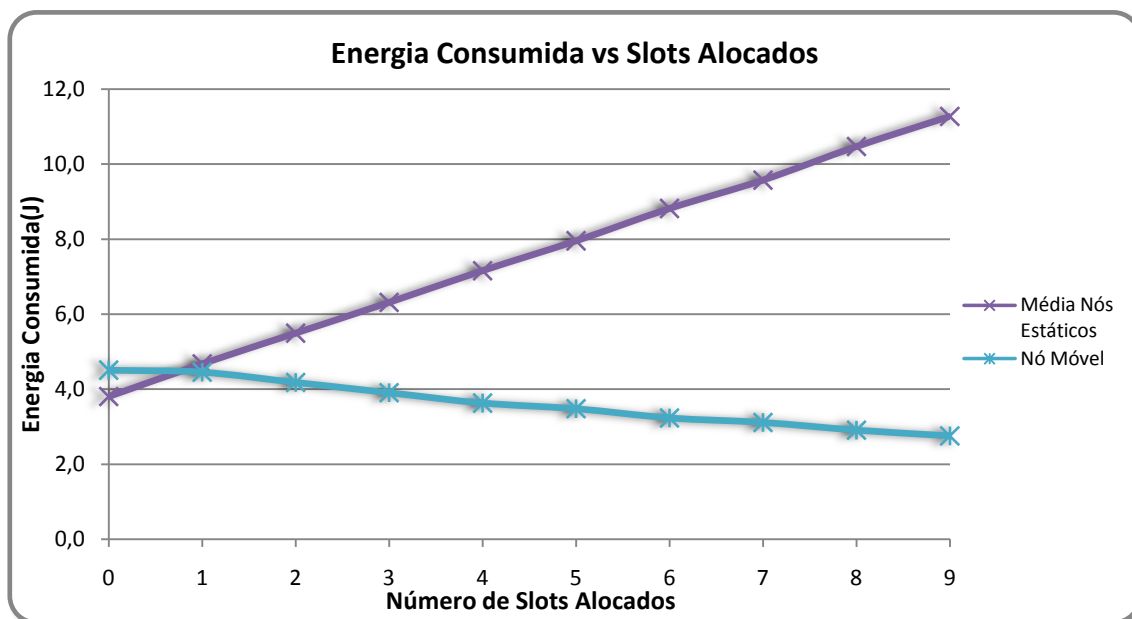


Figura 4. 11 – Energia consumida pelos nós estáticos e nós móveis, em função do número de *slots* alocados

Pela análise do gráfico acima, verifica-se o facto de a energia consumida pelo nó móvel diminuir consideravelmente à medida que o número de *slots* alocados durante o processo inicial de sincronização aumenta. Na situação dual encontra-se a energia consumida pelos nós estáticos. Ocorrendo um aumento do número de *slots* alocados, os nós passam a acordar em mais *slots* o que leva a um aumento do consumo energético verificado. O factor predominante para o nó móvel a curta duração do período de ligação aos nós estáticos, onde se contabiliza fortemente a fase de estabelecimento da ligação. Caso o número de *slots* alocados seja reduzido o nó móvel tem de enviar mais preâmbulos, até o receptor acordar e receber um deles. Pelo contrário, se o número de *slots* alocados for maior, o tempo de envio de preâmbulos diminui, conduzindo a uma diminuição do consumo energético do nó móvel.

Note-se o facto de esta representação ser um exemplo que visa dar uma noção do consumo energético com base no valor do tempo de permanência escolhido. Para tempos de permanência bastante superiores, a representação dos valores do nó móvel iria ter uma forma ascendente, tendendo a igualar a representação da média dos consumos dos nós fixos.

#### 4.4. Conclusões

Nas secções anteriores apresentaram-se os resultados que serviram de base à análise de desempenho do protocolo proposto nesta dissertação e permitiram estabelecer diversas conclusões sobre o funcionamento do protocolo.

No que respeita ao tempo de associação, os resultados obtidos permitiram a obtenção de tempos mínimos de aproximadamente 280ms na presença de uma carga de até 70% na rede. Este aspecto, comparativamente com o protocolo MH-MAC que esteve na base no desenvolvimento da presente dissertação representa uma diminuição considerável, já que o tempo anterior era de 3,3s.

Outro parâmetro que assume um enorme relevo nos resultados obtidos é o débito. No caso em que tanto os nós estáticos como os móveis se encontram assíncronos verifica-se a existência de um débito menor e de um maior consumo energético, já que antecedendo o envio de uma mensagem, é necessário verificar-se um método de *handshake* (Figura 3.2) em que o nó emissor envia preâmbulos, o receptor responde com um PACK e segue-se a transmissão dos dados. Inerente a este método está um considerável aumento do *overhead*.

Nesta situação (assíncrono-assíncrono), à medida que o número de nós emissores aumenta, o acréscimo no débito é reduzido comparativamente com o verificado no caso síncrono-síncrono, devido sobretudo ao aumento do número de colisões. Outro dos factores que condiciona este aumento do débito é o problema do nó escondido mas que devido à topologia utilizada, a sua influência não se revelou significativa na presente dissertação. Verificou-se ainda que a saturação do débito no caso assíncrono-assíncrono ocorre para cerca de 3 nós emissores, apresentando um valor que ronda os 2000 bytes/s. A vantagem desta abordagem é que os nós estáticos se mantêm com um consumo mínimo enquanto não existirem nós móveis na vizinhança.

Por outro lado, no caso em que o nó transita do modo assíncrono para o síncrono (assíncrono-síncrono) após a fase inicial de transição e alocação de *slots*, os nós acordam apenas em cada um dos slots alocados de modo a proceder ao envio dos dados no ritmo máximo. Isto, é os nós apenas acordam e procedem ao envio dos dados, não necessitando de qualquer troca anterior, o que diminui o consumo energético. À medida que o número de *slots* alocados aumenta este consumo também aumenta, mas por sua vez o débito aumenta proporcionalmente a esse número de *slots*. Daí o facto de o aumento do número de nós emissores aumentar proporcionalmente o débito, já que o número de *slots* alocados aumenta e não há qualquer disputa pelo meio. Para tirar proveito deste ganho de desempenho, e ser eficiente do ponto de vista energético a longo prazo, os nós estáticos devem transitar para o modo síncrono de forma controlada pela aplicação. Esta transição pode ocorrer após um primeiro acesso ainda em modo assíncrono, ou após detectarem uma condição de alarme no meio que estão a monitorizar.

# Capítulo 5.

## Conclusões

---

Neste último capítulo, realiza-se uma síntese geral, assim como a validação da hipótese formulada anteriormente face ao trabalho desenvolvido nesta dissertação. Resumem-se os principais contributos desta dissertação e são sugeridas algumas melhorias relegadas para trabalho futuro.

### 5.1. Síntese Geral

Na secção 1.3 desta dissertação foi apresentada uma hipótese caracterizada pela necessidade de desenvolver um protocolo de acesso ao meio destinado funcionar de uma forma otimizada em redes de sensores sem fios onde se verifica a mobilidade dos seus nós. Esta optimização está sobretudo relacionada com aspectos como a existência de um baixo consumo energético e um elevado débito.

Assim, esta dissertação tem como objectivo principal a inovação inerente aos protocolos de acesso ao meio desenvolvidos para redes de sensores sem fios móveis.

O desenho do protocolo idealizado foi apresentado no Capítulo 3. O objectivo de conjugar os conceitos base dos protocolo desenvolvidos para lidar com a mobilidade no domínio das MANET, com os destinados a minimizar a energia consumida numa rede de sensores sem fios, aumentou claramente a dimensão do desafio inerente à implementação do protocolo. Por outro lado, procurou-se aumentar a versatilidade comportamental do

protocolo, permitindo que determinados parâmetros do seu comportamento sejam definidos pela aplicação.

No Capítulo 4 procedeu-se à avaliação do desempenho do protocolo, de acordo com pontos de análise como o débito, o consumo energético e a capacidade de gestão da coexistência de nós assíncronos e síncronos numa rede. Nos diversos pontos, procurou-se complementar os resultados experimentais com modelos teóricos, permitindo assim a comparação entre os resultados esperados de um ponto de vista teórico e os obtidos através experimentalmente.

## 5.2. Conclusões

Nesta dissertação foi abordado um domínio emergente em termos de investigação científica nos tempos actuais, as redes de sensores sem fios. Foi sobretudo focado o desenvolvimento de protocolos de acesso ao meio destinados a redes de sensores sem fios móveis.

De entre os protocolos existentes para rede de sensores sem fios, existia, sobretudo uma preocupação com a poupança energética devido à curta longevidade das baterias, mas o comportamento dos mesmos não era otimizado para suportar mobilidade dos nós. Por outro lado nas MANET existem protocolos desenvolvidos de modo a suportar mobilidade dos nós, mas com elevados consumos energéticos.

A necessidade de existirem protocolos de acesso ao meio para redes de sensores sem fios, cujo funcionamento permita suportar a mobilidade dos nós e mantenham um reduzido custo energético, levou ao desenvolvimento da presente dissertação.

A principal contribuição desta dissertação consiste no desenvolvimento de um protocolo híbrido de acesso ao meio, o MMH-MAC, que minimiza o tempo de associação de nós móveis assíncronos a *clusters* síncronos. Por outro lado, verifica-se a existência de um *throughput* elevado apesar da existência de mobilidade dos nós.

Outra das contribuições desta dissertação consistiu na alteração efectuada ao simulador TOSSIM de modo a suportar mobilidade. A necessidade de proceder à simulação de diversos cenários de modo a verificar o desempenho do protocolo, obrigava à existência do suporte para essa funcionalidade no simulador TOSSIM. Como por defeito não existia, foi desenvolvida no âmbito desta dissertação. Esta caracteriza-se pela sua elevada

versatilidade, já que dá total liberdade ao utilizador para definir os instantes em que ocorre a mobilidade dos nós, assim como quais os nós considerados.

Uma das grandes inovações desta dissertação consistiu na optimização do intervalo de tempo de associação de um nó móvel assíncrono a um cluster síncrono, atingindo valores mínimos de aproximadamente 280ms. Este resultado revela uma considerável melhoria relativamente a protocolos como o MS-MAC [Pham 2004] e o MH-MAC [Bernardo 2007]. No caso do protocolo MH-MAC o tempo necessário para a conclusão dessa associação era de 3.3s, ou seja, conseguiu-se uma redução de 91,5%. Por outro lado, relativamente ao protocolo 802.15.4 [802.15.4 2003] também existe um tempo de associação mais curto, visto que nesse protocolo o nó móvel necessita de esperar por um *beacon* para iniciar o processo de associação a um *cluster* estático.

Relativamente ao débito, os resultados obtidos mostram a existência de claras melhorias impostas pelo algoritmo implementado para tratar os casos em que ocorre a transição do nó móvel para o modo síncrono (assíncrono-síncrono), comparativamente às situações em que os nós permanecem assíncronos (assíncrono-assíncrono). Nos casos em que apenas existe um emissor, verifica-se uma melhoria de 33% mas caso o número de emissores seja 3, essa melhoria é de 150%.

Outro aspecto a realçar nos resultados obtidos diz respeito ao facto de se verificar a redução do consumo energético dos nós estáticos no caso assíncrono-síncrono, comparativamente com a situação assíncrono-assíncrono. Por exemplo, para um tempo de permanência do nó móvel de aproximadamente 250s, verifica-se uma redução de 60,6% no valor do consumo energético desses nós. Como os nós fixos são a principal preocupação a nível energético, visto que geralmente não possuem uma forma simples de recarregar as baterias, este aspecto assume um particular relevo para a utilização do protocolo MMH-MAC em redes onde coexistem nós móveis e estáticos.

No desenvolvimento do protocolo recorreu-se à utilização de *cross-layering* entre as camadas MAC e de aplicação, de modo a permitir à aplicação controlar o estado de funcionamento e dependendo da escolha do utilizador, proceder à alocação de *slots* de acordo com os critérios definidos pela mesma. Este aspecto permite a existência de uma alocação de *slots* que corresponda aos requisitos de cada aplicação, aumentando a versatilidade do protocolo MAC.

O facto de a presente dissertação ter por base uma tecnologia recente à qual está subjacente uma enorme possibilidade de inovação, consistiu uma das principais fontes de motivação existentes durante a elaboração da dissertação. Fica, no entanto, a consciência de que apenas foram abordados alguns aspectos de uma grande área de investigação, ficando em aberto um conjunto alargado de assuntos, enumerados na secção seguinte.

### 5.3. Trabalho Futuro

Durante o desenvolvimento do trabalho, foram identificados alguns aspectos limitativos os quais permitem a sua exploração em trabalho futuro. De seguida procede-se à análise de aspectos acerca dos quais seria adequada uma abordagem futura.

Os resultados apresentados na presente dissertação, que serviram de base à análise do desempenho do protocolo foram obtidos através de simulações realizadas recorrendo ao simulador TOSSIM. É objectivo do autor a implementação do trabalho realizado nos sensores TelosB antes da data da apresentação e obtenção de resultados com base nessa implementação. De seguida pretende-se a realização da comparação entre os resultados provenientes das simulações com os obtidos a partir da implementação real nos sensores TelosB, e a sua inclusão numa publicação internacional a nível de revista ou conferência.

Por outro lado será útil a obtenção de resultados com base em cenários não considerados na presente dissertação, como por exemplo os nós móveis síncronos e os estáticos assíncronos. Neste caso a solução a considerar para proceder à sincronização entre os nós seria baseada no envio periódico de mensagens SYNC por parte dos nós móveis com uma frequência mais elevada, procedendo os nós assíncronos à transição para o modo síncrono com base nessa referência temporal.

Outra das questões a abordar futuramente será a transmissão sincronizada de vários nós no mesmo *slot* privado em vez de apenas se verificar um nó em cada um destes *slots*. Este aspecto poderá ser obtido considerando *micro-slots* com uma duração de por exemplo 15ms, transmitindo cada um dos nós no *micro-slot* correspondente. Atendendo à ineficiência do CC2420 presente nos TelosB e MicaZ, que apenas permite o envio de 7 mensagens (em média) por cada slot de 100ms (22,4 Kbps), seria possível aproximar o débito na recepção do limite teórico do 802.15.4, de 250Kbps.

## Bibliografia

---

- [802.11 1999] "802.11, IEEE Standard for Information technology - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications", IEEE Computer Society, 1999.
- [802.15.4 2003] The Institute of Electrical and Electronics Engineers; "802.15.4, IEEE Standard for Information technology— Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LRWPANs)", IEEE Standard for Information technology, Outubro 2003.
- [Água 2008] Miguel Pereira Hugo Água, Luis Bernardo, Rodolfo Oliveira, Rui Dinis, and Paulo Pinto; "A MAC protocol for mobile wireless sensor networks with bursty traffic"; 8º Seminário da Rede Temática de Comunicações Móveis (RTCM), Monte da Caparica, Lisboa, Junho 2008.
- [Ahn 2006] Emiliano Miluzzo Gahng-Seop Ahn, Andrew T. Campbell, Se Gi Hong, and Francesca Cuomo; "Funneling-MAC: A Localized, Sink Oriented MAC for Boosting Fidelity in Sensor Networks"; Proc. of Fourth ACM Conference on Embedded Networked Sensor Systems (SenSys 2006), Boulder, Colorado, USA, Novembro 2006.
- [Bernardo 2007] Rodolfo Oliveira Luis Bernardo, Miguel Pereira, Mário Macedo, Paulo Pinto; "A Wireless Sensor MAC Protocol For Bursty Data Traffic"; Personal Indoor and Mobile Radio Communications (PIMRC'07), Athens, Greece, Setembro 2007.
- [Bharghavan 1994] Alan Demers Vaduvur Bharghavan, Scott Shenker, Lixia Zhang; "MACAW: A Media Access Protocol for Wireless LAN's"; ACM SIGCOMM, ACM PRESS, pp. 212-225, London, U.K., Agosto-Setembro 1994.
- [Buettner 2006] Gary V. Yee Michael Buettner, Eric Anderson, Richard Han; "X-MAC: A Short Preamble MAC Protocol for Duty-Cycled Wireless Sensor Networks"; The 4th ACM Conference on Embedded Sensor Systems (SenSys), pp. 307-320, Boulder, Colorado, USA, Outubro - Novembro 2006.
- [Chakeres 2004] Elizabeth M. Belding-Royer Ian D. Chakeres; "AODV Routing Protocol Implementation Design"; Proceedings of the International Workshop on Wireless Ad Hoc Networking (WWAN), Tokyo, Japan, Março 2004.
- [Clausen 2001] P. Jacquet T. Clausen, A. Laouiti, P. Muhlethaler, a. Qayyum et L. Viennot; "Optimized Link State Routing Protocol"; IEEE INMIC, Pakistan, 2001.

- [Crossbow 2007a] Crossbow, "TelosB", 2007a. Disponível em <http://www.xbow.com/Products/productdetails.aspx?sid=252>.
- [Crossbow 2007b] Crossbow, "MICA2 868, 916 MHz", 2007b. Disponível em <http://www.xbow.com/Products/productdetails.aspx?sid=174>.
- [Crossbow 2007c] Crossbow, "MICAz 2.4GHz", 2007c. Disponível em <http://www.xbow.com/Products/productdetails.aspx?sid=164>.
- [El-Hoiydi 2003] J.-D. Decotignie A. El-Hoiydi, C. Enz, E. Le Roux; "WiseMAC, an Ultra Low Power MAC Protocol for the WiseNET Wireless Sensor Network"; ACM SenSys, Los Angeles, California, USA, Novembro 2003.
- [Gay 2003] Philip Levis David Gay, David Culler, Eric Brewer, "nesC 1.1 Language Reference Manuals", 2003. Disponível em <http://nescc.sourceforge.net/>.
- [Heinzelman 2000] Anantha Chandrakasan Wendi Rabiner Heinzelman, Hari Balakrishnan; "Energy-Efficient Communication Protocol for Wireless Microsensor Networks"; Proceedings of the 33rd Hawaii International Conference on System Sciences, Maui, Hawaii, Janeiro 2000.
- [Hill 2000] Robert Szewczyk Jason Hill, Alec Woo, Seth Hollar, David Culler, Kristofer Pister; "System architecture directions for networked sensors"; SIGPLAN Not; Vol. 35, pp. 93-104, 2000.
- [Levis 2003a] Nelson Lee Philip Levis, Matt Welsh, David Culler; "TOSSIM: Accurate and Scalable Simulation of Entire TinyOS Applications"; Proceedings of the 1st international conference on Embedded networked sensor systems (SenSys'03), ACM Press, pp. 126-137, Los Angeles, California, USA, 2003a.
- [Levis 2003b] Nelson Lee Philip Levis; "TOSSIM: A Simulator for TinyOS Networks Version 1.0s", 2003b.
- [Levis 2005a] David Gay Philip Levis, Vlado Handziski, Jan-Hinrich Hauer, Ben Greenstein, Martin Turon, Jonathan Huio, Kevin Klues, Cory Sharp, Robert Szewczyk, Joseph Polastre, Philip Buonadonna, Lama Nachman, Gilman Tolleo, David Cullero, Adam Wolisz; "T2: A Second Generation OS For Embedded Sensor Networks"; Technical Report TKN-05-007, Telecommunication Networks Group, Technische Universität Berlin, 2005a.
- [Levis 2005b] Sam Madden Philip Levis, Joseph Polastre, Robert Szewczyk, Kamin Whitehouse, Alec Woo, David Gay, Jason Hill, M. Welsh, Eric Brewer, David Culler; "TinyOS: An operating system for wireless sensor networks"; Ambient Intelligence. Springer-Verlag, 2005b.
- [Levis 2006] Philip Levis, "TinyOS Programmings", 2006. Disponível em <http://csl.stanford.edu/~pal/pubs/tinyos-programming.pdf>.

- [Levis 2007] Philip Levis, "TinyOS 2.0 Documentation", 2007. Disponível em <http://www.tinyos.net/tinyos-2.x/doc/>.
- [Marín 2008] J. Arias I. Marín, E. Arceredillo, A. Zuoaga, I. Losada, J. Mabe; "LL-MAC: A low latency MAC protocol for wireless self-organised networks"; *Microprocessors & Microsystems*, pp. 197-209, Junho 2008.
- [Mitrani 1982] Isi Mitrani; "Simulation techniques for discrete event systems"; Prentice Hall, ISBN: 0521238854, 1982.
- [Pereira 2007] Miguel Pereira; "Protocolo Híbrido de controlo de acesso ao meio para redes de sensores sem fios"; Dissertação de mestrado em Engenharia Electrotécnica e de Computadores, Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa, Lisboa, 2007.
- [Pham 2004] Sanjay Jha Huan Pham; "An Adaptive Mobility-Aware MAC Protocol for Sensor Networks (MS-MAC)"; *Proceedings of the IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS)*, pp. 214-226, Fort Lauderdale, Florida, USA, Outubro 2004.
- [Polastre 2004] Jason Hill Joseph Polastre, David Culler; "Versatile low power media access for wireless sensor networks"; *Proceedings of the 2nd international conference on Embedded networked sensor systems (SenSys'04)*, ACM Press, pp. 95-107, Baltimore, MD, USA, Novembro 2004.
- [Polastre 2005] Robert Szewczyk Joseph Polastre, David Culler; "Telos: Enabling Ultra-Low Power Wireless Research"; *Proceeding of International Symposium on Information Processing in Sensor Networks (IPSN'05)*, IEEE Press, pp. 364-369, Los Angeles, California, USA, Abril 2005.
- [Raja 2008] Xiao Su Anjali Raja; "A Mobility Adaptive Hybrid Protocol for Wireless Sensor Networks"; *Consumer Communications and Networking Conference (CCNC)*, pp. 692-696, Las Vegas, Nevada, USA, Janeiro 2008.
- [Ramanathan 1997] Ram Ramanathan; "A unified framework and algorithms for (T/F/C)DMA channel assignment in wireless networks"; *IEEE Infocom*, pp. 900-907, Kobe, Japan, 1997.
- [Rentel 2007] Thomas Kunz Carlos Rentel; "Clock Sampling Mutual Network Synchronization for Mobile Multihop Wireless Ad Hoc Networks"; *Military Communications Conference (MILCOM)*, Orlando, Florida, 29-31 Outubro 2007.
- [Rhee 2004] Ajit C. Warriar Injong Rhee, Lisong Xu; "Randomized Dining Philosophers to TDMA Scheduling in Wireless Sensor Networks"; Technical Report, Computer Science Department, North Carolina State University, Raleigh, NC, 2004.

- [Rhee 2005] Ajit Warrier Injong Rhee, Mahesh Aia, Jeongki Min; "ZMAC: a Hybrid MAC for Wireless Sensor Networks"; The ACM International Conference on Embedded Networked Sensor Systems (SenSys), pp. 90-101, San Diego, USA, November 2005.
- [Rhee 2006] Ajit Warrier Injong Rhee, Jeongki Min, Lisong Xu; "DRAND: Distributed Randomized TDMA Scheduling For Wireless Adhoc Networks"; The Seventh ACM International Symposium on Mobile Ad Hoc Networking and Computing, ACM Press, pp. 190-201, Florença, Italia, Maio 2006.
- [Shi 2007] Guido Stromberg Xiaolei Shi; "SyncWUF: An Ultra Low-Power MAC Protocol for Wireless Sensor Networks"; IEEE transactions on mobile computing, Vol. 6, pp. 115-125, Janeiro 2007.
- [Shnayder 2004] Mark Hempstead Victor Shnayder, Borrong Chen, Geoff Werner Allen, Matt Welsh; "Simulating the Power Consumption of Large Scale Sensor Network"; Proceedings of the 2nd international conference on Embedded networked sensor systems (SenSys'04), ACM Press, pp. 188-200, Baltimore, MD, USA, Novembro 2004.
- [Silva 2008] Miguel Silva; "Monitorização e recolha de dados em redes de sensores sem fios dispersas"; Dissertação de mestrado em Engenharia Electrotécnica e de Computadores, , Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa, Lisboa, 2008.
- [Srinivasan 2006] Philip Levis Kannan Srinivasan; "RSSI is Under Appreciated", 2006.
- [Taha 1982] Hamdy Taha; "Operations Research - an introduction"; Mcgraw-Hill Publishing Company, ISBN: 0073211141, 1982.
- [Toh 1996] Chai-Keong Toh; "A Novel Distributed Routing Protocol To Support Ad hoc Mobile Computing"; IEEE 15th Annual International Phoenix Conference on Computers and Communications, IEEE IPCCC, Phoenix, AZ, USA, Março 1996.
- [Tomlinson 1975] Raymond Tomlinson; "Selecting Sequence Numbers"; Proceeding SIGCOMM/SIGOPS Inter-process Commun. Workshop, ACM, pp. 11-23, 1975.
- [TOSSIM 2007] TinyOS, "TOSSIM, a TinyOS simulator", 2007. Disponível em <http://www.tinyos.net/dist-2.0.0/tinyos-2.0.1/tos/lib/tossim/>.
- [van Dam 2003] Koen Langendoen Tijs van Dam; "An Adaptive EnergyEfficient MAC Protocol for Wireless Sensor Networks"; The ACM International Conference on Embedded Networked Sensor Systems (SenSys), pp. 171-180, Novembro 2003.
- [van Hoesel 2004] P. Havinga L. van Hoesel; "A lightweight medium access protocol (LMAC) for wireless sensor networks"; Proc. Int'l Conf. on Networked Sensing Systems (INSS), 2004.

- [Warneke 2003] B.A. Warneke; "Ultra-Low Energy Architectures and Circuits for Cubic Millimeter Distributed Wireless Sensor Networks"; PhD Dissertation, UC Berkeley, 2003.
- [Ye 2004] John Heidemann Wei Ye, Deborah Estrin; "Medium Access Control With Coordinated Adaptive Sleeping for Wireless Sensor Networks"; IEEE/ACM Transactions on Networking, Vol. 2 N° 3, pp. 493-506, Junho 2004.
- [Ye 2006] Fabio Silva Wei Ye, John Heidemann; "Ultra-Low Duty Cycle MAC with Scheduled Channel Polling"; Proceedings of the ACM International Conference on Embedded Networked Sensor Systems (SenSys), ACM, pp. 321-334, Boulder, Colorado, USA, Outubro/Novembro 2006.



## Apêndice A.

# Mobilidade no simulador TOSSIM

---

A necessidade de simular um protocolo MAC para redes de sensores sem fios móveis implicava o suporte para mobilidade do simulador utilizado, o TOSSIM. No entanto, o facto de não existir tal funcionalidade conduziu à necessidade de a desenvolver.

Num ficheiro de texto definem-se os instantes em que ocorrem alterações na rede e quais os nós envolvidos nessas mesmas alterações. O ficheiro destinado ao controlo das operações relativas à simulação propriamente dita, em código *python*, procede à leitura dos dados presentes no ficheiro de texto e à alteração do estado da rede, nos instantes definidos.

No que respeita à estrutura do ficheiro de texto utilizado para descrever as diversas alterações da rede, encontra-se descrita na Figura A.1:

| INSTANTES (s) | INSTANTEM (ms) | NÓ A  | NÓ B  | ALTERAÇÃO NO GANHO |
|---------------|----------------|-------|-------|--------------------|
| 50            | 200            | 1     | 5     | -20                |
| ....          | ....           | ..... | ..... | .....              |
| 250           | 388            | 4     | 8     | +6                 |
| <b>RANDOM</b> | 1              | _____ |       |                    |

Figura A. 1 - Estrutura do Ficheiro de texto usado para definir a mobilidade na rede

Os campos “InstanteS” e “InstanteM” definem o momento em que ocorre a alteração na rede, sendo o primeiro referente aos segundos decorridos desde o início da simulação e o segundo, aos milissegundos dentro desse segundo. Este aspecto permite aumentar consideravelmente a precisão do instante definido para ocorrência da alteração na rede.

Os campos “Nó A” e “Nó B” são os nós cuja ligação entre os mesmos será alterada e “Alteração do Ganho” diz respeito à mudança ocorrida no valor do ganho existente nessa ligação. Caso já exista uma ligação estabelecida entre os dois nós, este valor corresponde a uma soma/subtração a esse valor inicial de acordo com o sinal. Pelo contrário, se ainda não existe qualquer ligação entre esses dois nós, admite-se que o valor do ganho dessa nova ligação passa a ser o apresentado no ficheiro de texto.

Outro dos aspectos a salientar prende-se com o facto de o instante em que ocorre a alteração do ganho poder variar aleatoriamente relativamente ao definido no ficheiro de texto. Isto é, se o valor presente no parâmetro “RANDOM” (Figura A.1) for 0 (zero), a alteração na rede ocorrerá exactamente no instante definido pelo utilizador. Caso o valor desse parâmetro seja 1, é obtido um valor aleatório entre 0 e 900ms que constituirá o *offset* a aplicar ao valor inicial presente no ficheiro de texto.

Após iniciar a simulação, procede-se às mudanças na topologia nos instantes configurados no ficheiro (valor exacto no ficheiro de texto ou valor com *offset*).