Universidade Nova de Lisboa

Faculdade de Ciências e Tecnologia

Departamento de Informática

Dissertação de Mestrado em Engenharia Informática

Distribuição Segura de Chaves Criptográficas em Redes de Sensores sem Fios de Grande Escala: Estudo e avaliação experimental em ambiente de simualção

Aluno nº 28044 – Ivo André Lopes Rodrigues

Orientador:
Prof. Doutor Henrique João Lopes Domingos

14 de Fevereiro de 2011

Nº do aluno: 28044

Nome: Ivo André Lopes Rodrigues

Título da dissertação:

Distribuição Segura de Chaves Criptográficas em Redes de Sensores sem Fios de Grande Escala: Estudo e avaliação experimental em ambiente de simulação

Palavras-Chave:

- Redes de Sensores sem Fios (RSSF)
- RSSF e ambientes de Grande Escala
- Segurança em RSSF
- Métodos e Protocolos para Gestão e Distribuição Segura de Chaves
- Ambientes de simulação para RSSF de Grande Escala

Keywords:

- Wireless Sensor Networks (WSN)
- Large Scale WSN
- WSN Security
- Key-Management and Distribution: Methods and Protocols
- Simulation environments for Large Scale WSN

Resumo

O objectivo da presente dissertação tem em vista conceber, implementar e testar um ambiente de simulação para redes de sensores sem fios de grande escala. Pretende-se que o ambiente possa ser usado para aferir e comparar, numa base experimental e de forma sistemática, as características de diferentes protocolos de distribuição de chaves par-a-par, baseados em esquemas de auto-organização aleatória da rede.

O ambiente de simulação é baseado num núcleo de eventos discretos e serviços de suporte de acordo com a pilha IEEE802.15.4 para sensores TinyOS. O ambiente de simulação inclui um modelo de comunicação por radiofrequência e mecanismos de detecção e controlo de colisões, conforme a especificação de referência de controlo de acesso ao meio (MAC) da norma IEEE802.15.4 ou Zigbee.

O ambiente de simulação utilizado constitui uma ferramenta genérica para análise comparativa de protocolos de distribuição e estabelecimento de chaves criptográficas. Para tal possibilita análise de efectividade desses protocolos em relação aos seguintes critérios: condições de cobertura e conectividade efectiva da rede; latência da disseminação de dados bem como estabilização do processo de estabelecimento de chaves de acordo com a topologia gerada; condições de fiabilidade propiciadas pela topologia formada; impacto energético dos protocolos; e avaliação da eficácia dos mesmos face à projecção de ataques internos que sejam desencadeados a partir de intrusões em diferentes nós.

No âmbito da dissertação foram estudados protocolos que constituem referências da investigação na área dos protocolos de distribuição e estabelecimento de chaves para redes de sensores sem fios. Os resultados e contribuições da dissertação permitem complementar a literatura e a análise prévia desses protocolos com base em observações experimentais. Estas observações permitem uma análise crítica da sua comparação ou validação face a condições aproximadas a condições reais de operação. Os resultados da dissertação permitem assim complementar condições teóricas avançadas anteriormente na literatura com resultados mais próximos de condições reais de desempenho desses protocolos, nomeadamente em redes de grande escala operando de forma não supervisionada. Esta avaliação complementar constitui uma contribuição relevante para uma compreensão mais completa, sistemática e abrangente do funcionamento desses protocolos, em condições que não tinham ainda sido avaliadas anteriormente.

Abstract

The purpose of this thesis aims to design, implement and test a simulation environment for large scale wireless sensor networks. It is intended that the environment can be used to measure and compare, on an experimental basis and systematically, the characteristics of different pairwise key distribution protocols, based on self-organizing random network schemes.

The simulation environment is based on a core of discrete events and support services according to IEEE802.15.4 stack for TinyOS sensors. The simulation environment includes a radio frequency communication model and mechanisms of collision detection and control, according to the reference specification of medium access control (MAC) of the IEEE802.15.4 or Zigbee standard.

The simulation environment used is a generic tool for comparative analysis of protocols for establishment and distribution of cryptographic keys. It provides analysis of effectiveness of these protocols on the following criteria: coverage and effective connectivity of the network, latency of data dissemination as well as stabilization of the key establishment process according to the generated topology; reliability offered by the formed topology; energy impact in the protocols; and evaluation of the effectiveness of them against the projection of internal attacks that are triggered from intrusions on different nodes.

As part of this thesis several protocols, that are references in the research area of key distribution and establishment for wireless sensor networks, were studied. The results and contributions of this thesis allow supplementary literature and analysis of these protocols based on experimental observations. These observations allow a critical analysis of the protocols comparison or validation against approximate real operating conditions. The results of this thesis thus allow a complementation of theoretical conditions advanced in the literature with results closer to real conditions of performance of these protocols, particularly in large scale networks operating in an unsupervised manner. This complementary evaluation is an important contribution to a fuller, systematic and comprehensive understanding of the operation of these protocols, in conditions that had not been previously evaluated.

Índice

Índice de Figuras

Índice de Tabelas

1. Introdução

1.1. Motivação e enquadramento da dissertação

A proposta de protocolos seguros para distribuição e estabelecimento de chaves criptográficas (genericamente segredos criptográficos) para Redes de Sensores sem Fios, é uma das dimensões mais relevantes da segurança destas redes e suas aplicações. Um serviço de distribuição, estabelecimento e gestão de chaves numa RSSF é condição necessária para a segurança efectiva de protocolos de comunicação segura sobre a pilha 802.11.15 [1] ou Zigbee [2].

A investigação recente produziu diversas propostas de protocolos e modelos de distribuição, estabelecimento e gestão de chaves [3]. Os esquemas mais interessantes que foram propostos partem do princípio de estabelecimento de chaves simétricas, par-a-par, de modo a evitar que, devido a ataques por intrusão ou captura de nós[1] com comprometimento de dados armazenados nos sensores, a segurança da rede fique imediatamente e globalmente comprometida. De entre estes protocolos emergiram, com particular relevância, os chamados protocolos de estabelecimento probabilístico de chaves, com refrescamento dinâmico das chaves par-a-par e partindo de soluções de chaves iniciais pré-instaladas, anteriormente a uma fase de auto-organização aleatória da rede [4], [5], [6], [7]. A partir de esquemas base para protocolos ditos probabilísticos com pré-distribuição de chaves iniciais, têm sido propostos protocolos mais ou menos vocacionados para auto-organização aleatória da rede, visando diferentes topologias que sejam mais adequadas para diferentes aplicações. Em geral, os esquemas base anteriormente citados têm em vista topologias de redes planas. Contudo, no âmbito da presente dissertação, interessam particularmente analisar propostas de protocolos probabilísticos e de auto-organização dinâmica e aleatória adaptados para o estabelecimento de topologias hierárquicas ou em *clusters* [8], [9], [10], [11]. Estas últimas soluções são particularmente interessantes para exploração de heterogeneidade de sensores (com maior ou menor capacidade de computação,

[1] Ao longo deste relatório pode vir a ser utilizado o termo "nó" para representar um sensor distribuído na rede. A razão para tal prende-se na facilidade de interpretação segundo o contexto onde o termo é inserido.

comunicação e energia disponível) bem como para privilegiar estratégias de processamento intermédio e agregação de dados pela rede. Estas estratégias minimizam o impacto associado a formas de disseminação de dados por inundação e sem agregação, que provocam maior consumo de energia, visto que o impacto energético da comunicação nas RSSF é bastante superior ao consumo energético associado ao processamento nos nós.

Os protocolos probabilísticos e suas variantes diferem entre si de algumas características fundamentais em relação a critérios de análise tais como: condições de cobertura da rede, escalabilidade das soluções, revogação e refrescamento de chaves, tolerância a falham, condições de mapeamento de identificadores à localização dos nós, tolerância face a condições de negação de serviço, tolerância a intrusões, independência de componentes ou entidades de confiança, tipo de criptografia alvo ou adequação a diferentes topologias resultantes dos processos de auto-organização da rede.

Por outro lado, para além das anteriores características, a eficácia de um protocolo de distribuição de chaves requer uma avaliação de desempenho em relação a condições de operação real, nomeadamente em relação a critérios de latência e estabilização do processo de estabelecimento das chaves, cobertura da rede e topologia alcançada, condições de fiabilidade propiciadas pela mesma e impacto no consumo energético e adequação às limitações de computação e de comunicação dos dispositivos utilizados como nós das RSSF (vulgarmente conhecidos por nós sensores ou *motes*).

Em geral, as propostas realizadas não abarcam uma discussão sistemática e completa dos anteriores critérios. Em muitos casos, são apenas realizados estudos de índole teórico, sobre condições de cobertura e de auto-organização topológica subjacente ao processo de estabelecimento das chaves. Em geral estes indicadores são calculados a partir de análises de teoria de grafos e cálculos probabilísticos, que abstraem a rede como um grafo (teórico) sobre o qual se pretendem analisar condições de cobertura face a hipóteses iniciais. Estes estudos não têm no entanto em conta a operação real subjacente à pilha IEEE 802.15.4, Zigbee ou ao funcionamento de diferentes variantes de protocolos de controlo de colisões e de acesso ao meio inerentes a essas pilhas. As análises publicadas também não permitem antecipar ou aproximar o verdadeiro impacto de diferentes protocolos de distribuição e estabelecimento de chaves face a condições reais de operação e face aos critérios enunciados.

Para se ultrapassarem as dificuldades anteriormente referidas, torna-se necessário dispor de ambientes de simulação ou de emulação que permitam analisar o comportamento dinâmico e o desempenho de diferentes protocolos, face a diferentes condições de operação das redes. Nesta abordagem, é particularmente interessante que tais condições de avaliação possam antecipar, da forma mais aproximada possível, as condições reais de funcionamento das RSSF.

1.2. Objectivos da dissertação e contribuições

O objectivo da presente dissertação teve em vista conceber, implementar e testar um ambiente de simulação para redes de sensores sem fios de grande escala, capaz de simular condições próximas do funcionamento real de redes de grande escala. Pretende-se que o ambiente possa ser usado para aferir e comparar, numa base experimental e de forma sistemática, as características de diferentes protocolos de distribuição probabilística de chaves par-a-par e baseados em esquemas de auto-organização aleatória da rede. Este ambiente de simulação tem por base um núcleo de eventos discretos e uma base inicial de simulação que implementa a camada MAC (protocolo de ligação de dados e de acesso ao meio) de acordo com a norma IEEE 802.15.4 ou Zigbee.

As contribuições da presente dissertação são em seguida apresentadas:

- C1: Disponibilização de uma plataforma de simulação como ferramenta genérica para prototipagem de protocolos de distribuição e estabelecimento de chaves para RSSF de grande escala e análise comparativa dos mesmos tendo em conta os seguintes critérios:
    - Latência e estabilização do processo de estabelecimento de chaves, de forma a medir o tempo de instalação das chaves que permitam obter a fiabilidade máxima possível por parte de aplicações e camadas superiores da pilha associadas à disseminação e encaminhamento de dados;
    - Cobertura efectiva da rede, que levem em linha de conta o impacto real do funcionamento da pilha IEEE802.15.4 (ou Zigbee) bem como o protocolo de acesso ao meio e a gestão do ciclo de vida de operação dos sensores;
    - Fiabilidade do protocolo, que permite verificar qual o impacto no sucesso da disseminação de dados em cada um destes quando desencadeado concorrentemente em diversos nós da rede;

- Latência na disseminação de dados após a estabilização da rede;
- Condições de sobrecarga de comunicação que permitam avaliar experimentalmente o impacto energético face à complexidade dos protocolos;
- Projecção de ataques internos, por captura de sensores, sobre os nós participantes da rede e o seu impacto nos indicadores anteriormente referidos, especialmente a taxa de cobertura e fiabilidade da disseminação de dados.

- C2: Utilização do ambiente de simulação para análise experimental dos critérios referidos nos seguintes protocolos: F-Leach [8], SecLeach [9], LHA-SP [10] e *framework* de Bohge [11], todos orientados para redes auto-organizadas segundo estruturas hierárquicas orientadas a *clustering*. Estes são protocolos que constituem referências da investigação na área dos protocolos de distribuição e estabelecimento de chaves em RSSF.
- C3: Apresentação sistemática dos indicadores comparativos dos anteriores protocolos que se encontram referidos em C1, com base nas observações experimentais baseadas em condições reais para sensores Mica 2 Motes da Crossbow funcionando em regime normal de acordo com a pilha IEEE802.15.4 ou Zigbee.

1.3. Organização do relatório

Os restantes capítulos do presente relatório de elaboração de dissertação, estão organizados do seguinte modo:

O capítulo 2 é dedicado à apresentação e análise crítica de trabalho relacionado com os objectivos e contribuições previstas para a dissertação. Esta apresentação é realizada segundo três vertentes principais. A secção 2.1 apresenta uma visão comparativa de simuladores ou emuladores para RSSF, tendo em vista a selecção de um ambiente de simulação de base para extensão e concepção de módulos de simulação, de modo a atingir os objectivos e contribuições previstas para a dissertação. A secção 2.2 apresenta a formulação de hipóteses para um modelo de adversário e tipologias de ataques que podem afectar as condições de segurança dos protocolos de distribuição e estabelecimento seguro de chaves em RSSF, nomeadamente quando funcionam de forma autónoma, em ambientes de grande escala e sem supervisão de operação. Finalmente, a secção 2.3 constitui uma síntese do estudo de esquemas base e protocolos de

distribuição e estabelecimento de chaves para condições de auto-organização aleatória da rede, bem como para organizações topológicas hierárquicas ou baseadas em *clustering*.

O capítulo 3 apresenta a especificação e concepção da plataforma final que contém o núcleo base de simulação apresentado e escolhido no capítulo 2.1. A secção 3.1 detalha os mecanismos que fazem parte desse núcleo inicial e a sua concepção no mesmo. A secção 3.2 apresenta depois as extensões que foram feitas à plataforma começando, na secção 3.2.1 e 3.2.2, com o detalhe de dois mecanismos importantes de tratamento de eventos que fornecerão as primitivas necessárias para a criação dos módulos adicionais à plataforma, detalhados na secção 3.2.3. Terminando o capítulo, a secção 3.3 apresenta uma visão geral da arquitectura final da plataforma, mostrando o enquadramento do núcleo base de simulação com a implementação dos módulos adicionais, resumindo também as funcionalidades oferecidas pela mesma e o modo de interacção com o utilizador.

O capítulo 4 trata da especificação e implementação dos protocolos de distribuição e estabelecimento de chaves criptográficas que foram propostos como elemento de estudo e avaliação comparativa sobre a plataforma implementada. Inicialmente é apresentado, na secção 4.1, o modelo assumido de rede assim como as características dos nós que dela fazem parte. Na secção 4.2 é abordada uma problemática do encaminhamento *multi-hop* em alguns protocolos. Nas restantes secções são então apresentadas as especificações dos protocolos, divididas pelas duas fases principais: fase de organização da rede e fase de operação.

Avançando para o capítulo 5 são tratados aspectos directamente relacionados com a implementação da plataforma de simulação, assim como dos protocolos propostos. Na secção 5.1 é apresentada a arquitectura de implementação, assim como a descrição dos componentes que dela fazem parte. Finalmente a secção 5.2 apresenta uma aproximação à dimensão de todo o trabalho efectuado.

O capítulo 6 é dedicado à avaliação experimental dos protocolos descritos no capítulo 4 enquanto operam sobre a plataforma descrita no capítulo 3. Inicialmente são apresentadas as parametrizações de teste, assim como a topologia da rede usada nos mesmos. As restantes secções apresentam os indicadores obtidos face aos critérios apresentados acima, divididos pelas várias fases dos protocolos. Por fim, o capítulo 7 apresenta o resumo das principais conclusões do trabalho desenvolvido, bem como apresenta algumas direcções de trabalho futuro.

Adicionalmente, o relatório da dissertação contém um anexo com uma introdução geral às Redes de Sensores sem Fios e à problemática associada às diversas vertentes do estudo da segurança destas redes. Este anexo constitui uma síntese complementar associada ao estudo preliminar para o enquadramento do tema da dissertação e dos seus objectivos. A leitura deste texto não é obrigatória para a compreensão do enquadramento e objectivos da dissertação, sendo apenas complementar e indicativa do estudo que teve de ser efectuado na fase de preparação.

2. Trabalho Relacionado

2.1. Ambientes de simulação e emulação

Redes de sensores sem fios são geralmente formadas por um grande número de sensores ligados em rede. Embora a melhor abordagem para estudar estas redes e respectivos algoritmos fosse o uso de sensores reais, tal estudo fica comprometido devido ao custo, ainda alto, destes aparelhos. Assim torna-se necessário recorrer ao uso de simuladores e/ou emuladores para efectuar o estudo antes da actual implementação.

Dentro do domínio da emulação/simulação existem várias *frameworks*, cada uma com características diferentes relativamente ao ambiente simulado das redes de sensores e o próprio formato dos mesmos. Em seguida são apresentados três ambientes de simulação e emulação que foram estudados e apresentam diversas vantagens no seu uso para a avaliação de protocolos.

2.1.1. TOSSIM

Desenvolvido pela Universidade de Berkeley [12], é um simulador e emulador que trabalha ao nível do bit, ou seja, para cada bit transmitido ou recebido é gerado um evento em vez de gerar para o pacote inteiro. É capaz de simular a execução de código nesC em TinyOS, o mesmo que corre em sensores reais, especialmente em sensores do tipo Mica Mote e permite ainda a emulação de hardware com mapeamento para eventos.

É portanto um simulador e emulador com altos graus de fiabilidade comparado com as aplicações a correr em sensores reais, mas devido à sua natureza complexa de emulação tem uma escalabilidade baixa relativamente a outras *frameworks*. Outra limitação é o facto de todos os sensores terem de correr o mesmo código, não podendo ser usado para avaliar aplicações heterogéneas e o tempo de execução do mesmo em cada sensor é assumido como zero.

Com a *framework* são oferecidos dois modelos de rádio, '*simple*' e '*lossy*'. O primeiro é um modelo determinista que coloca todos os nós numa única célula. Todos os bits são transmitidos

sem erro e dois sensores podem transmitir ao mesmo tempo, o que poderá originar na realidade um pacote corrupto mas a probabilidade de isso acontecer é muito baixa devido ao protocolo CSMA do TinyOS. Já o modelo '*lossy*' é um modelo probabilístico, que distribui os sensores como nós de um grafo ligados por um valor entre si. Esse valor corresponde à probabilidade de um bit enviado por um dos sensores vir a ser um bit corrupto.

Apesar de ser uma *framework* fiável, assume algumas simplificações que podem originar comportamentos imprevistos. Exemplo disso é a não preempção de interrupções que, em sensores reais, podem deixar a aplicação num estado inválido e deixarem de correr. Existem outras limitações como a não existência de um modelo de consumo energético e interface gráfica, problemas essas corrigidos nas extensões PowerTossim e TinyViz respectivamente.

2.1.2. Freemote

Freemote [13] é uma ferramenta em Java para a emulação, simulação e desenvolvimento de código para sensores baseados na norma IEEE 802.15.4, mais concretamente sensores que correm uma *jvm* optimizada (Squawk, Sentilla Point) e plataformas (Java cards, SunSPOT). A fiabilidade do ambiente é atingida misturando nós emulados com nós reais que comunicam com o simulador através de um nó especial que faz ponte entre ambos. Este ambiente tem também uma divisão arquitectural em três camadas independentes ligadas por interfaces: nível de aplicação, nível de encaminhamento e nível *data link* e físico.

É um bom emulador capaz de simular novos algoritmos para redes de sensores sem fios de larga escala, suportando um grande número de sensores emulados, incluindo reais, baseados na norma IEEE 802.15.4. No entanto não é uma ferramenta direccionada para a análises de performance dos algoritmos e possui um modelo de rádio bastante limitado onde é assumindo que não existem obstáculos entre sensores.

2.1.3. jProwler

JProwler [14] é a versão java do Prowler, um simulador de redes de sensores sem fios que captura a natureza baseada em eventos do TinyOS. É bastante escalável, capaz de simular uma larga quantidade de sensores com qualquer tipo de aplicação (heterogeneidade) e ainda sobre

uma topologia dinâmica da rede formada por estes, apesar do custo de cálculo das vizinhanças de cada sensor ser elevado.

O simulador é constituído por um modelo de rádio não determinista e por uma modelo de camada MAC. Ambos os modelos seguem a especificação da norma IEEE 802.15.4. O simulador é capaz de reproduzir as transmissões de rádio e propagações, incluindo colisões em redes *ad-hoc* e operações da camada MAC. Estes modelos são baseados no cálculo de intensidade de sinal combinado com erros aleatórios. Já a comunicação na camada MAC é modelada por um canal de eventos que simula o protocolo CSMA dos sensores Berkeley, introduzindo tempos de espera nas transmissões (ex: *BackOffTime*) e gerandos eventos na conclusão das mesmas para as mensagens transmitidas.

Os benefícios deste simulador passam pela fácil prototipagem de aplicações, integração de diferentes algoritmos, uma boa interface gráfica para visualização e *debug* da rede, facilidades para analisar o desempenho de soluções com recurso a métricas gráficas, entre outros. No entanto não é fornecido um modelo de consumo energético, apenas são dados dois modelos de rádio e apenas um protocolo MAC do TinyOS está implementado por defeito. Mas, devido à facilidade de extensibilidade da ferramenta através de módulos, estes e outros problemas podem ser ultrapassados, fazendo do jProwler um bom simulador para a validação e avaliação de protocolos e outras funcionalidades, quer seja para *debugging*, optimização ou acerto de parâmetros. Tudo isto faz com que seja um simulador bastante fiável ao simular o comportamento dos nós quando sujeitos à operação real sobre a pilha IEEE 802.15.4.

2.1.4. Resumo

A presente dissertação orienta-se para redes de grande escala e, portanto, o Tossim é um simulador pouco escalável. Não menos importante é a fiabilidade do simulador e, neste ponto, é onde o uso da *framework* do freemote poderá trazer algumas complicações devido ao seu modelo de rádio bastante simples que admite não existirem obstáculos.

Assim, de entre os três, o simulador que melhores características iniciais ofereceu foi o jProwler, conseguindo equilibrar melhor entre a fiabilidade da simulação e o estudo, com fins estatísticos, de redes de sensores de fios de larga escala. Como tal este foi o simulador escolhido para integrar o núcleo base de simulação. No entanto o simulador ainda é bastante limitado para

o estudo dos critérios apresentados na introdução, tornando indispensável a criação de módulos adicionais que estão referidos no capítulo 3.

Por fim, apresenta-se a respectiva grelha de critérios que resume um pouco o que foi dito acima sobre cada simulador e serve de fundamento para a escolha do jProwler:

Critérios	TOSSIM	Freemote	jProwler
Linguagem	nesC	Java	Java
Simulação/Emulação	Emulação	Simulação/Emulação	Simulação
Escalabilidade	Limitada	Alta	Alta
Usabilidade	Díficil	Fácil	Fácil
Verificação e análise de protocolos/performance	Sim	Sim (limitado)	Sim
Fiabilidade da simulação	Alta	Média	Alta
Trace/Debug gráfico	Sim (TinyViz)	Sim	Sim
Modelo de consumo energético	Sim (PowerTOSSIM)	Sim (plugin)	Sim (plugin)
Rede dinâmica/estática	Dinâmica/Estática	Dinâmica/Estática	Dinâmica/Estática
Modelos de Rádio	Probabilistic, Deterministic	Simples (inexistência de obstáculos)	Probabilistic, Deterministic
Tipos de Sensores	MICA Motes	Java Motes baseados em IEEE 802.15.4	MICA Motes
Estruturação	Por componentes	Por camadas	Por camadas
Heterogenidade de aplicações	Não	Sim	Sim
Baseado em eventos	Sim (por bit)	Sim	Sim
Extensibilidade	Complexa	Simples	Simples

Tabela 2.1 - Comparação entre simuladores de RSSF

2.2. Modelo de adversário

As redes de sensores sem fios, devido a operarem durante longos períodos sem intervenção humana estão, portanto, expostas a vários ataques sem que se tenha conhecimento dos mesmos. O facto de operarem sobre comunicações sem fios torna-as um alvo fácil pois estão abertas a toda a gente, incluindo atacantes que nem sequer precisam de capturar fisicamente um nó para causar danos em toda a rede. Como já foi referido, o consumo energético é também um factor essencial na construção destas redes e pode muito bem ser aproveitado como vantagem para os atacantes, que tentam gastar os poucos recursos energéticos disponíveis. Não obstante, também é uma limitação na escolha de algoritmos criptográficos para tornar a rede mais segura. Por estes

motivos é importante definir um modelo de adversário. Em seguida é apresentado um estudo dos níveis MAC e de encaminhamento que podem ser alvos do atacante. Após entender os conceitos por detrás de cada nível serão apresentadas as topologias de ataques sobre os mesmos. No final será definido então o modelo de adversário assumido para a realização da presente dissertação

2.2.1. Nível MAC e nível físico

Tendo em conta as características dos sensores e as suas necessidades [15], especialmente a do consumo energético baixo, os protocolos tradicionais de controlo de acesso ao meio não são adequados, tornando necessária a existência de novos protocolos específicos para estas redes. Uma primeira aproximação tem em conta protocolos que evitam colisões como TDMA, FDMA e CDMA mas tal implicaria baixa escalabilidade, como em caso de adição de novos nós e deslocação dos existentes, e ainda pouco aproveitamento do canal devido à divisão do mesmo. Assim tem-se como melhor aproximação protocolos de contenção como CSMA que, de facto, melhoram a escalabilidade, flexibilidade e latência, mas no entanto não podem ser implementados seguindo a especificação original pois efectuam uma gestão energética ineficiente, visto os nós encontrarem-se constantemente à escuta, procurando possíveis colisões e ainda a possibilidade de retransmissões. Actualmente a norma usada em sensores é a IEEE 802.15.4 [1] que detalha o nível MAC e nível físico, mas existem outras alternativas.

S-MAC e outros

O principal objectivo é reduzir o custo energético, mantendo a flexibilidade associada a protocolos de contenção. Para tal os nós transitam entre o estado adormecido, com tempo variável, e em escuta, com tempo fixo. Os nós sincronizam-se também nos tempos para dormir através da troca de mensagens e, da mesma maneira, notificam nós vizinhos para dormirem durante transmissões que não lhes sejam destinadas. Finalmente, o mecanismo de contenção combina CSMA com RTS-CTS-DATA-ACK e é ainda usado um esquema de *message-passing*, que permite enviar vários pacotes de seguida usando apenas um par RTS-CTS.

Posteriormente foram apresentados novos protocolos com novas abordagens e grande parte deles foram melhorias do S-MAC. São exemplos desses protocolos NanoMAC [16], T-MAC

[17], B-MAC [18] e Z-MAC [19]. A título de exemplo, o protocolo T-MAC consegue melhorar o custo energético introduzindo tempo activo variável.

IEEE 802.15.4

Tipos de Nós e Topologia

São previstos dois tipos de nós: FFDs (*Full Function Devices*) e RFDs (*Reduced Function Devices*). FFDs podem assumir o papel de coordenadores da rede, funcionando como *gateways* para outras redes, de coordenadores, realizando tarefas de encaminhamento de dados e funções de organização com outros coordenadores, ou ainda de participantes, que só podem comunicar com coordenadores. Já os RFDs apenas podem assumir o papel de participantes.

Duas topologias são previstas: em estrela e *peer-to-peer*. Na primeira todas as comunicações passam pelo coordenador da rede. Já na segunda os nós podem comunicar directamente entre si desde que sejam alcançáveis entre si.

Acesso ao meio

São suportados dois modos de acesso ao meio de comunicação: modo *beacon-enabled* e modo *non-beacon-enabled*. No primeiro modo um *frame* especial (*beacon*) é enviado periodicamente pelo coordenador da rede, o qual é usado para os nós se sincronizarem. O tempo entre a ocorrência de dois *beacons* (*Beacon Interval*) é dividido num período de actividade e outro de inactividade onde os nós costumam adormecer. Durante o período de actividade o *beacon* é dividido em *slots* temporais que podem ser usados para transmitir. Assim os nós vão competir para ganhar acesso a estes *slots* bem definidos através do algoritmo de *slotted* CSMA/CA. A estes *slots* dá-se o nome de CAP, mas existem outros *slots* que podem ser reservados pelos nós, fazendo parte do CFP. Por fim, no segundo modo sem *beacon* os nós tentam aceder ao meio pelo algoritmo *unslotted* CSMA/CA.

O algoritmo usado para evitar colisões procede à escuta do meio e se detectar alguma procede a um *backoff* aleatório antes de escutar de novo o canal. No caso de *slotted* CSMA/CA esse *backoff* coincide com a dimensão das *slots* e o nó procede à transmissão durante o próximo *slot* de *backoff* que se encontre disponível. Já na transmissão consideram-se dois sentidos: *Uplink* e *Downlink*. Transferências *Uplink*, de um nó para o coordenador são sincronizadas com o *beacon*

e devem ocorrer durante o período de actividade do mesmo (*data* + *acknowledgment*). Já transferências *Downlink*, do coordenador para os nós têm de vir assinaladas no *beacon* para que os mesmos possam, explicitamente, pedir ao coordenador os dados que lhes são destinados.

2.2.2. Ataques ao nível MAC e físico

No nível físico o atacante pode criar interferência nas frequências de rádio, bloqueando assim os meios de comunicação entre nós. A única solução passa por os nós, coordenadamente, mudarem para novas frequências. Ataques no nível MAC podem-se dividir em duas categorias:

Ataques que seguem o protocolo MAC

Nesta categoria o atacante pode actuar como um membro legítimo da rede. Um método simples de ataque é inundar a rede com imensos pacotes e de elevada dimensão, causando uma perda de performance considerável na mesma (negação de serviço). Em alternativa apenas um nó específico pode ser atacado e, devido ao mecanismo de *Downlink*, tanto o nó atacado como o coordenador da rede sofrem com isso, gastando eventualmente os recursos energéticos. O atacante pode também explorar variáveis do algoritmo como a *aMacBattLifeExt* (funcionamento a bateria) que permite diminuir o tempo de *backoff*, tentando portanto ganhar prioridade de acesso ao canal face aos restantes nós legítimos. O impacto destes ataques [20] é o decréscimo da probabilidade de sucesso na entrega de pacotes entre nós legítimos e ainda um aumento do *delay* na entrega dos mesmos.

Ataques que não seguem o protocolo MAC

Nesta categoria de ataques é prevista a captura física de nós que depois são alterados pelo atacante e repostos na rede. Com isso o atacante é capaz de atacar a comunicação em geral segundo um modelo bizantino. Pode portanto interceptar, retransmitir e injectar mensagens na rede, fazendo-se passar, ou não, por um nó legítimo ou ainda tem a opção de copiar a identidade de um. Vários ataques são possíveis nesta categoria, como por exemplo, fabrico e injecção de pacotes, *replying*, *tampering*, *spoofing*, entre outros. Em alternativa o atacante pode apenas alterar ligeiramente o funcionamento do protocolo MAC e, por exemplo, não respeitar as duas fases previstas para detectar possíveis colisões, ganhando mais tempo de acesso ao meio e causando possíveis colisões.

2.2.3. Nível de encaminhamento

A norma mais usada em sensores é a IEEE 802.15.4 mas a mesma só específica o nível físico e MAC. Assim, para níveis superiores, como o nível de encaminhamento, existem várias propostas, de entre as quais os algoritmos de distribuição de chaves abordados na secção seguinte que definem o encaminhamento que os dados devem ter na sua entrega à *base station*. A existência de várias propostas evidenciam o facto de os protocolos em redes de sensores serem muito específicos para a aplicação a ser suportada (*application driven*). Uma rede de sensores é uma rede *ad-hoc* e como tal os protocolos de encaminhamento desenhados e propostos têm em conta o aspecto de *multi-hop*.

2.2.4. Ataques ao nível de encaminhamento

Neste nível o atacante pode realizar dois tipos de ataques: passivos e activos. Um ataque passivo a este mecanismo pode ser difícil de detectar pois não altera os resultados esperados e geralmente apenas recolhe informação possivelmente sensível por escuta no canal. Já um ataque activo tenta atrapalhar o mecanismo de encaminhamento intencionalmente, modificando mensagens, ganhando autorizações e até tomando controlo de partes da rede. Para tal o atacante pode injectar pacotes na rede, modificar pacotes existentes em circulação e descartá-los. Os ataques podem também considerar-se externos, quando os nós maliciosos não pertencem à rede, ou internos, quando pertencem visto terem sido capturados pelo atacante, modificados e introduzidos de novo na rede. Assim, os ataques ao encaminhamento, relevantes aos mecanismos de distribuição e estabelecimento de chaves, podem ser divididos em 2 categorias:

Ataques à selecção de rotas

Com esta categoria de ataques o atacante pretende levar nós legítimos a escolherem rotas que o envolvam directamente para que possa escutar as mensagens trocadas ou lançar vários tipos de ataques. Por analogia a uma rede hierárquica baseada em *clusters*, um atacante tenderá a comportar-se como um *cluster head* para atrair os restantes nós a formarem *cluster* com ele. Consideram-se os seguintes ataques à selecção de rotas:

Hello Flood Attacks → Os nós para se conhecerem trocam pacotes *HELLO* e ao receberem um assumem que o emissor está à distância de 1 *hop*. Ora se o atacante possuir um portátil com maior intensidade de sinal pode emitir estes pacotes e muitos nós legítimos irão vê-lo e provavelmente escolhê-lo para a composição de rota porque supostamente está a uma distância curta de 1 *hop*.

Sinkhole Attacks → O objectivo é atrair tráfego da rede para si como a inundação de pacotes *HELLO*, mas ao contrário do último ataque o atacante não precisa de ter mais intensidade de sinal, bastando-lhe enviar mensagens alteradas, segundo a especificação do protocolo, que lhe tornam uma escolha mais correcta face a outros nós legítimos.

Wormhole Attacks → Em ataques *wormhole* o atacante possui normalmente dois nós e uma ligação física entre eles que não pertence à rede e que só ele tem acesso. Com isto é possível fazer um túnel de comunicação entre estes dois nós que serão vistos como atractivos pelos nós vizinhos no estabelecimento de rotas pois geralmente possuem baixa latência face à hipótese alternativa de *multi-hop*. Com isto o atacante pode direccionar tráfego para estes dois nós e dar uma falsa aparência da topologia da rede.

Sybil Attacks → Num ataque deste género um nó malicioso faz-se passar por outros nós reportando várias identificações aos vizinhos. O objectivo é aumentar a probabilidade de os vizinhos escolherem o nó para o estabelecimento de rotas. Ataques *Sybil* são também muito fortes contra protocolos de tolerância a falhas que estabelecem várias rotas para resistir a ataques. Essas rotas podem ser diferentes do ponto de vista do nó legítimo, mas na realidade passam todas pelo nó malicioso.

Ataques após o estabelecimento de rotas

Consideram-se os seguintes ataques após o estabelecimento de rotas:

Blackhole Attacks → Estando o atacante no meio de uma rota, este pode descartar pacotes que passem por ele, não respeitando o protocolo *multi-hop*. Em alternativa pode descartar selectivamente, evitando que os emissores se apercebam de que a rota está em baixo já que também não o conseguem distinguir de um ataque deste género.

Spam Attacks → O objectivo é gerar um imenso número de mensagens inúteis de modo a gastar recursos da rede, especialmente a capacidade energética dos sensores que vão seguir o protocolo *multi-hop* e gastar recursos para fazer encaminhamento das mensagens.

2.2.5. Modelo de adversário definido

Foi visto que uma rede de sensores pode sofrer ataques nos níveis físicos, MAC e de encaminhamento. Partindo do princípio, o modelo de adversário deve contemplar a presença de um atacante forte, tipicamente com mais recursos que o tradicional nó da rede. Este atacante segue um modelo de Dolev-Yao [21], ou seja, é um atacante externo que ataca as comunicações em geral (intercepção de mensagens, omissão, etc..). Com base neste modelo também é previsto que o atacante possa operar em qualquer localização da rede e que todas as suas instâncias (nós maliciosos) partilham informação, não necessariamente pelo mesmo meio de comunicação dos nós legítimos. No entanto este modelo não prevê a captura física de nós, ou seja, toda a categoria de ataques internos. Assim o modelo definido tem de estender o modelo de Dolev-Yao ao considerar a captura física de nós, mas não de todos porque isso já implicaria o total controlo da rede e de nada adiantaria o estudo do impacto de diversas classes de ataques.

Para o âmbito da presente dissertação e tendo em conta os objectivos traçados, o modelo de adversário não contempla ataques ao nível físico e nível MAC pois não é a esse nível que o a dissertação se orienta. Já no nível lógico podem-se dividir os ataques em externos (às comunicações) e internos (captura física de nós e modificação dos mesmos).

A topologia de ataques externos, que segue o modelo de Dolev-Yao, pode ser defendida por um protocolo que seja seguro segundo o mesmo modelo e/ou segundo a recomendação X.800 da arquitectura OSI [22]. Na investigação de RSSF têm sido propostas diversas aproximações e analisadas as suas implementações a nível de custo energético, complexidade de comunicação e fiabilidade. Dentro dessas contribuições destacam-se as seguintes: [23], [24] e [25]. A sua posição na arquitectura dos sensores pode ser vista na secção 2.3.1.

Já em ataques internos assume-se um modelo bizantino [28] onde o atacante pode capturar nós, modificá-los e inseri-los novamente na rede sob o seu total controlo e onde poderão ser desencadeados os mais variados ataques. Os nós comprometidos, sobre os quais o atacante pode descobrir as chaves, podem ser usados para seguir, ou não, à letra o protocolo e com isso tenta

obter um maior conhecimento e potencial controlo da rede. Pode também apenas atrapalhar a execução normal do protocolo entre os restantes nós, por exemplo na descoberta de caminhos entre nós que não partilhem directamente uma chave simétrica ou na formação de *clusters*.

Um modelo bizantino é muito geral e, para efeito da realização desta dissertação e respectiva avaliação experimental, serão considerados apenas os ataques ao encaminhamento que foram estudados na secção 2.2.4, os quais são também hipóteses de ataque sobre os protocolos de distribuição e estabelecimento de chaves tanto na fase de formação hierárquica da rede, como na fase de operação onde a rede já se encontra formada.

2.3. Esquemas e protocolos de distribuição segura de chaves

Comunicações seguras entre sensores só são garantidas se ambos forem capazes de correr algoritmos criptográficos entre si. Devido às suas limitações, criptografia assimétrica tem de ser excluída sobrando portanto criptografia simétrica. Como tal, chaves seguras e segredos criptográficos têm de ser partilhados entre sensores, os quais são obtidos por um esquema de distribuição de chaves. De entre os métodos existentes, estudos recentes mostram que a pré-distribuição de chaves [3] é o mais indicado, ou seja, as chaves são inseridas nos nós sensores antes do *deployment*[2].

Nas restantes secções do presente capítulo serão abordados esquemas base de pré-distribuição de chaves e os protocolos de distribuição de chaves que serão alvo de simulação na presente dissertação. Dito isto será feita uma introdução à arquitectura da pilha de segurança nos sensores e nos capítulos 2.3.2 e 2.3.3 serão apresentados os esquemas base e os protocolos respectivamente. Os esquemas estudados serão todos probabilísticos e os protocolos demonstrarão uma auto-organização aleatória de redes baseadas em *clusters*

2.3.1. Arquitectura de segurança dos sensores

A arquitectura de segurança dos sensores dispostos na rede pode ser especificada e estruturada pela seguinte 2.1.

[2] Ao longo deste relatório utiliza-se o termo "*deployment*" em língua inglesa, como termo comum usado na terminologia usual das Redes de Sensores sem Fios. Este termo poderia ser traduzido por "implantação". O termo em língua inglesa é no entanto utilizado por facilidade de interpretação conceptual face à literatura da área.

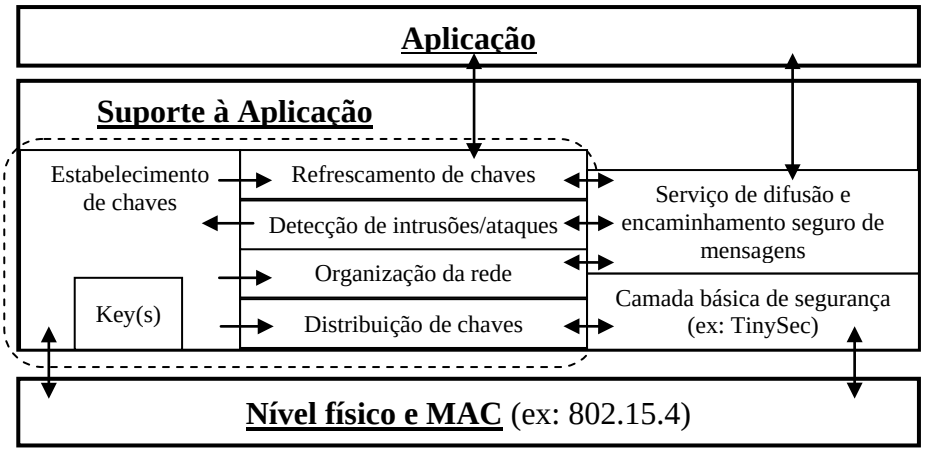

Figura 2.1 - Arquitectura de segurança dos sensores

Os nós sensores, ao virem de fábrica, possuem uma camada básica de segurança que lhes permite criar comunicações seguras entre si, de encaminhamento e não só, completamente protegidas de ataques externos que sigam o modelo de Dolev-Yao. No entanto o que não é definido é como essa chave (ou chaves) é obtida e é nesse domínio que se situa um protocolo de distribuição e estabelecimento de chaves. Um protocolo simples pode simplesmente atribuir uma chave global a todos os nós e a camada de segurança trata de cifrar todas as mensagens com a mesma, garantindo a confidencialidade dos dados.

No entanto um protocolo de distribuição de chaves, com especial nota para os propostos na presente dissertação, geralmente redefine a camada básica de segurança para se adaptar às suas necessidades, como o fornecimento de garantias de autenticidade, confidencialidade, integridade, entre outras. Estes protocolos acabam também por ser divididos em vários mecanismos nos quais se destaca o processo de organização da rede que pode dar origem a redes totalmente planas ou hierárquicas e em consequência da geração e distribuição de chaves conforme a arquitectura da mesma. Já o mecanismo de detecção de intrusões é responsável por reportar possíveis intrusões ou ataques (pode obter a informação directamente do mecanismo de encaminhamento) e, face a este problema, o mecanismo de estabelecimento de chaves tem de tomar a decisão de refrescar as chaves nos nós, ou de proceder a uma reorganização da rede que pode ser total. A reorganização pode também ser efectuada por outros motivos, como por exemplo a criação de um mecanismo de resiliência pró-activa.

Para terminar, o âmbito da presente dissertação pode ser visto na imagem como a área a tracejado. Os protocolos propostos irão redefinir a camada de segurança e definir o seu próprio

mecanismo de encaminhamento seguro de mensagens, que tratará apenas da problemática de encaminhamento de dados de um nó sensor até à *base station* e vice-versa.

2.3.2. Esquemas básicos de pré-distribuição probabilística de chaves

2.3.2.1. Pré-distribuição básica de uma chave global

Considera-se o esquema mais simples em que, na fase antes do *deployment*, os nós são inicializados com uma chave partilhada entre todos. Graças a esta chave ganha-se protecção contra ataques externos mas, face a uma captura física de um nó, o atacante pode obter a chave partilhada e com isso conseguir comprometer todos os canais de comunicação da rede. Devido a esta enorme fragilidade foram propostos vários esquemas aleatórios de distribuição de chaves, nos quais se destaca o esquema de Eschenauer e Gligor.

2.3.2.2. Pré-distribuição aleatória de chaves

Esquema proposto por Eschenauer e Gligor [4], baseado na partilha probabilística de chaves e um mecanismo simples de descoberta de chaves partilhadas. O esquema é dividido em 3 fases:

Pré-distribuição de chaves: Antes do *deployment* os nós são iniciados com um conjunto de chaves aleatórias (*key ring*[3]) de uma grande colecção de chaves (*key pool*). A cada chave está associado um identificador da mesma que será introduzido no nó a par da chave. Com esta fase pretende-se que, com um pequeno número de chaves, os nós possam partilhar pelo menos uma entre si com grande probabilidade.

Descoberta de chaves partilhadas: Após o *deployment* os nós terão de comunicar e descobrir chaves partilhadas com os vizinhos (dentro do alcance do rádio), bastando apenas uma para se estabelecer uma comunicação segura e directa. Para evitar que um atacante possa obter as chaves, durante a troca de informação apenas os identificadores são passados em claro, o que

[3] Ao longo deste relatório utilizam-se os termos "key ring" e "*key pool*", em língua inglesa, como termos comuns usados na terminologia usual das Redes de Sensores sem Fios. Estes termos poderiam ser traduzidos por "anel de chaves" e por "universo de chaves". Os termos em língua inglesa são no entanto utilizados por facilidade de interpretação conceptual face à literatura da área.

permite aos nós descobrirem se partilham alguma chave sem que o atacante descubra qual é, visto que não o consegue inferir só com base no identificador.

Descoberta de caminhos seguros: Após a descoberta de chaves partilhadas podem existir ainda nós que queiram estabelecer uma ligação mas não partilham chave alguma. Assim, nesta fase final, estabelecem-se caminhos entre estes através de nós intermédios que possuam ligação directa entre si. Se o grafo final for ligado então uma ligação indirecta pode ser estabelecida entre um emissor e um receptor. Para isso o emissor gera uma chave (*path-key*) e envia a mesma, para o receptor, de forma segura pelo caminho descoberto.

O grande desafio passa então por escolher valores correctos para os parâmetros (*key pool* e *key ring*) de modo a que, com alta probabilidade, os nós vizinhos consigam estabelecer ligações seguras. Eschenauer e Gligor propuseram também um modelo teórico para o cálculo destes. parâmetros mas este não tem em conta a posição física dos nós e as condições reais de operação.

2.3.2.3. Shared-Key Threshold R-KPS: q-Composite R-KPS

Esquema [5] que representa uma modificação ao anterior, sendo diferente apenas no tamanho da *key pool* e no facto de usar múltiplas chaves para estabelecer uma ligação segura, o que aumenta a resiliência face à captura de nós. Inicialmente extrai-se a *key pool* do espaço total de chaves e, para cada nó, um certo número de chaves aleatórias são escolhidas da *key pool* e inseridas nos respectivos *key rings*. O objectivo é que nós vizinhos partilhem pelo menos um certo número de chaves ‘q’. Durante a descoberta de chaves partilhadas os nós vizinhos calculam uma chave específica para cada ligação com base numa função de *hash* entre todas as chaves partilhadas.

Por aqui entende-se que o tamanho da *key pool* é importante na medida em que se for muito grande a probabilidade de nós vizinhos partilharem pelo menos ‘q’ chaves diminui e se for muito pequeno a segurança face à captura diminui. Já ‘q’ também é um parâmetro importante visto que se for muito pequeno acaba por não trazer grandes vantagens face ao esquema anterior, mas se for muito grande então pode começar a revelar grandes fracções da rede assim que um número suficiente de nós sejam capturados. No entanto se apenas um número pequeno de nós for capturado, um número grande para ‘q’ é benéfico porque o adversário não consegue inferir grande coisa com tão poucos nós a partilharem tantas chaves.

2.3.2.4. Pré-distribuição de chaves com conhecimento da localização

Em alguns casos a informação da localização, não necessariamente exacta, do *deployment* dos nós facilita a pré-distribuição de chaves pois pode-se determinar os sensores que provavelmente serão vizinhos entre si e fazer uma melhor escolha das chaves que cada um possuirá. Assim o que muda para os anteriores esquemas é apenas o modelo de *deployment*, sobre o qual passamos a conhecer mais informação, e a fase da pré-distribuição que pode ser optimizada devido ao conhecimento do mesmo.

Closest R-KPS

Esquema proposto por Liu e Ning [6] onde são distribuídas chaves entre pares de sensores que possam ser vizinhos. O modelo de *deployment* deverá então reflectir essa probabilidade de vizinhança de alguma maneira. Na teoria se dois sensores aparecem no sinal de rádio um do outro, com alta probabilidade, então devem partilhar uma chave em comum. O esquema, para cada sensor *a* descobre então os vizinhos e para cada um desses vizinhos *b*, cria uma chave $K_{a,b}$ que distribui por ambos. Quanto mais precisa for a informação da localização, maior será a eficiência do esquema (maior probabilidade de obter um grafo totalmente ligado). No entanto uma grande limitação vem dos próprios sensores, mais concretamente na sua capacidade de armazenamento já que podem ser geradas muitas chaves para cada um.

Location-based R-KPS

Mais um esquema proposto por Liu e Ning [6] que toma em consideração as limitações do anterior, criando no entanto, um *tradeoff* entre a hipótese de dois sensores vizinhos estabelecerem ligação face às limitações de armazenamento e a segurança face à captura física. A técnica é baseada na pré-distribuição baseada em polinómios. Toda a área de *deployment* é dividida em células (polinómios invariáveis) e o modelo refere que os sensores são espalhados por essas células, ou seja, à partida saber-se-á com grande probabilidade a célula onde o sensor será largado. A diferença para o esquema anterior é que aqui um sensor não vai partilhar uma chave diferente com todos os seus vizinhos mas, em vez disso, vai possuir segredos do polinómio onde se insere e dos polinómios adjacentes, tornando possível estabelecer ligações seguras intra-polinomiais e inter-polinomiais se forem polinómios adjacentes. No geral isto vai

reduzir a quantidade de chaves que cada sensor terá de armazenar mas a segurança será reduzida pois um atacante pode capturar um sensor e obter os segredos dos polinómios que o mesmo conhece, o que engloba bastantes sensores à sua volta.

Group-Based R-KPS

O modelo de *deployment* admitido prevê que os sensores são largados no campo em grupos sequenciais, tornando altamente provável que, sensores que façam parte do mesmo grupo, estejam próximos entre si e possam comunicar após o *deployment*. Esta é uma observação que pode optimizar a fase de pré-distribuição de chaves. O conhecimento do *deployment* pode também ser representado por uma função densidade de probabilidade. Quando a função é uniforme então conclui-se que os sensores podem ser largados em qualquer lugar com a mesma probabilidade [4], mas se não o for então pode-se aproveitar o conhecimento da localização de um sensor para optimizar também a fase da pré-distribuição.

Foi com estas duas optimizações que Du et al. [7] propôs um novo esquema. Uma primeira assunção é de que os sensores ficam estáticos relativamente à sua posição após o *deployment*. Já durante o mesmo os sensores são largados em grupos e cada um deles, dentro do seu grupo, segue uma distribuição de probabilidades gaussiana de onde deverá ficar na sua posição final com centro num determinado ponto de onde foi largado. Com isto em mente foi proposta também uma divisão da *key pool* por grupos já que dois nós distintos em grupos bastantes distantes não deverão partilhar chaves, não fazendo sentido portanto irem buscar as chaves ao mesmo espaço. No entanto deverá existir uma pequena intercepção de chaves entre grupos próximos. Em termos de resultados este esquema mostrou uma maior conectividade que o esquema probabilístico básico e, indirectamente, um aumento da segurança devido aos sensores não possuírem chaves desnecessárias.

2.3.3. Protocolos de distribuição e estabelecimento de chaves para RSSF hierárquicas

2.3.3.1. Leach

O protocolo Leach (Low-*energy Adaptive Clustering Hierarchy*) [26] assume uma rede de sensores populada por nós homogéneos, limitados energeticamente, que fazem chegar os seus

relatórios[4] de eventos à *base station*. Para tal o protocolo define uma formação hierárquica baseada em *clusters* onde os nós podem desempenhar funções de *cluster head* ou de nós membros.

A rede é então particionada em *clusters* e em cada um destes um nó, denominado *cluster head*, é responsável por criar e manter um esquema TDMA de envio de relatórios. Os restantes nós vizinhos, denominados nós membros, receberão do mesmo uma *slot* TDMA na qual terão a hipótese de trocar dados e fazer passar os seus relatórios para o mesmo. Este fica depois responsável por agregar os diversos relatórios e de produzir um resultado final o qual deverá ser entregue à *base station*. Assumindo que a *base station* poderá estar muito afastada do *cluster head* então o protocolo prevê que o mesmo deverá usar uma força de sinal suficiente para a alcançar, gastando uma quantidade multo alta de energia no processo. Para um nó membro o custo é bastante mais baixo, transmitindo apenas para o respectivo *cluster head*. O protocolo prevê também um sistema de rondas de forma a distribuir o peso de um nó ser *cluster head* por toda a rede, distribuindo assim o custo energético das transmissões de longa distância.

O protocolo assume a sincronização dos relógios dos nós e, no início de cada ronda, todos os nós decidem em simultâneo se vão desempenhar a função de *cluster head*, localmente e com base em critérios probabilísticos, tendo também em conta a última ronda em que cada um o foi de modo a distribuir justamente a carga. Os nós membros escolhem depois a que *cluster head* vizinho se devem juntar, geralmente ao que se encontra mais perto, permitindo uma optimização na força de sinal necessário para o mesmo receber os relatórios enviados.

A especificação do protocolo prevê a divisão de cada ronda em duas fases: *setup* (fase de organização) e *steady-state* (fase de operação). A fase de *setup* inicia-se com a auto-eleição de alguns nós a *cluster head*. Aqueles que se elegeram notificam os nós membros vizinhos. Cada nó membro recebe vários anúncios e decide depois com qual deve formar *cluster*, enviando um pedido de junção ao mesmo. Os *cluster heads* coleccionam os pedidos de junção durante um tempo determinado e no final constroem um esquema de *slots* temporais TDMA e difundem-no pelos vizinhos que se juntaram, terminando assim a fase de *setup*. Na fase de operação os nós membros respeitam a sua vez segundo as *slots* temporais e fazem chegar os seus relatórios ao *cluster head* que vai receber vários, agrega-os e envia o resultado para a *base station*.

[4] O termo "relatório" (ou "relatórios") será usado durante o resto do presente documento como a noção de captura de evento e geração de um relatório do mesmo para entrega à *base station*. A razão para o seu uso, em vez da entrega de "eventos" ou "dados" prende-se na facilidade de interpretação segundo o contexto onde o termo é inserido.

Resumindo, o protocolo Leach assume uma rede inicial plana de nós homogéneos e tende a convergir a mesma para uma formação hierárquica baseada em *clusters* onde certos nós assumem o papel especial de *cluster head* e ficam responsáveis por agregar dados de nós membros e fazê-los chegar à *base station*. Periodicamente a rede reorganiza-se seguindo a mesma especificação mas com o cuidado de distribuir a tarefa pesada de um nó ser *cluster head* por nós diferentes em cada ronda. O modelo de encaminhamento definido é bastante simples no sentido em que só assume comunicações *single-hop*, quer entre *cluster head* e nós membros, quer entre *cluster head* e base station.

Limitações tendo em conta uma avaliação experimental

Tendo em conta que a avaliação experimental será realizada para redes de grande escala é importante realçar que a especificação original do protocolo Leach e, consequentemente, dos protocolos F-Leach e SecLeach não se adequa directamente a tais dimensões. A razão prende-se no facto de o autor assumir que um *cluster head* é capaz de comunicar sempre, por *single-hop*, com a *base station* utilizando uma força de sinal maior que o normal. Tal assumpção é insustentável para redes na ordem dos milhares de nós onde a distância entre dois nós pode ser abismal. Assim, como contribuição da presente dissertação, foi abordada uma vertente *multi-hop* que se encontra detalhada no capítulo 4.2.

2.3.3.2. F-Leach

Proposto por Ferreira et al. [8], introduz segurança no protocolo Leach com o principal objectivo de evitar que nós não legítimos se possam tornar *cluster heads*, limitando assim ataques da génese do *sink-hole* para atacantes externos. Tal feito é alcançado com um esquema de pré-distribuição de chaves pelos nós.

Cada nó terá duas chaves, uma chave par-a-par com a *base station* e a última chave de uma cadeia de chaves gerada pela mesma que segue apenas um sentido e não é reversível. A ideia é fazer com que os *cluster heads* auto-eleitos se identifiquem primeiro perante a *base station*. Na fase de *setup*, cada *cluster head* ao difundir o anúncio inicial terá agora de garantir que o mesmo seja recebido pela *base station* e que o mesmo contenha uma prova de autenticação do mesmo. A prova é gerada através de um *mac* calculado com a chave partilha com a *base station*, a qual receberá vários anúncios e irá gerar uma lista de *cluster heads* autenticados na presente ronda.

Essa lista será depois difundida por toda a rede usando um esquema de *broadcast* seguro µTESLA [24] para os nós que receberem a lista da *base station* poderem autenticar a mesma.

Um nó membro, tendo conhecimento da lista de *cluster heads* autenticados, poderá depois aceitar qualquer anúncio de um que esteja nessa lista, garantindo assim que está a formar *cluster* com um nó que dará encaminhamento aos seus relatórios. O resto do protocolo decorrerá de forma quase idêntica ao protocolo Leach descrito acima, apenas com a introdução de um sistema de detecção de intrusos visto que os nós membros terão de se autenticar também perante a *base station* para os seus relatórios serem considerados pela mesma. Mais uma vez utilizam-se *macs* através das chaves partilhadas entre as duas entidades.

2.3.3.3. SecLeach

Protocolo proposto por Leonardo B. Oliveira et al. [9] tem como principal contribuição a introdução de segurança ao nível da comunicação par-a-par entre nós membros e *cluster heads* com recurso a chaves simétricas distribuídas pelos nós segundo o esquema básico de pré-distribuição aleatória de chaves de Eschenauer.

É proposto então que seja gerada uma *key pool* com chaves e respectivos identificadores e, antes do *deployment*, cada nó recebe um conjunto destas chaves de forma pseudo-aleatória. A atribuição das chaves é feita pelo identificador e será necessária uma função pseudo-aleatória que, para cada identificador de um nó, lhe atribua um conjunto de chaves. Devido a todos os nós terem conhecimento dessa função, os nós membros vão poder conhecer os identificadores das chaves que um *cluster head* tem e facilmente descobrem se partilham chaves ou não. Além disso, cada nó vai receber também uma chave par-a-par que partilha com a *base station*.

As fases do protocolo Leach sofrem com ligeiras alterações no conteúdo das mensagens, como a introdução de *nonces* e *macs* para a autenticação dos nós membros perante os *cluster heads*, garantindo assim segurança ao nível de autenticação, integridade e frescura.

2.3.3.4. LHA-SP

Protocolo proposto por Leonardo B. Oliveira et al. [10] que apresenta uma formação de uma topologia da rede em hierarquia com vários níveis por onde se dividem os sensores conforme as suas características, criando um sistema de *clusters*. Ao contrário do protocolo Leach não é

prevista a homogeneidade dos nós, muito pelo contrário. A atribuição de níveis aos nós é feita estaticamente antes do *deployment* e esse nível será atribuído conforme as capacidades dos mesmos, sendo a *base station* considerada como o nó de nível mais elevado.

Relativamente à formação em *clusters*, a especificação prevê, à semelhança de outros protocolos, um sistema de anúncio, pedido de junção e resposta. Esta sequência de troca de mensagens para formação de *clusters* vai-se iniciar nos nós de nível mais elevado primeiro e vai-se propagar até ao nível mais baixo (nível 1), ou seja, um nó de nível *h* só recebe um anúncio de um nó de nível *h*+1 quando este último já tiver formado *cluster* com um nó de nível *h*+2.

O esquema de pré-distribuição de chaves usado é determinístico e híbrido entre uma pré-distribuição duma chave global por todos os nós e uma chave par-a-par entre cada um e a *base station*. A chave global serve apenas para a organização da rede, ou seja, para a construção da hierarquia em *clusters* e após essa fase é descartada devido aos problemas de segurança que introduz. O protocolo, além da fase de organização, prevê também a fase de operação e a fase de manutenção. Em seguida apresenta-se uma breve descrição sobre cada uma das fases.

A fase de organização segue o esquema de anúncio-pedido-resposta e inicia-se quando os nós fazem difusão dando a conhecer aos nós de um nível inferior a sua disponibilidade para adoptarem novos nós membros. Aqueles que se decidem juntar enviam então o pedido e se forem aceites é-lhes devolvido uma chave simétrica usada para proteger o novo canal estabelecido. Todas as mensagens trocadas nesta fase são cifradas com a chave global que deixa de ser necessária para proteger o canal após a introdução da nova chave partilhada entre ambos. Após a organização estar completa pode iniciar-se a fase de operação onde um nó envia os relatórios para o respectivo *cluster head* cifrado com a chave partilhada com este. A especificação prevê ainda mecanismos de adopção, em caso de um *cluster head* morrer e adição de nós à rede visto que não usa um sistema de rondas como o protocolo Leach e, portanto, não beneficia de uma reorganização topológica periódica e como tal os órfãos e os novos nós terão de seguir o protocolo se quiserem entrar (ou reentrar) na rede.

2.3.3.5. Framework de Bohge

Bohge e Trappe [11] propuseram uma *framework* de autenticação para redes de sensores hierárquicas onde os participantes se distribuem por níveis conforme os seus recursos computacionais. Os mecanismos de autenticação são também diferentes entre os níveis de

sensores e os que têm maiores capacidades podem usar então criptografia assimétrica. São propostos 2 níveis hierárquicos por onde se distribuem 4 classes de participantes. No primeiro nível, o mais fraco, encontram-se os nós sensores (D) que estão limitados ao uso de criptografia simétrica. Já no segundo nível encontram-se as *base stations* (no caso de haverem várias) (B) e uma nova classe de nós, os nós encaminhadores (C) que fazem apenas encaminhamento de dados dos sensores para estas últimas. A *framework* prevê também a separação entre a *base station* e a aplicação (A), estando as duas ligadas por um meio físico externo à rede de sensores. A arquitectura pode ser vista na figura 2.2.

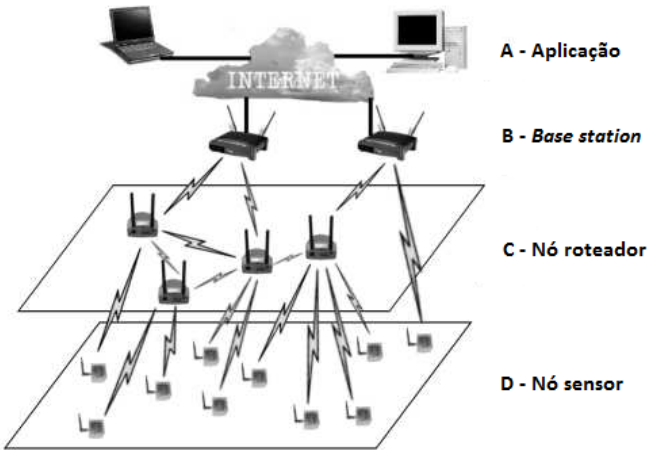

Figura 2.2 - Hierarquia da framework de Bohge

A principal motivação é garantir autenticidade e integridade entre os níveis. A *framework* especifica como autenticar novos nós, como estabelecer segredos entre eles e com a aplicação, como reagir perante mudanças topológicas e como garantir autenticidade de dados dos sensores.

A autenticação é garantida com base no *pré-deployment* de certificados pelos participantes da rede tendo em conta as capacidades dos mesmos. Estes certificados são gerados por um TTP que pode ser qualquer nó em qual todos confiam e que seja capaz de executar assinaturas RSA. Quando um nó entra então na rede terá de apresentar o seu certificado à aplicação e se o mesmo for validado a mesma distribui novas chaves simétricas para fornecer garantias de autenticidade perante ela durante a operação da rede. A aplicação é também responsável por gerar chaves partilhadas entre nós sensores a as *base stations*.

Após o estabelecimento das chaves e ligações seguras necessárias, as mesmas são usadas para garantir autenticação no envio de dados para a aplicação através de *macs*. Nesta fase a troca de mensagens processa-se da mesma forma da fase de organização: os nós sensores enviam mensagens para a *base station* e os nós encaminhadores tratam de providenciar o encaminhamento necessário.

2.3.4. Síntese e análise crítica

A problemática da distribuição de chaves em redes de sensores concluiu que é necessário usar métodos de pré-distribuição baseados em criptografia simétrica devido às limitações dos mesmos. No entanto, em redes hierárquicas, podem haver sensores com diferentes recursos e esse facto pode ser explorado para atribuir tarefas mais pesadas àqueles que mais recursos possuem. Os protocolos propostos por Bohge el al. [11] e Oliveira et al. [10] defendem então essa vertente, com este último a propor um esquema que se adapta a um nível arbitrário de níveis hierárquicos (fixo no esquema de Bohge) e que depende apenas de criptografia simétrica. Mais recentemente foram propostos dois esquemas [8], [9], baseados no protocolo Leach que visam uma rede totalmente homogénea onde todos os sensores são iguais e são candidatos a desempenhar as mesmas funções. Nestes interessa destacar o SecLeach que se baseia no esquema básico de distribuição probabilística de chaves de Eschenauer e Gligor para garantir autenticidade dos nós perante os *cluster heads*.

Apesar do protocolo SecLeach ser o único a implementar um dos esquemas básicos de pré-distribuição de chaves estudado, é possível argumentar sobre o seu possível uso nos restantes protocolos, mais concretamente no LHA-SP visto que a implementação de tal mecanismo na *framework* de Bohge alteraria profundamente o protocolo e no F-Leach faria com que ficasse bastante semelhante ao próprio SecLeach. Assim, analisando o protocolo LHA-SP, se em vez de usada uma chave global para formação da rede fosse usado um mecanismo de pré-distribuição de aleatória de chaves, então a segurança face à captura de sensores sairia reforçada devido à fragilidade que uma chave global representa. No entanto a cobertura da rede seria potencialmente menor devido à possível inexistência de *cluster heads* que partilhem chaves na proximidade de um nó, criando aqui um *tradeoff* entre segurança e cobertura da rede. Como argumento adicional pode, e deve ser referido que o uso de um esquema que aproveita o conhecimento da localização traria benefícios à cobertura, como aliás foi essa a conclusão tirada com o estudo dos mesmos.

Finalmente, para uma análise crítica sobre os protocolos estudados, interessa apresentar uma análise dos mesmos face a alguns indicadores que vão ser obtidos no capítulo de testes.

Cobertura efectiva da rede

Todos os protocolos estão sujeitos às condições de operação reais de uma rede de sensores que afectam obviamente a cobertura, ou seja, estão sujeitos ao modelo de comunicação e colisões rádio da pilha IEEE 802.15.4. Adicionalmente existem também factores, muitas vezes aleatórios, especificados pelos próprios protocolos que acabam por se reflectir na cobertura efectiva da rede.

Tanto no protocolo F-Leach como no SecLeach a cobertura é directamente afectada pela probabilidade de um nó se tornar *cluster head* e ainda por critérios probabilísticos da distribuição destes pela rede. Adicionalmente, neste último, é preciso ter também em conta as restrições de vizinhança impostas pelo esquema de pré-distribuição aleatória de chaves.

O protocolo LHA-SP tem uma pré-distribuição determinística de chaves portanto a cobertura será afectada apenas pela distribuição topológica dos nós, ou seja, um nó de nível *h* terá de ter pelo menos um vizinho de nível *h+1* para formar *cluster*. Finalmente, a cobertura na *framework* de Bohge é apenas directamente afectada pelo facto de os nós sensores terem pelo menos um nó encaminhador como vizinho e este conseguir encaminhar os dados até à *base station* segundo um algoritmo de encaminhamento não especificado pela *framework.*

Condições de sobrecarga de comunicação e impacto energético

Começando pelos protocolos de Bohge e LHA-SP, que assumem uma hierarquia heterogénea, o problema do custo das operações é endereçado de maneira a que os nós de nível superior executem as operações mais pesadas. Já nos protocolos SecLeach e F-Leach, a não existência de nós com mais recursos é resolvida com um mecanismo periódico de troca de *cluster heads* que permite distribuir a carga por todos.

Relativamente às comunicações, os protocolos F-Leach e SecLeach usam agregação de dados nos *cluster heads* para minimizar o número de transmissões necessárias, mas por outro lado "consomem" muitas mensagens devido à reorganização periódica da rede para formar novos *clusters*. Já o protocolo de Bohge não usufrui de tal mecanismo (agregação) devido a fornecer segurança *end-to-end*, mas não necessita de constante reorganização da rede. Por fim o protocolo LHA-SP também não necessita de várias reorganizações e parece levar vantagem no número de

mensagens trocadas durante a fase de organização e autenticação de nós, mas isso também se reflecte num *tradeoff* com segurança devido a inicialmente usar uma chave global.

Condições de segurança e de resiliência face à captura de nós

Uma primeira aproximação ao estudo da segurança pode passar por uma comparação entre protocolos relativamente à resistência face à topologia de ataques do modelo de adversário:

	Ataques externos	**Eavesdropping aos dados**	**Sinkhole / Wormhole**	**Sybil**	**Blackhole**	**Dados falsos**
Leach	Não resiste	Não resiste	Não resiste	Não resiste	Não resiste	Não resiste
F-Leach	Resiste	Não resiste	Forte	Média	Média	Fraca
SecLeach	Fraca	Não resiste	Forte	Média	Média	Resiste
LHA-SP	Resiste	Resiste	Fraca	Fraca	Fraca	Resiste
Bohge	Resiste	Não Resiste	---	---	---	Resiste

Tabela 2.2 - Tabela comparativa da resistência dos protocolos face a ataques

Em primeiro lugar a *framework* de Bohge é um protocolo um pouco diferente dos outros no que toca à organização em *clustering*. A razão prende-se no facto de que é o protocolo de encaminhamento dos nós encaminhadores que determina a topologia da rede e a segurança da mesma face a um atacante interno com as hipóteses vistas no capítulo do modelo de adversário.

Nos restantes protocolos a segurança prende-se sobretudo nos *cluster heads*, que são o principal alvo dos atacantes. A razão é que se um *cluster head* for comprometido ou se um atacante for eleito, então passa a controlar o destino de várias mensagens, neste caso todas as enviadas pelos nós membros que se juntaram a ele. De todos os protocolos, apenas o SecLeach permite que um atacante externo se torne *cluster head* isto porque a autenticação é apenas garantida entre um nó membro perante um *cluster head* e não vice-versa. Todos os restantes garantem que apenas nós legítimos se tornem *cluster heads*.

Relativamente aos dados dos sensores, estes só são protegidos de ataques de *eavesdropping* no protocolo LHA-SP pois é o único que prevê cifrar os dados durante as várias fases. Já em ataques onde o atacante envia dados falsos ou apenas *spam*, só o protocolo F-Leach parece vulnerável visto que, ao contrário do SecLeach, este não implementa qualquer tipo de autenticação entre os nós sensores e os *cluster heads*.

Como contribuição desta dissertação, no capítulo de testes foram obtidos indicadores de cobertura e fiabilidade dos protocolos face a ataques *sinkhole* e *blackhole*.

3. Modelação e implementação da plataforma de simulação

Neste capítulo descreve-se a arquitectura da plataforma de simulação final e as suas funcionalidades. A arquitectura base é a mesma arquitectura do simulador jProwler [14] que foi escolhido por apresentar melhores características e funcionalidades para a prototipagem e análise de protocolos de distribuição e estabelecimento de chaves em redes de sensores sem fios de grande escala.

Sobre o núcleo base do jProwler foram implementados os módulos necessários à avaliação dos protocolos, assim como algumas novas funcionalidades que serão descritas neste capítulo. Inicialmente será feita uma descrição do núcleo de simulação inicial do simulador e em seguida será apresentada a modelação da arquitectura final assim como a descrição de cada um dos módulos adicionais.

3.1. Ambiente de Simulação

Para a implementação e validação dos protocolos utilizou-se o simulador jProwler desenvolvido em Java. Este é um simulador baseado em eventos discretos que simula a natureza não-determinística dos canais de comunicação e os protocolos de baixo nível que fazem parte dos sensores baseados em TinyOS e comunicações por rádio segundo a norma IEEE 802.15.4, tendo por base as características gerais de processamento de sensores usuais (do tipo Mica Motes da Crossbow). O simulador forma uma base extensível que permite a adição de módulos adicionais, onde se realçam os módulos de aplicação usados para a implementação dos protocolos de distribuição e estabelecimento de chaves.

Em seguida será feita uma descrição dos mecanismos essenciais que compõem o núcleo básico de simulação que compõem o jProwler e que serão essenciais para a simulação não-determinística dos canais de comunicação.

3.1.1. Modelo de Rádio

O modelo de rádio determina a força de um sinal ao ser transmitido pela rede em qualquer ponto no espaço e para qualquer nó sensor. Com base nesta informação as condições de recepção dos sensores podem ser avaliadas e colisões podem ser detectadas. O jProwler oferece dois modelos de rádio mas aquele considerado para a implementação e análise dos protocolos foi o modelo gaussiano que assume os nós como maioritariamente estáticos e que não mudam de posição ao longo do tempo, ou seja, está de acordo com as especificações dos protocolos.

O modelo gaussiano inicialmente assume que um sensor tem uma força de sinal máxima e com isso calcula o *static fading*[5] (baseado na distância e um factor aleatório) para cada nó da rede, determinando os vizinhos que potencialmente receberão as mensagens enviadas por este. Já a força de sinal entre transmissor e receptor no envio de mensagens é calculada dinamicamente com base nas leituras estáticas iniciais, um factor dinâmico aleatório e a força de sinal do emissor que não deverá exceder o valor máximo inicial. Sabendo a força de sinal com que recebe transmissões de cada vizinho, um nó simulado manterá uma soma das forças dos vizinhos a transmitir num determinado instante (ruído) e determinará se a mensagem que está a receber será corrompida, isto se estiver efectivamente a receber alguma mensagem. Se esse ruído não exceder um certo limite então diz-se também que o canal de comunicação de um nó está em *idle* e pronto a transmitir dados. Todas as constantes como o limite encontram-se já parametrizadas segundo as propriedades dos sensores Mica 2 Mote.

3.1.2. Modelo de camada MAC

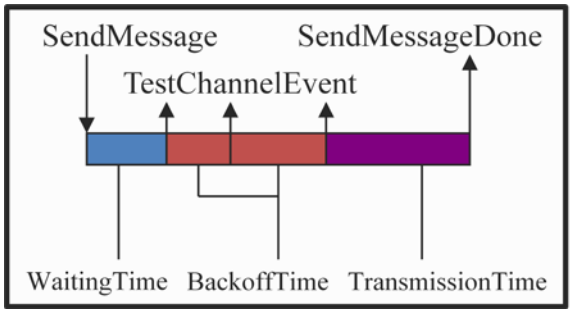

Figura 3.1 - Especificação do modelo de camada MAC

[5] Ao longo deste relatório utiliza-se o termos "*static fading*" em língua inglesa mas que poderia ser traduzido por "desgaste estático". O termos em língua inglesa é no entanto utilizado por facilidade de interpretação conceptual face à literatura da área.

O modelo de camada MAC presente no jProwler pode ser descrito pela figura 3.1. Quando a aplicação executa *SendMessage* a camada MAC gera um evento de espera *WaitingTime* e após este terminar procede então à verificação do estado do canal de comunicação. São gerados diversos eventos sequenciais até que o canal seja detectado como estando em modo *idle*. Entre cada evento existirá um tempo de espera *BackoffTime*.

Quando o canal estiver finalmente em *idle* a emissão inicia-se e, após um evento com tempo *TransmissionTime* fixo mais um valor aleatório, a aplicação recebe ordem de execução de *SendMessageDone* sinalizando o fim do envio da respectiva mensagem. Do lado do receptor a camada MAC irá verificar a integridade da mensagem, ou seja, se não foi corrompida devido ao ruído gerado por nós vizinhos. Se a mensagem não estiver corrupta será entregue à aplicação do receptor através dum evento *ReceiveMessage*. Por fim resta referir que os tempos *WaitingTime* e *BackoffTime* são variáveis aleatórias uniformemente distribuídas.

3.2. Extensão da Plataforma

3.2.1. Marcação de eventos e mensagens

Foi implementado um sistema de marcações (*flags*) de eventos e mensagens que permitem ter um maior controlo sobre os diversos tipos de eventos que são processados pelo simulador. No caso da marcação de mensagens foi ainda criado um mecanismo de propagação das *flags* das mesmas para os eventos correspondentes e para todos os restantes eventos gerados em consequência do primeiro. Se virmos, por exemplo, os eventos que estão associados ao envio de uma mensagem, estes herdam as *flags* da mesma, propagando uma determinada característica, representada por essa *flag*, por todos os eventos gerados posteriormente (TestChannelEvent, TransmissionTime, etc..). Assim, esta funcionalidade permitirá à implementação dos protocolos, assim como ao simulador e módulos adicionais, especificar o tipo de eventos que estão por processar na fila de eventos ou já foram processados.

No geral esta é uma importante funcionalidade que será essencial para a implementação de alguns dos módulos adicionais para a extensão da plataforma face aos objectivos pretendidos.

3.2.2. Subscrição de eventos

Em consequência da funcionalidade de marcação de eventos e mensagens anteriormente descrita, foi implementado um mecanismo de subscrição de eventos, identificados pelas *flags* que estes possuem. Esta subscrição pode ser feita quer à adição de eventos, quer à execução dos mesmos. Em geral, este mecanismo, juntamente com a funcionalidade de marcação é útil no sentido em que permite uma fácil extensão da plataforma em diversos módulos, os quais poderão introduzir novas *flags* nos eventos da rede (gerados por si, por outros módulos ou pelo simulador) e ainda subscreverem-se perante os mesmos.

3.2.3. Módulos adicionais

Apesar de o jProwler possuir já boas condições de simulação de uma RSSF, foi ainda necessário acrescentar alguns módulos para permitir a correcta implementação e validação dos protocolos tendo em conta os objectivos definidos. Dito isto os seguintes módulos foram desenvolvidos e adicionados à plataforma inicial:

MTL	Módulo de teste de latência
MACR	Módulo de análise de cobertura da rede
MAF	Módulo de análise de fiabilidade
ME	Módulo de teste de consumo energético
MIA	Módulo de injecção de ataques
MDP	Módulo de controlo de definições e parametrizações
MVCR	Módulo de visualização e controlo da rede

3.2.3.1. Módulo de teste de latência

Para medir o tempo de instalação de chaves nos vários nós participantes da rede e, numa visão mais geral, medir o tempo de organização da mesma foi necessário adicionar um módulo de teste de latência à arquitectura base do simulador. Este módulo baseia-se na premissa de que a rede fica estável/organizada a partir do momento em que não transitem mais mensagens de descoberta de chaves, estabelecimento de ligações seguras, atribuições de chaves e outros tipos de

mensagens que envolvam uma alteração da formação actual da rede. Assim, quando a rede atinge tal estado, poder-se-á dizer que a rede estabilizou e o módulo será capaz de calcular a variação de tempo entre o início da sua formação e o actual instante.

Modelação

A modelação deste módulo baseou-se no mecanismo de marcação de mensagens e subscrição de eventos para determinar o instante em que a rede estabiliza. Para tal os eventos e as mensagens, específicas a cada protocolo, de formação da rede devem ser marcados com a *flag* de *SETUP* (tarefa a cargo da implementação do protocolo). O módulo determina então o instante em que a rede estabiliza quando a fila de eventos do simulador deixa de ter eventos marcados com *SETUP*. Este processo pode ser visto em mais detalhe na figura 3.2.

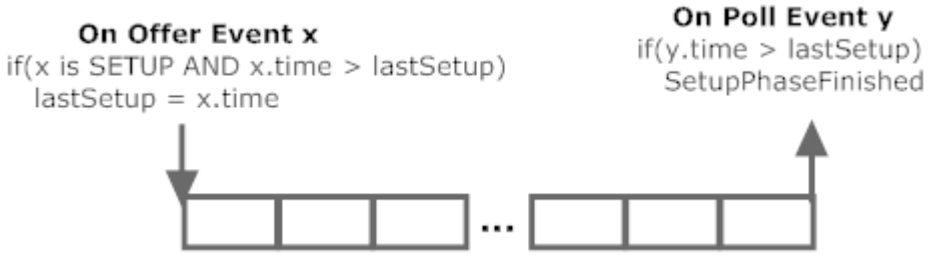

Figura 3.2 - Especificação do módulo de teste de latência

Inicialmente o módulo subscreve a adição de eventos marcados como *SETUP*. Estes eventos podem ter essa *flag* devido a terem sido explicitamente declarados como tal ou devido a terem surgido em consequência de uma mensagem assim marcada. Assim, sempre que for adicionado um evento de formação da rede à fila de eventos do simulador, o módulo de teste de latência será informado e manterá o tempo do último evento de *SETUP* conhecido. Além da subscrição à adição, o módulo subscreve também a execução de todos os eventos, verificando para cada um o seu tempo e, em caso de ser superior ao último tempo dum evento de *SETUP* conhecido, então estamos perante o fim da formação da rede, não existindo mais mensagens de formação a serem enviadas/recebidas nem eventos específicos do protocolo que alterem a formação actual. Após atingir esse estado a latência é calculada com base no instante inicial de formação da rede e o instante em que foi executado o último evento marcado como *SETUP*.

3.2.3.2. Módulo de análise de cobertura da rede

Face aos objectivos propostos foi necessário especificar e implementar um mecanismo que permita analisar a cobertura da rede pois trata-se de um importante indicador para a análise da eficácia e escalabilidade de um protocolo a operar em condições próximas do real.

Em última análise um sensor está coberto se conseguir enviar um relatório para a b*ase station*, mas com a implementação desde módulo decidiu-se ir um pouco mais longe e dividiu-se o conceito de cobertura em vários tipos distintos, permitindo um maior controlo sobre a análise dos resultados obtidos. Assim foram criados dois grandes grupos de cobertura que serão em seguida apresentados. As condições de cobertura de cada tipo acabam sempre por ser específicas a cada protocolo mas será por agora apresentada uma definição geral adequada a qualquer um deles.

Análise estática de cobertura

Neste grupo o módulo realiza um estudo mais teórico sobre a cobertura da rede, ou seja, não tendo em consideração todas aquelas características que possam causar perdas de mensagens e posteriores perdas de cobertura inerentes à operação real subjacente à pilha IEEE 802.15.4. Foram assim identificados e implementados 3 tipos de cobertura:

- *Cobertura Física*: Diz-se que um sensor está coberto fisicamente se possuir um outro sensor como vizinho. O objectivo aqui será identificar possíveis sensores isolados do resto da rede ou, de um ponto de vista de estudo teórico de grafos, identificar nós que tornem a rede num grafo desconexo.
- Cobertura Topológica: Devido à especificação de cada protocolo um sensor apenas se pode ligar a um conjunto bem definido de nós de entre os vizinhos e é este factor que este tipo de cobertura pretende realçar. A título de exemplo um nó do protocolo SecLeach só se pode ligar e enviar relatórios para outro nó vizinho que esteja a desempenhar o papel de *cluster head* na ronda actual.
- *Cobertura do Protocolo*: Ter um ou mais nós topologicamente válidos como vizinhos não é suficiente para garantir que um relatório enviado por um nó alguma vez chegue à *base*

station. É exactamente estas situações que este tipo de análise pretende realçar. Assumindo como exemplo um protocolo de encaminhamento que funcione por *flooding*, não basta um nó ter vizinhos para onde enviar dados, é também necessário existir um caminho para a *base station*.

Análise dinâmica de cobertura

Neste grupo o módulo analisa a cobertura efectiva da rede após a execução do protocolo sobre a pilha IEEE 802.15.4. Os conceitos são os mesmos da análise estática mas aqui a cobertura já irá ser afectada pelas características da operação real em redes de sensores. Assim o módulo analisa 2 tipos de cobertura:

- *Cobertura Parcial*: Um sensor estará parcialmente coberto se, após terminar a organização da rede, estiver ligado a algum vizinho, o qual possa dar seguimento ao envio dos relatórios deste.
- *Cobertura Total*: Se, após a organização da rede, um sensor estiver parcialmente coberto e a organização obtida da rede permitir um caminho até à *base station* seguindo a especificação do protocolo então um nó estará totalmente coberto.

Modelação

Podemos ver a arquitectura deste módulo como um controlador que testará, sensor a sensor, os diversos tipos de cobertura seguindo ao detalhe a especificação do protocolo a operar na rede. A figura 3.3 demonstra exactamente essa arquitectura. As condições de cobertura de cada sensor na análise dinâmica são específicas a cada protocolo e como tal serão detalhadas no seu respectivo capítulo de implementação.

Outro aspecto importante é que apenas é interessante analisar a cobertura da rede após a mesma estar estável, ou seja, após a execução total do protocolo de formação da rede. Para tal, este módulo segue exactamente a mesma especificação do módulo de teste de latência para a detecção da estabilização da rede (subscrição à adição de eventos *SETUP* e subscrição à

execução de todos os eventos). No entanto nada proíbe que a análise de cobertura seja feita antes, ou depois, da rede estabilizar pois o módulo fornece primitivas de análise de cobertura ao simulador.

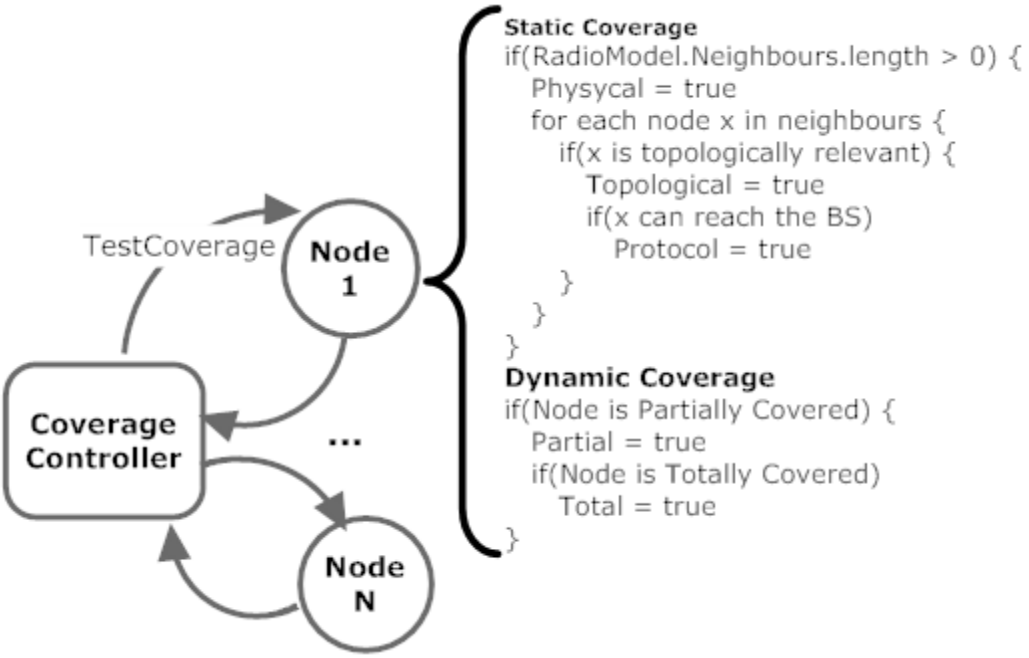

Figura 3.3 - Especificação do módulo de análise de cobertura

3.2.3.3. Módulo de análise de fiabilidade

A taxa de fiabilidade de entrega de mensagens de uma RSSF indica qual é a probabilidade de entrega de um qualquer relatório, sendo por isso um fenómeno interessante de observar, não só para validar o protocolo, como para estabelecer uma base comparativa em futuros testes que meçam essa mesma fiabilidade quando a rede se encontra sob ataque. Assim, face aos objectivos propostos, foi necessário criar um novo módulo que permita testar as condições de fiabilidade de um protocolo. Este módulo mede o impacto de um protocolo na entrega de relatórios quando desencadeado concorrentemente em vários sensores e através de um processo de disseminação ou difusão dos mesmos através da rede.

É sabido que o parâmetro que mais influencia a fiabilidade da rede está relacionado com o débito de mensagens, pois quantas mais mensagens forem geradas maior será a probabilidade de existirem colisões no nível físico e é nessa premissa que este módulo se baseia, dando a

parametrizar o número de sensores envolvidos no envio de relatórios simultâneos e o número de relatórios que serão enviados por cada. Outro factor importante é que este módulo apenas abrange os sensores cobertos após a formação da rede pois os restantes terão sempre à partida uma fiabilidade de 0% na entrega de relatórios.

Modelação

Inicialmente o controlador de fiabilidade obtém uma percentagem, parametrizável pelo utilizador, de sensores cobertos após a formação da rede e para cada um deles gera um determinado número, também parametrizável, de eventos os quais serão convertidos em relatórios e enviados para a *base station* seguindo a especificação do protocolo.

A modelação deste módulo, à semelhança do módulo de teste de latência, baseou-se também no mecanismo de marcação de mensagens e eventos para propagar o teste de fiabilidade por todos os eventos de envio e recepção de mensagens gerados em consequência da disseminação dos relatórios. Como tal, todas as mensagens e eventos gerados para um teste de fiabilidade recebem a *flag RELIABILITY* e o controlador poderá então aperceber-se do início e fim, em termos de fila de eventos do simulador, do teste de fiabilidade. Para tal é criada uma subscrição à adição de eventos do tipo *RELIABILITY* na fila de eventos e uma subscrição à execução de qualquer evento.

Após o teste estar concluído o controlador vai obter um relatório detalhado, gerado na b*ase station*, contendo a informação necessária ao cálculo da fiabilidade em termos percentuais e outros dados importantes como a latência média na recepção dos relatórios, número médio de *hops*, entre outros. Na figura 3.4 pode-se ver uma representação da modelação descrita.

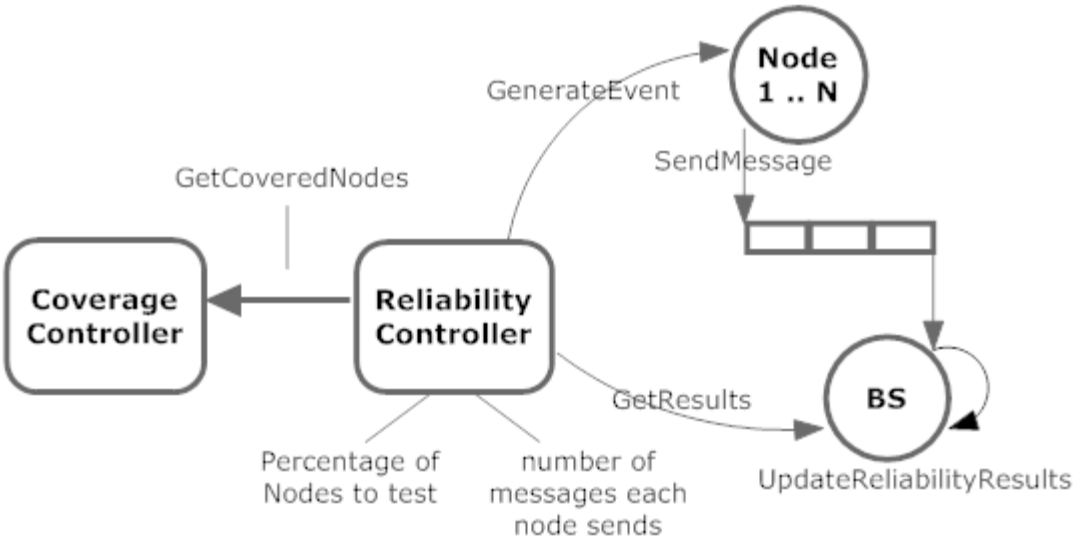

Figura 3.4 - Especificação do módulo de análise de fiabilidade

3.2.3.4. Módulo de teste de consumo energético

Para a validação dos protocolos propostos e para a extensão da plataforma foi necessária a adição de um módulo de teste de consumo energético à mesma dada a sua inexistência na presente versão. Um dos mais importantes factores numa rede de sensores é exactamente a baixa capacidade energética dos seus participantes, daí a importância de um módulo que consiga simular o consumo dos mesmos durante o seu tempo de vida em operação na rede.

O modelo de consumo energético tem por base a simulação do comportamento do TinyOS, onde os sensores podem estar em 3 diferentes estados: activo, *idle* ou adormecido. Todos os estados pressupõem gastos energéticos ao longo do tempo diferenciados que dependem, essencialmente, dos dispositivos do sensor que se encontram ligados. Assim, no estado adormecido o sensor não tem qualquer dispositivo ligado, no estado *idle* o sensor não tem o dispositivo de comunicação por rádio ligado e no estado activo todos os dispositivos encontram-se ligados. Contudo, com o desenvolvimento dos protocolos propostos, os nós necessitam de se encontrar em estado activo, pois a natureza não determinista da recepção e envio de mensagens dos mesmos assim o dita. A título de exemplo um nó de nível *h+1* no protocolo LHA-SP pode receber a qualquer momento dados de um nó de nível *h*. Se o mesmo estiver em modo adormecido ou *idle* tal implicaria uma perda dos dados.

Foram também contemplados vários indicadores energéticos que afectam o consumo de energia dos sensores quando executam várias tarefas. Nomeadamente o envio e recepção de mensagens, cifra e decifra de dados e criação de *macs* e assinaturas.

Os indicadores energéticos utilizados na adição do módulo de consumo energético, apresentados na tabela 5.1, foram estudados e analisados em [29], [30], [31] e [32].

Acção	Custo energético
Capacidade Energética	9360 J
Transmissão	59,2 μJ/byte
Recepção	28,6 μJ/byte
Cifra/Decifra (RC5/Skipjack)	1,241 / 1,788 μJ/byte
Assinatura (RSA)	304 mJ/byte
Verificação de assinatura (RSA)	11,9 mJ/byte
Criação de MAC (RC5/Skipjack)	1,721 / 2,474 μJ/byte
Verificação de MAC (RC5/Skipjack)	1,721 / 2,474 μJ/byte
Mudar de estado	2,86 μJ
Estado Activo	0,096 μJ/s
Estado *Idle*	0,0096 μJ/s
Estado *Sleep*	0,00033 μJ/s

Tabela 3.1 - Indicadores do módulo de teste de consumo energético

Modelação

Ao utilizador foi dada a possibilidade de modelar a aplicação com zonas específicas do código (*EnergyConsumptionAction*) onde são executadas operações que estão sujeitas ao modelo de consumo energético atrás descrito. Cabe também ao utilizador aplicar a dimensão da operação e as *flags* associadas à acção para posterior descriminação de consumos energéticos, como por exemplo o custo energético associado ao encaminhamento de dados detalhado no capítulo 4.2.

Para cada acção de consumo energético o módulo será depois capaz de aplicar o custo segundo o modelo e fará em seguida uma actualização de um controlador global de consumos associado a cada simulação. Esse controlador, além de fornecer o consumo global da rede,

fornece também os consumos energéticos discriminados por nó, classe de nó (nó sensor, *base station*, ...), tipo de evento (cifra, transmissão, ...) e tipo de *flag*.

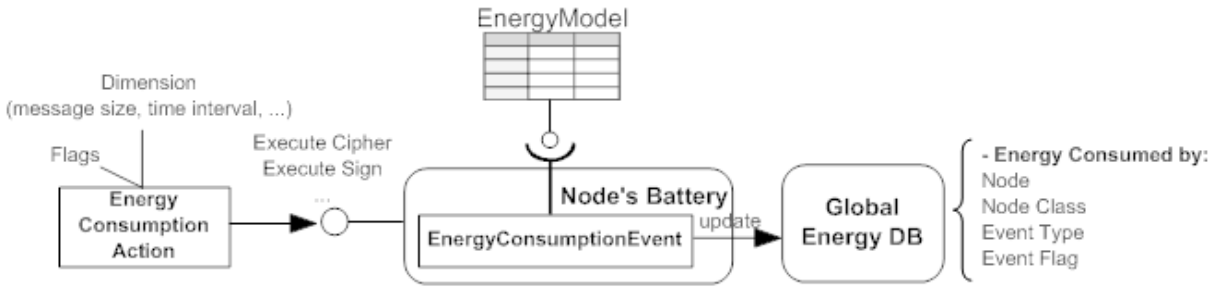

Figura 3.5 - Especificação do módulo de teste de consumo energético

3.2.3.5. Módulo de injecção de ataques

De modo a analisar o comportamento do protocolo a operar na rede face a um cenário sujeito a ataques foi necessário definir e implementar um módulo que permita a injecção de diversas tipologias de ataques na rede. O módulo implementado permite ao utilizador simular ataques por captura (internos) que sigam as hipóteses que foram apresentadas no capítulo do trabalho relacionado. O módulo também tem em conta que um atacante poderá atacar em qualquer fase do protocolo a ser simulado e dá ao utilizador a liberdade para implementar o ataque durante qualquer uma delas.

Adicionalmente foram também criadas algumas classes de ataques que servirão de base comparativa para o impacto sobre os protocolos no capítulo de testes. Mais concretamente foram implementados ataques do tipo *Blackhole* e *Sinkhole* para cada protocolo proposto.

Modelação

Para modelar um ataque interno foi definida uma nova classe de nó na rede, o nó atacante. O nó atacante é um nó que inicialmente se enquadra no tipo de nó participante, mas que em determinada fase do protocolo sofre uma intrusão por parte de um adversário. A esse nó será atribuída uma aplicação especial, considerada comprometida, que se irá comportar de acordo

com o tipo de ataque definido nas parametrizações do módulo de injecção de ataques e conforme a especificação dada pelo utilizador, actuando como alternativa a uma aplicação legítima.

Em termos concretos, foi criado um controlador de ataques que fornece primitivas ao utilizador para escolher um dos ataques especificados e para alterar as configurações do mesmo, quando aplicáveis. Posteriormente, na fase de *pré-deployment*, onde os nós são criados, o simulador utilizará as configurações do controlador, parametrizadas pelo utilizador, para determinar um certo conjunto de nós aleatórios que se tornarão em nós atacantes e neles será injectada a aplicação correspondente ao ataque e que deverá ter sido especificada pelo utilizador e mapeada para o mesmo. Esta especificação pode ser vista na figura 3.6.

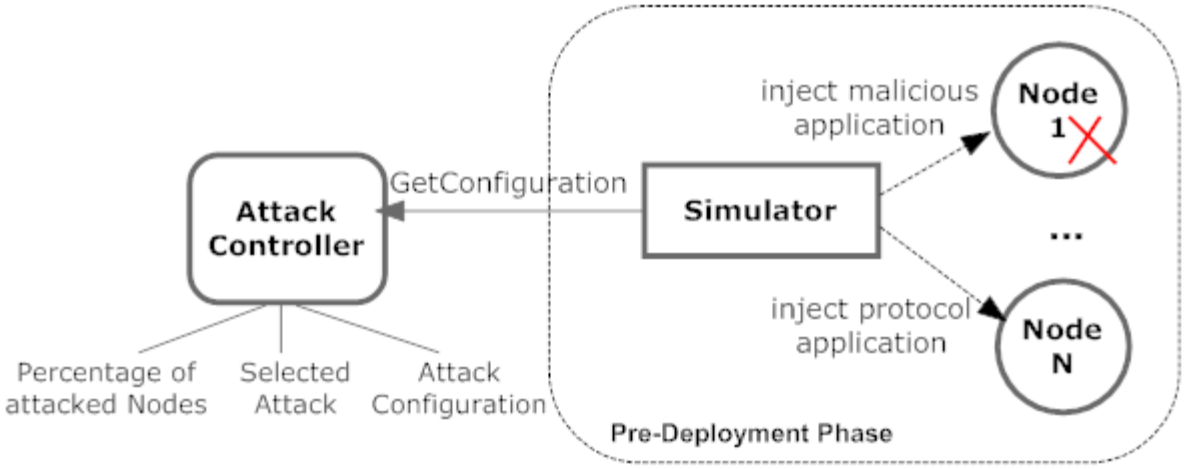

Figura 3.6 - Especificação do módulo de injecção de ataques

3.2.3.6. Módulo de controlo de definições e parametrizações

Para fornecer uma boa base para a validação de protocolos e realização de vários testes foi implementado um módulo que possibilita o controlo das várias definições e parametrizações dos mesmos, assim como do ambiente de simulação em geral, como parametrizações dos comportamentos dos nós, da simulação e do próprio ambiente gráfico onde a simulação é apresentada. Estas parametrizações reflectem-se posteriormente nos resultados da presente simulação. Na figura 3.7 pode-se observar uma amostra da interface gráfica que permite definir então essas parametrizações.

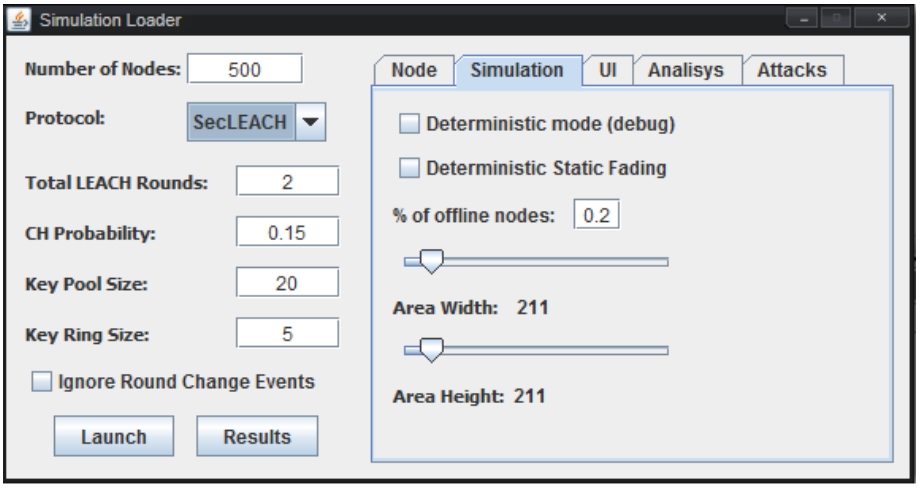

Figura 3.7 - Interface do módulo de controlo de definições e parametrizações

Passa-se agora a destacar algumas das mais relevantes parametrizações. Algumas destas correspondem mesmo a novas funcionalidades introduzidas na plataforma que cumpre a contribuição de disponibilização de uma plataforma genérica de avaliação de protocolos.

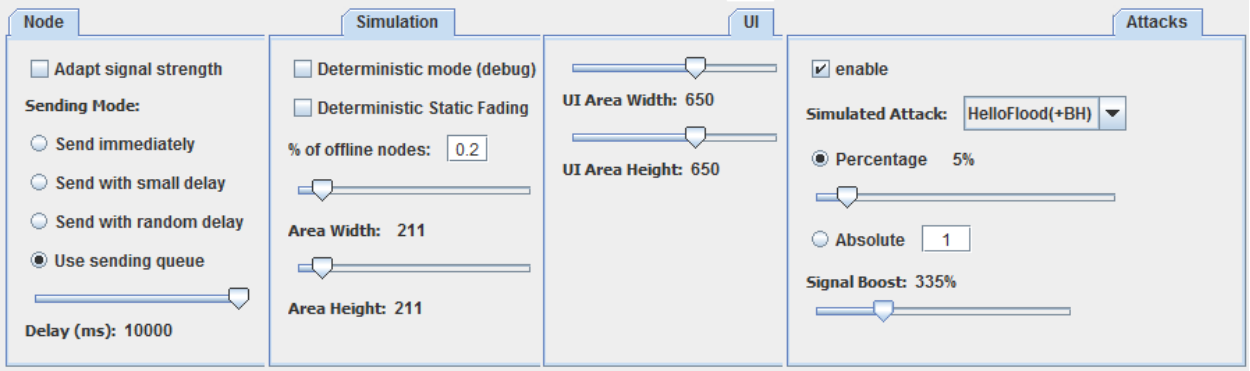

Figura 3.8 - Opções do módulo de controlo de definições e parametrizações

Adapt Signal Strength: Alguns protocolos podem tomar partido de uma adaptação da força de sinal na emissão de mensagens, permitindo uma emissão com menor sinal e consequentemente um gasto menor de energia. O protocolo Leach é um bom exemplo para esta parametrização pois um nó sensor pode reduzir a força de sinal na resposta enviada a um *cluster head* conforme a força de sinal com que recebeu o anúncio do mesmo.

Sending Mode: A camada MAC do simulador jProwler simula o protocolo CSMA dos sensores Berkeley e, como tal, as colisões de mensagens acabam por ser frequentes entre sensores vizinhos que decidem transmitir para o ar, separados apenas por um curto espaço de tempo. Esta foi a motivação para a implementação de vários métodos de envio de mensagens, possibilitando um maior espaçamento entre estas. A implementação foi feita ao nível da aplicação e no geral existem três grandes possibilidades:

- Envio imediato: A mensagem é imediatamente entregue à camada MAC a qual seguirá com a execução do protocolo CSMA, com um tempo aleatório de espera, verificação do estado do canal e tempo aleatório de *backoff*;
- Envio com atraso aleatório: Neste caso a aplicação aplica o seu próprio atraso, aleatório conforme um valor máximo parametrizável, antes de entregar a mensagem à camada MAC;
- Envio com recurso a fila de mensagens: Neste caso as mensagens são inseridas numa fila que a aplicação gere e existe um evento contínuo que vai despachando as mesmas, por ordem de adição na fila, com intervalos de tempo variáveis até um máximo parametrizável. A grande diferença para o método anterior é que desta maneira é possível garantir um débito máximo de envio de mensagens por uma certa unidade de tempo e, mais importante, é possível garantir a ordem de envio das mesmas, o que não é garantido ao aplicar um atraso aleatório a cada mensagem individualmente.

Deterministic Mode: O simulador jProwler não possui, por defeito, um mecanismo determinístico de envio de mensagens e, como tal, foi decidido implementar um pois as vantagens são muitas para o desenvolvimento e estudo de protocolos em RSSF, especialmente para efeitos de *debug* dos protocolos. O mecanismo implementado substitui a camada MAC de um nó e simula igualmente o tempo de espera até transmitir, assim como o tempo de transmissão. A grande diferença está na ausência da verificação do estado do canal, pois é considerado como estando sempre em *idle* e, consequentemente, no ruído gerado na sua vizinhança ao transmitir, que é considerado nulo evitando então colisões. A figura 3.9 demonstra o funcionamento deste modo (ver também figura do modelo de camada MAC).

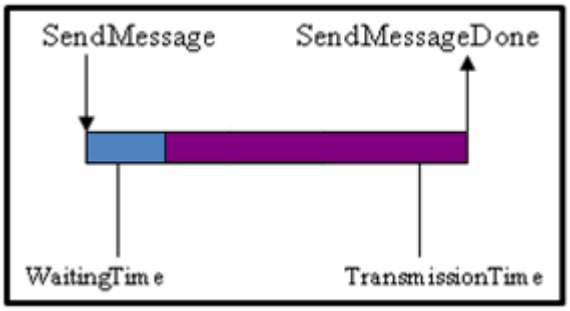

Figura 3.9 - Especificação do modelo de envio determinístico de mensagens

Além do mecanismo determinístico de envio de mensagens foi também implementado um esquema determinístico de *static fading*. O objectivo é garantir a vizualização mútua entre dois nós vizinhos, ou seja, se um vê o outro segundo o modelo de rádio (cálculo com base na distância e um factor aleatório), então o outro também vai ver este e ambos serão vizinhos um do outro para todos os efeitos. No final, se uma simulação tomar partido destes dois mecanismos individualmente, podemos analisar o impacto na rede tanto de colisões de mensagens por vários nós estarem a transmitir ao mesmo tempo, como o impacto das propriedades da natureza não determinística do modelo de rádio. Num caso extremo, de os dois mecanismos estarem a ser usados numa simulação (útil para *debug*), pode-se observar resultados idênticos a um estudo apenas teórico de um protocolo, ou seja, o módulo de cobertura irá dar os mesmos valores para a cobertura estática e dinâmica e o módulo de análise de fiabilidade devolverá sempre um valor de 100% nos testes efectuados.

Adição de nós *offline*: Uma das extensões à plataforma inicial passa por um mecanismo de adição de nós à rede que depois seguem a especificação do protocolo (se houver) para se juntarem efectivamente à mesma. Para uma implementação eficiente os nós que serão adicionados no futuro terão de ser, na realidade, adicionados ao mesmo tempo que os nós iniciais e a quantidade máxima destes deverá ser especificada pelo utilizador. A razão é que a operação de cálculo dos vizinhos no modelo de rádio é muito dispendiosa, na ordem de $O(n^2)$ onde n representa o número total de nós na rede e, portanto, a alternativa foi implementar um novo estado para esses nós: *offline*. Neste estado as mensagens recebidas não são processadas nem consideradas para consumo energético, assim como também não são usados para a análise feita por módulos como o de cobertura da rede, fiabilidade, entre outros que possivelmente existam na

plataforma. Quando a simulação der a ordem de adição de uma certa quantidade de nós, serão escolhidos aleatoriamente de entre os nós *offline* e passarão então para o estado *online*.

Parametrização da dimensão da rede: Duas propriedades muito importantes numa simulação que deverão ser cuidadosamente parametrizadas e pelo utilizador são a dimensão da área de simulação (em metros) e a quantidade de sensores que deverão ser lançados nesta. Um grande número de nós concentrados numa área muito pequena resultará em muitas colisões de mensagens e, por outro lado, uma pequena quantidade de nós lançados numa área muito grande resultará em baixos índices de cobertura e fiabilidade. O módulo oferece então a possibilidade de estas duas propriedades serem parametrizáveis pelo utilizador.

Parametrização de ataques: Na secção de ataques é possível parametrizar o controlador de injecção de ataques da maneira que foi descrita no respectivo capítulo. As parametrizações passam por permitir uma rede com e sem ataques e, no caso de haver, qual o tipo de ataque utilizado pelo atacante, as suas configurações e ainda o número de nós directamente afectados, ou seja, que são alvos de intrusão.

Modelação

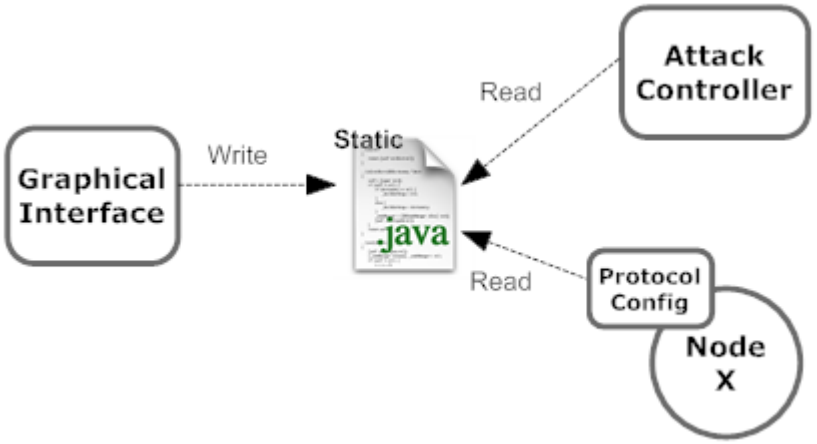

Figura 3.10 - Especificação do módulo de controlo de definições e parametrizações

Foi criado um objecto global à simulação que serve como repositório de parametrizações. Durante o decorrer da simulação, tanto os módulos como os intervenientes (nós) podem aceder ao repositório e obter o parâmetro que necessitam. A interface gráfica serve como um meio eficaz de alterar parâmetros importantes da simulação que deverá ser lançada mas, no entanto, foi tomado o cuidado de apenas apresentar parâmetros na interface que actuem directamente sobre os resultados da simulação, podendo depois o repositório ser estendido com os parâmetros que o utilizador achar convenientes.

3.2.3.7. Módulo de visualização e controlo da rede

De modo a permitir uma boa compreensão do funcionamento da rede durante a sua execução e de modo a permitir uma visualização mais apurada de uma simulação, foi estendido o módulo de visualização básico que o jProwler apresenta. As extensões visaram dar ao utilizador um maior controlo sobre a rede destacando-se uma maior interacção entre os mesmos com a possibilidade de parar a simulação, avançar um número parametrizável de eventos, adição ou remoção de nós e geração de dados para efeitos de teste de fiabilidade ou não. Foram também criadas primitivas que permitem definir listagens de informação relevante para cada nó participante, as quais podem ser consultadas em tempo real durante o funcionamento da rede.

Para dar suporte a estas extensões foi também implementada uma solução que permite ao utilizador seleccionar, em tempo real com um clique do rato, um nó participante e depois interagir com ele. Finalmente, foi também implementada a opção de guardar os resultados obtidos pela execução dos restantes módulos que constituem a base dos indicadores analisados e estudados no capítulo de testes. Este mecanismo foi implementado usando serialização de dados em disco.

A figura 3.11 mostra uma captura do ambiente de simulação onde estão explícitas as extensões descritas.

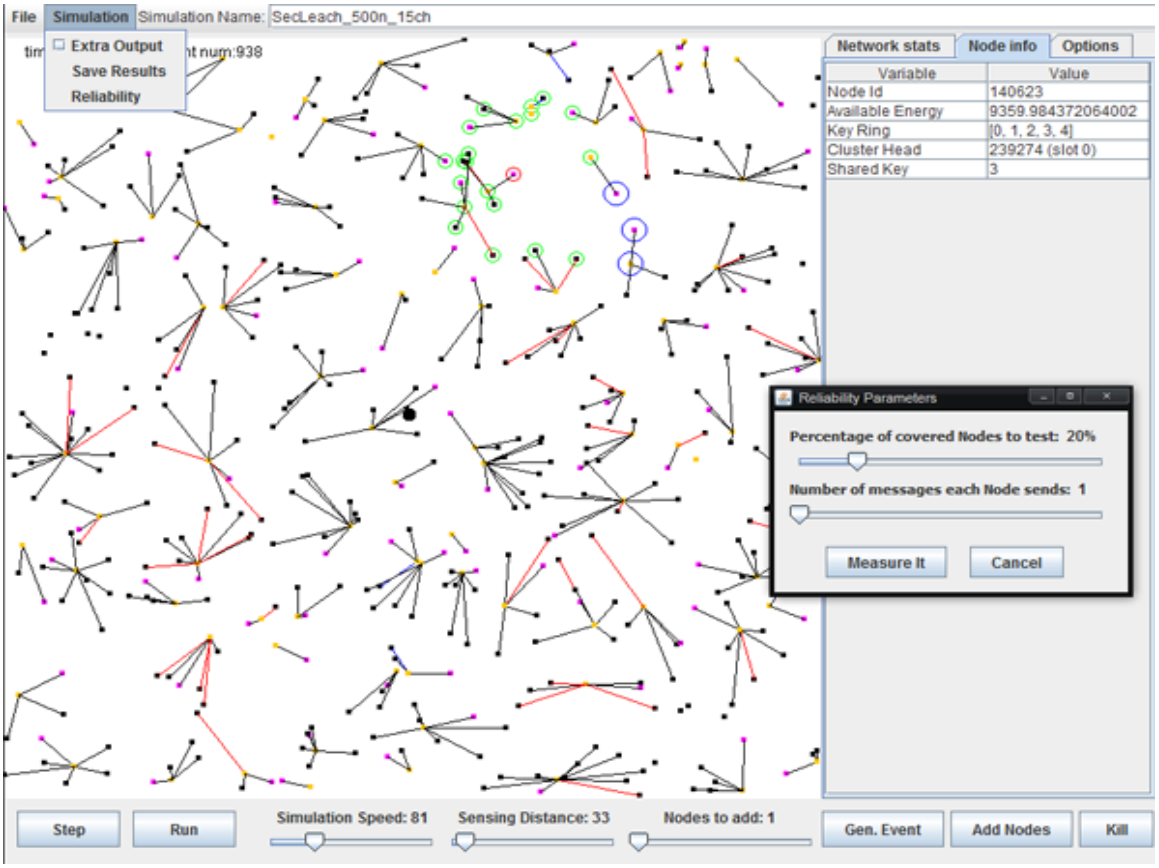

Figura 3.11 - Módulo de visualização (ambiente gráfico de simulação)

3.3. Resumo e visão geral da arquitectura da plataforma

A arquitectura final da plataforma composta pelo núcleo base de simulação e as extensões efectuadas pode ser vista na seguinte figura:

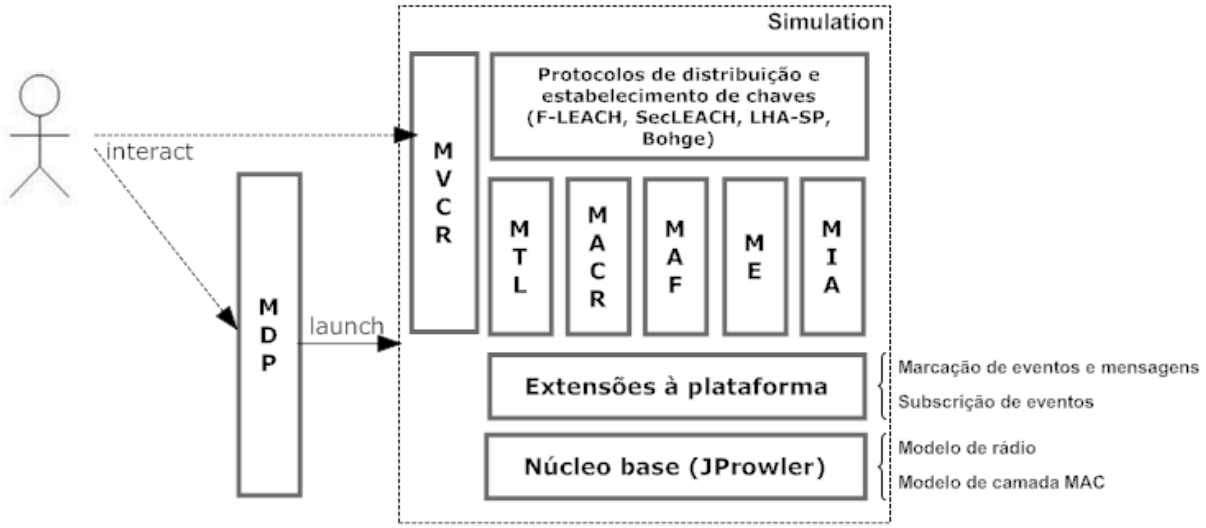

Figura 3.12 - Arquitectura da plataforma de simulação

O núcleo base de simulação fornece as primitivas do modelo de rádio em RSSF e respectiva camada MAC baseada na pilha IEEE 802.15.4, simulando sensores do tipo Mica Mote através de um motor de eventos discretos. Sobre esse núcleo base foi implementado um mecanismo de marcação e subscrição de eventos que serviu de base para a especificação e implementação de alguns dos módulos necessários à avaliação dos protocolos propostos bem como a concretização do objectivo de disponibilizar uma plataforma genérica para a simulação e extracção de indicadores referentes a protocolos para redes de sensores sem fios.

Ao utilizador foi dada a funcionalidade de parametrizar uma simulação, através do módulo de controlo de definições e parametrizações, permitindo então especificar o comportamento do protocolo a ser simulado assim como certas definições do núcleo base de simulação e respectivas extensões implementadas. O utilizador tem também controlo sobre o módulo de visualização e controlo da rede de modo a poder interagir com a mesma e poder observar o seu comportamento em tempo real, assim como dar ordem de escrita de resultados em disco para posterior análise.

4. Protocolos de distribuição e estabelecimento de chaves

Ao longo deste capítulo será descrita a arquitectura e especificação dos protocolos de distribuição e estabelecimento de chaves criptográficas que foram propostos e brevemente descritos no trabalho relacionado. Os protocolos têm em comum a formação hierárquica da rede, mas diferem bastante na atribuição de funções a cada nó e na definição do mesmo.

Nestes protocolos não foram seguidas todas as assumpções que os autores detalharam nos respectivos artigos de maneira a ser possível adaptar os mesmos a redes de grande escala e de maneira a poderem ser comparáveis entre si pois essa será uma das contribuições da presente dissertação. Outro aspecto importante é o facto de que certos mecanismos dos protocolos não se encontrarem detalhados nos artigos e ficam um tanto em aberto quanto à sua especificação pois os autores apenas assumem a presença dos mesmos. Em ambos os casos, e em cada protocolo, será sempre referido quando algo implementado não respeita a especificação original e/ou é implementado sem nenhuma especificação já ter sido detalhada pelo autor original.

A organização do capítulo começará com a descrição do modelo da rede onde os protocolos irão operar, seguido da especificação de cada um dos mesmos. Por protocolo será seguida uma estrutura consistente, onde serão apresentadas as diversas fases de operação, a especificação das inicializações dos nós antes do *deployment*, condições de cobertura, entre outros aspectos que sejam relevantes para a implementação e posteriormente para a obtenção de resultados e indicadores no capítulo de testes.

4.1. Modelo da rede e características dos nós

Para fornecer uma base de validação dos protocolos e permitir testes de âmbito variado foram implementadas três classes distintas de nós participantes na rede: nó sensor, nó encaminhador e *base station*. Os nós são distintos entre si nas suas capacidades/limitações e no seu modo de funcionamento e papel na rede. O nó encaminhador é o único específico a um dos protocolos

(Bohge) e como tal será descrito no respectivo capítulo. Para os restantes nós foram assumidas as seguintes características:

1) **Limitações dos nós**: Como descrito na introdução, os nós sensores possuem uma baixa capacidade energética, baixas capacidades computacionais e de comunicação muito limitadas devido às suas pequenas dimensões. Como tal, as aplicações desenvolvidas no âmbito de cada protocolo terão esse factor em conta. Ao contrário dos nós sensores, a *base station* é assumida como um nó especial que não sofre de restrições energéticas, de armazenamento, processamento, entre outras;

2) **Não mobilidade dos nós**: Desde que são inseridos na rede, os nós, sejam de que classe forem, ficam imóveis nas coordenadas onde foram lançados. Continua, no entanto, a ser possível o lançamento de novos nós na rede (préviamente lançados como *offline*) após o início do funcionamento desta, assim como a morte de nós que dela fazem parte (captura física, desgaste energético, etc..). Para redes sem mobilidade de nós a plataforma de simulação oferece um modelo de rádio adequado: o modelo gaussiano;

3) **Força de sinal**: Todos os nós sensores terão a mesma força de sinal e, consequentemente, o mesmo alcance máximo. Isto vai influenciar algumas assumpções de determinados protocolos e alterar a especificação dos mesmos mas é uma característica necessária para a escalabilidade da rede. A única excepção provem da primeira assumpção, ou seja, a *base station* poderá ter um poder de sinal suficiente para alcançar qualquer nó da rede por *single-hop*;

4) **Sincronização de relógios**: Assume-se que, após o *deployment*, todos os nós terão os respectivos relógios locais sincronizados e a noção de tempo é igual para qualquer participante da rede. Esta sincronização é uma característica fundamental para protocolos *clock-driven* como o LHA-SP e Leach.

4.2. Encaminhamento de dados

Mesmo não sendo o foco desta dissertação, foi necessário especificar e implementar uma camada universal de encaminhamento para os nós participantes na rede. A razão deve-se ao facto

de ser necessário fazer chegar dados à *base station* (e vice-versa) e, na grande maioria dos casos, não o ser possível fazer por *single-hop* visto que estamos a considerar redes de grande escala onde os sensores têm um alcance limitado. Como tal foi implementado um algoritmo básico de *flooding* com detecção de duplicados em cada nó. É também importante referir que o único protocolo que não dá uso a esta camada de encaminhamento é o LHA-SP visto possuir já uma arquitectura bem definida de encaminhamento de dados de nós sensores para a *base station*.

4.2.1. Algoritmo de encaminhamento

Trata-se então de um algoritmo de *flooding* cego onde os nós não possuem estado nem formam qualquer tipo de estrutura de disseminação de mensagens. A especificação do algoritmo pode ser vista na figura 4.1.

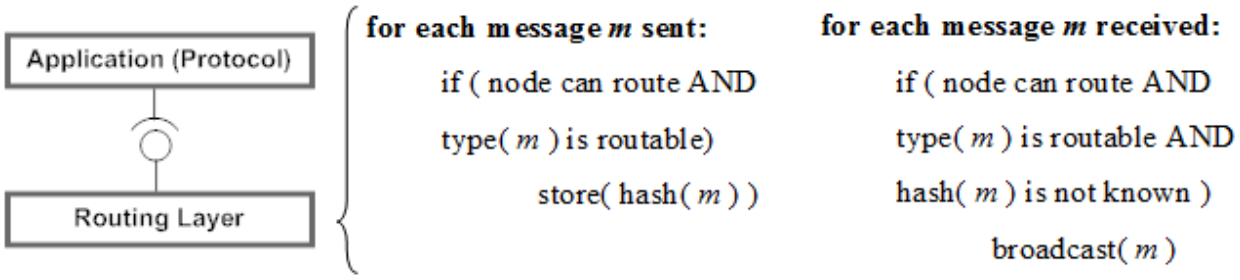

Figura 4.1 - Algoritmo de flooding cego de dados

Em termos de impacto energético este algoritmo implicará um custo bastante mais elevado nos protocolos que lhe dêem uso. No entanto o módulo de teste de consumo energético deverá será capaz de discriminar a percentagem do custo total que corresponde ao processamento do encaminhamento de dados para os protocolos que usem o serviço (usando a *flag ROUTING*). Em termos de impacto na cobertura será o menor possível visto que um algoritmo de *flooding* cego será sempre o melhor para a taxa de entrega de mensagens com sucesso. Já em termos de latência a mesma subirá consideravelmente e finalmente, em termos de resiliência a ataques, nomeadamente, descarte de pacotes, deverão obter-se bons resultados devido à exploração de múltiplas rotas.

4.3. F-Leach

O protocolo F-Leach é uma extensão ao Leach com o seu sistema de rondas, que introduz a garantia de autenticação dos nós que se querem tornar *cluster heads* (CH) através de segredos partilhados com a *base station* (BS) a qual usará um esquema µTESLA [24] para difundir listas de *cluster heads* autenticados. A implementação deste segue, quase na totalidade, a especificação apresentada pelo autor e será em seguida apresentada.

4.3.1. Especificação do protocolo

Setup phase

1. H ⇒ G : *adv*, $mac_{Kh}(id_H \mid c_H \mid adv)$

   A_i : store(idH)

   A_i →→ BS : *adv*, $mac_{Kh}(id_H \mid c_H \mid adv)$

   BS : if $mac_{kH}(id_H \mid c_H \mid adv)$ is valid, add(id_H,*V*)
2. BS ⇒ G : *list*, *V*, $mac_{Ki}(V)$
3. BS ⇒ G : *key*, k_i

   A_i : if($f(k_i) == k_{i+1}$) and ($id_H \in V$), H is authentic
4. A_i → H : *join_req*
5. H ⇒ G : *schedule*, (..., < id_{Ai},t_{Ai} >, ...)

Legenda	
H	: *Cluster Head*
Ai	: Nó comum
G	: Grupo de vizinhos
id_x	: Identificador do nó X
$mac_{kx}()$	: MAC calculado usando K_x
K_x	: Chave partilhada entre X e a BS
V	: Array de identificadores de nós
f	: Função de hash one-way
t_x	: Time slot (TDMA) para o nó X
⇒	: *Broadcast* para vizinhos
→	: *Unicast* para vizinho
→→	: *Unicast multi-hop*
d_x	: Relatório de evento do nó X
F	: Função de agregação

Steady-state phase

6. Ai → H : *report*, d_{Ai}, $mac_{Kai}(id_{Ai} \mid c_{Ai})$
7. H →→ BS : *report*, F(..., d_{Ai}, ...), $mac_{Kh}(id_H \mid c_H \mid F(..., d_{Ai}, ...))$
8. H →→ BS : *mac_array*, (..., id_{Ai}, $mac_{Kai}(id_{Ai} \mid c_{Ai})$, ...), $mac_{Kh}(id_H \mid c_H)$
9. BS → H : *intruder_list*, id_1, ..., id_N

Figura 4.2 - Especificação do protocolo F-Leach

Assim como o protocolo Leach, a operação do protocolo F-Leach é dividida em rondas e em cada ronda ocorre uma fase de formação da rede (*Setup*), sub-dividida em três fases, e uma fase de operação (*Steady-State*) onde os nós sensores transmitem dados que devem ser entregues à *base station*. Em seguida são apresentadas as fases pelas quais o protocolo se divide em cada ronda.

Pré-Deployment

Cada nó sensor X é carregado inicialmente com duas chaves: K_x, uma chave simétrica que X partilha com a *base station* e K_n, uma chave de grupo que é partilhada por todos os membros da rede. A chave K_x vai permitir ao nó autenticar-se perante a *base station* na fase de organização, enquanto que a chave K_n é a última de uma sequência S gerada aleatoriamente após aplicar sucessivamente uma função de *hash f* a uma chave inicial K_0 ($S = k_0, k_1, k_2, ..., k_{n-1}, k_n$, onde $f(k_i) = k_{i+1}$). A *base station* guarda S em segredo mas partilha a última chave da sequência conforme as rondas avancem. Este é o esquema µTESLA que permite *broadcasts* autenticados pela rede e que será importante na disseminação da lista de nós líderes de *clusters* autenticados. Por fim, cada nó X vai também partilhar um *counter* C_x com a *base station* para efeitos de frescura das mensagens.

(*Setup*) Fase de decisão

Durante a formação de *clusters* cada nó sensor terá de tomar a decisão de se tornar um *Cluster Head* ou de se juntar a um existente. Esta decisão segue a especificação do protocolo Leach e é baseada na percentagem sugerida de *cluster heads* para a rede, determinada antes da mesma ser formada, e o número de vezes que o nó já terá desempenhado essa função nas rondas anteriores. Para tal o nó X gera um número aleatório entre 0 e 1 e se esse número for menor que um certo limite então será *cluster head* na ronda actual. A função que devolve o limite para um nó X é dada por:

$$T(x) = \begin{cases} \dfrac{P}{1 - P * \left(r \bmod {}^{1}/_{P}\right)} & se\ x \in G \\ 0 & caso\ contrário \end{cases}$$

onde *P* é igual à percentagem sugerida de *cluster heads*, *r* é igual à ronda actual e G equivale ao grupo de nós que não o foram nas passadas 1/*P* rondas. O uso desta fórmula garante que um nó seja *cluster head* algures entre 1/P rondas e quando o for, não voltará a ser pelo menos durante o mesmo número de rondas.

(*Setup*) Fase de anúncio

Após um nó decidir ser *cluster head*, este vai transmitir um anúncio para os seus vizinhos (mensagem 1) a publicar a sua intenção e disponibilidade. Um nó comum que receba este anúncio vai guardar a referência para o mesmo e fica à espera da confirmação da *base station* em como é autêntico e o anúncio não foi feito por um nó atacante externo. Se ocorrer o caso em que um nó receba mais que um anúncio, este irá escolher o *cluster head* do anúncio que tenha recebido com maior força de sinal.

A mensagem de anúncio deverá ter um *mac* gerado a partir da chave que o anunciante partilha com a *base station* de modo a que a mesma, ao receber o anúncio, possa autenticar o nó em questão. É previsto que a BS receba vários anúncios (conforme o total de CHs na ronda corrente) e, após um determinado intervalo de tempo (parametrizável), divulga uma lista de *cluster heads* autenticados (mensagem 2) usando o esquema μTESLA para autenticar o *broadcast* da lista, ou seja, divulgando a chave da ronda corrente (k_i) que deverá ser validada após os nós aplicarem a função de *hash* (*f*) à mesma e o resultado coincidir com a chave da ronda anterior (k_{i+1}), preservando a ordem da série (S) que só a *base station* conhece na totalidade (mensagem 3). Foi também implementado um esquema de recuperação da ordem da série nos nós comuns que possam não receber uma ou mais divulgações das suas chaves. O esquema aplica a função de *hash* até um máximo número parametrizável de vezes, tentando compensar a perda, possivelmente contínua, da divulgação de chaves em cada ronda em alguns nós.

(*Setup*) Fase de Junção

Um nó comum, após receber a lista de *cluster heads* válidos, irá desencadear um pedido de junção àquele que tenha escolhido e que tenha sido autenticado pela *base station* (mensagem 4).

Já a cada *cluster head* cabe a tarefa de receber os vários pedidos e no final divulgar uma lista de *slots* temporais (mensagem 5), gerada por si, onde cada nó que tenha pedido a junção terá a sua vez de enviar relatórios para o mesmo. A atribuição de slots é feita segundo uma divisão do tempo (TDMA) e pela ordem de chegada dos pedidos de junção. Após o envio e recepção de todas as listas a fase de organização termina.

Fase de operação

Concluída a fase de organização todos os *clusters* encontram-se devidamente formados e os nós comuns enviam relatórios para o respectivo *cluster head* na sua vez (mensagem 6). Para autenticar a origem destes relatórios, os nós enviam, juntamente com a mensagem, um *mac* produzido com a chave que partilham com a *base station*. Os *cluster heads* recebem os relatórios e agregam-nos, encaminhando depois o resultado final para a mesma (mensagem 7) também autenticado por um *mac*. Já os *macs* dos nós comuns são posteriormente encaminhados para a *base station* numa outra mensagem que contém um *array* destes (mensagem 8). Finalmente a *base station* verifica a autenticidade de todos os relatórios e se detectar alguma falha, a mesma descarta o relatório final agregado e divulga uma lista de intrusos (mensagem 9).

4.3.2. Limitações da especificação original

O protocolo apresenta limitações graves para redes de grande escala. Isto deve-se ao facto de a especificação original do protocolo prever que a *base station* recebe os anúncios dos *cluster heads* por *single-hop*, levando a que cada um destes utilize uma intensidade de sinal suficiente para alcançar a mesma, o que torna a especificação bastante irrealista para redes de larga escala. Assim a especificação original foi ignorada neste contexto e a solução encontrada passa por encaminhar os anúncios até à *base station* usando camada de encaminhamento descrita na secção 4.2 e onde todos os nós são potenciais encaminhadores, levando os anúncios até à mesma por *multi-hop*.

A mesma situação aplica-se na entrega de relatórios após o agrupamento nos respectivos *cluster heads*. A especificação original prevê, mais uma vez, que o *cluster head* comunique com a *base station* por *single-hop* e, pelas mesmas razões, a implementação passou por ignorar tal especificação e recorreu-se à camada de encaminhamento.

4.3.3. Condições de cobertura

A cobertura deste protocolo segue a especificação definida pelo módulo de cobertura (capítulo 3.2.3.2) e, portanto, foram definidas as seguintes condições para os tipos de cobertura estática e dinâmica:

- *Cobertura Topológica*: um nó sensor está topologicamente coberto de possuir um vizinho que seja *cluster head* na ronda actual ou ele mesmo o seja. Com este tipo de cobertura facilmente se identifica os nós que não conseguem formar um *cluster*.
- *Cobertura do Protocolo*: um nó está coberto pelo protocolo se estiver topologicamente coberto e, se a partir do *cluster head* respectivo, houver um caminho até à *base station* dado pelo protocolo de encaminhamento *multi-hop*. A mesma condição de haver caminho também se aplica para os nós que sejam *cluster heads* na ronda actual.
- *Cobertura Parcial*: após a conclusão da fase de organização e execução do protocolo, um nó estará parcialmente coberto se fizer parte de um *cluster*, ou seja, se enviou um pedido de junção para um *cluster head* vizinho e se tiver recebido do mesmo uma slot temporal para envio periódico de relatórios. Em alternativa se um nó for *cluster head* então estará sempre parcialmente coberto.
- *Cobertura Total*: um nó poderá ser considerado totalmente coberto se estiver parcialmente coberto e coberto também pelo protocolo através de uma análise estática, ou seja, se fizer parte de um *cluster* e, se a partir do respectivo *cluster head*, houver caminho até à *base station*.

4.4. SecLeach

O protocolo SecLeach é também uma extensão ao Leach. A grande diferença está na introdução de um sistema de autenticação de nós comuns perante *cluster heads*, usando para isso o esquema de pré-distribuição de chaves de Eschenauer [4]. A implementação deste segue, quase na totalidade, a especificação apresentada pelo autor e será em seguida apresentada.

4.4.1. Especificação do protocolo

Setup phase

1. H ⇒ G : *adv*, *nonce*

   A_i : choose *r* such that $r \in (R_H \cap R_{Ai})$
2. $A_i \rightarrow H$: *join_req*, *r*, $mac_{Kr}(id_{Ai} \mid id_H \mid r \mid nonce)$
3. H ⇒ G : *schedule*, (..., $< id_{Ai}, t_{Ai} >$, ...)

Steady-state phase

4. $A_i \rightarrow H$: d_{Ai}, mac_{Kr} ($id_{Ai} \mid id_H \mid nonce + j$)
5. H →→ BS : F(..., d_{Ai}, ...), mac_{Kh}(F(..., d_{Ai}, ...) | c_H)

Legenda

Todos os símbolos anteriormente definidos com as seguintes adições:

r : identificador de uma chave do anel de chaves	K_r : Chave simétrica associada com o identificador r
R_x : Conjunto de chaves no anel de chaves do nó X	*j* : contador de ciclo de envio de relatórios

Figura 4.3 - Especificação do protocolo SecLeach

O protocolo SecLeach também se divide em rondas, assim como o Leach e F-Leach. Em cada ronda voltam a ocorrer as mesmas duas fases principais, as quais são em seguida especificadas.

Pré-Deployment

Esquema RKPS

Foi abordado no trabalho relacionado um esquema de distribuição aleatória de chaves proposto por Eschenauer e Gligor [4]. Esse esquema visa gerar uma grande colecção de chaves aleatórias (*key pool*) de onde serão aleatoriamente escolhidas algumas para integrarem o conjunto de chaves de cada nó (*key ring*). Após o *deployment* os nós deverão verificar com os vizinhos directos a partilha de chaves para estabelecimento de uma ligação segura entre ambos.

Segundo Eschenauer e Gligor a probabilidade de dois nós partilharem pelo menos uma chave em função das dimensões do *key ring* (*k*) e da *key pool* (*P*) é dada pela fórmula abaixo:

$$p = 1 - \frac{\left((P-k)!\right)^2}{(P-2k)!\,P!} \cong \frac{(1-k/P)^{2\left(P-k+{}^1/_2\right)}}{(1-2k/P)^{\left(P-2k+{}^1/_2\right)}}$$

Inicialização de nós

O protocolo toma partido do esquema RKPS para garantir a autenticidade de nós perante *cluster heads*. A especificação do protocolo prevê a criação de um conjunto de *S* chaves e os seus respectivos identificadores antes do *deployment* da rede. Cada nó receberá depois *m* chaves pertencentes a esse conjunto obtidas de forma pseudo-aleatória e sem repetições. Esse conjunto em cada nó será o equivalente ao anel de chaves no esquema RKPS. Para gerar o conjunto de *m* chaves é usado um gerador de números pseudo-aleatório que, para o identificador correspondente do nó, devolverá uma sequência de *m* números. Cada um desses números será depois mapeado à chave correspondente (identificador) e a mesma será atribuída ao anel do nó.

Por fim, a cada nó é atribuída uma chave simétrica partilhada entre o mesmo e a *base station* para efeitos de autenticação perante a mesma. A par dessa chave será também atribuído um *counter*, partilhado entre ambos, para efeitos de frescura de mensagens.

(*Setup*) Fase de decisão

Durante a formação de *clusters* cada nó sensor terá de tomar a decisão de se tornar um *cluster head* na ronda corrente ou de se juntar a um existente. Esta decisão segue a especificação do protocolo Leach e é baseada na percentagem sugerida de *cluster heads* para a rede, determinada antes da mesma ser formada, e o número de vezes que o nó já o foi nas rondas anteriores. O cálculo da decisão de um nó se tornar *cluster head* está detalhado no sub-capítulo anterior referente ao protocolo F-Leach e é exactamente o mesmo usado no SecLeach.

(*Setup*) Fase de anúncio

Após um nó decidir ser *cluster head*, este vai transmitir um anúncio para os seus vizinhos (mensagem 1) a publicar a sua intenção e disponibilidade. Um nó comum que receba este anúncio vai guardar a referência para o mesmo e fica à espera de mais anúncios de modo a decidir a qual *cluster head* se vai juntar. Sabendo o identificador do *cluster head* (id_H), que é obtido na mensagem de anúncio, um nó comum irá obter os identificadores das chaves que o mesmo possui usando a função geradora de números pseudo-aleatórios que foi descrita na fase do *pré-deployment*. Sabendo o conjunto de chaves do *cluster head* e o seu próprio conjunto, um nó irá então escolher um *cluster head* com quem partilhe pelo menos uma chave ($R_H \cap R_{Ai}$!= $\emptyset$) e desempata pela força de sinal com que recebeu o anúncio, o que muito provavelmente evidencia o *cluster head* mais próximo.

(*Setup*) Fase de Junção

Após a escolha de um *cluster head*, o nó comum envia um pedido de junção ao mesmo (mensagem 2). O pedido terá o identificador *r* da chave escolhida pelo nó para proteger o canal, escolhida de entre as chaves partilhadas entre ambos ($R_H \cap R_{Ai}$). Além do identificador o *cluster head* recebe também um *mac* gerado usando a chave K_r, correspondente ao identificador, o que garante a autenticidade do nó perante o mesmo pois um atacante externo não consegue inferir a chave sabendo só o identificador. Esse *mac* irá conter o *nonce* do anúncio respectivo do CH para efeitos de frescura do pedido de junção.

Após receber os vários pedidos de junção de nós vizinhos, um *cluster head* difunde uma lista de slots TDMA (mensagem 3) usando a mesmo especificação do protocolo F-Leach descrita no subcapítulo anterior.

Fase de operação

Concluída a fase de organização todos os *clusters* encontram-se devidamente formados e os nós comuns enviam relatórios para o respectivo *cluster head* na sua vez (mensagem 4). Para autenticar a origem destes relatórios, os nós enviam, juntamente com a mensagem, um *mac*

gerado com a chave escolhida durante a fase de organização. Para efeitos de frescura o *mac* contém o *nonce* esperado pelo CH somado com um contador de ciclo de envio de relatórios.

Os *cluster heads* recebem os relatórios e agregam-nos usando a função F, encaminhando o resultado, na sua vez segundo o esquema TDMA, para a *base station* (mensagem 5). Essa mensagem, que contém o resultado, será autenticada mais uma vez por um *mac*, desta vez gerado a partir da chave que o *cluster head* partilha com a *base station*. Convém também referir que a especificação original foi alterada neste último passo. A razão é a mesma do protocolo F-Leach, ou seja, é impensável assumir que qualquer nó sensor, consiga alcançar a mesma com apenas um *hop* e uma força de sinal brutal numa rede de grande escala. Assim a alternativa passou mais uma vez por enviar o relatório segundo o protocolo de encaminhamento *multi-hop* por *flooding*.

4.4.2. Condições de cobertura

Foram definidas as seguintes condições de cobertura para os tipos especificados pelo módulo de cobertura:

- *Cobertura Topológica*: um comum nó sensor está topologicamente coberto de possuir um vizinho que seja *cluster head* ou ele mesmo o seja. O *cluster head* vizinho tem também de partilhar pelo menos uma chave no seu *key ring*.
- *Cobertura do Protocolo*: um nó está coberto pelo protocolo se estiver topologicamente coberto e, se a partir do *cluster head* respectivo, houver um caminho até à *base station* dado pelo protocolo de encaminhamento *multi-hop*.
- *Cobertura Parcial*: após a conclusão da fase de *setup* e execução do protocolo, um nó estará parcialmente coberto se fizer parte de um *cluster*, ou seja, se enviou um pedido de junção para um *cluster head* vizinho, com quem partilha chave, e se tiver recebido do mesmo uma *slot* temporal para envio periódico de relatórios. Em alternativa se um nó for *cluster head* então estará sempre parcialmente coberto.
- *Cobertura Total*: um nó poderá ser considerado totalmente coberto se estiver parcialmente coberto e coberto também pelo protocolo através de uma análise estática, ou seja, se fizer parte de um *cluster* e, se a partir do respectivo *cluster head*, houver caminho até à *base station*.

4.5. LHA-SP

O protocolo LHA-SP prevê a formação de uma hierarquia de níveis desde o nó mais forte (a *base station*) até aos nós de nível mais baixo, onde um nó de nível *h+1* irá tentar formar *cluster* com nós de nível *h*. O protocolo é recursivo na medida em que se inicia nos níveis superiores e rapidamente se propaga para os restantes. Em seguida será apresentada a sua especificação que, neste caso, seguiu na totalidade a especificação original dada pelo autor.

4.5.1. Especificação do protocolo

Setup phase

1. $A_h \Rightarrow G_{h-1} : \{adoption\text{-}ad \mid h \mid id_A \}\ K_G$
   $B_h \Rightarrow G_{h-1} : \{adoption\text{-}ad \mid h \mid id_B \}\ K_G$
   ...
2. $M_{h-1} \rightarrow A_h : \{adoption\text{-}req \mid id_M \mid id_A\}\ K_G$
   $N_{h-1} \rightarrow B_h : \{adoption\text{-}req \mid id_N \mid id_B\}\ K_G$
   ...
3. $A_h \rightarrow M_{h-1} : \{send\text{-}key \mid id_A \mid id_M \mid K_{A,M}\}\ K_G$
   $B_h \rightarrow N_{h-1} : \{send\text{-}key \mid id_B \mid id_N \mid K_{B,N}\}\ K_G$
   ...

Network operation phase

4. $A_h \rightarrow D_{h+1} : \{sensing \mid n_A \mid m_A\}\ K_{A,D}$

Legenda

Todos os símbolos anteriormente definidos com as seguintes adições:

h : identificador de nível	K_G : Chave simétrica global
A_x : Nó pertencente ao nível x	$K_{A,B}$: Chave simétrica partilhada entre A e B
n_A : *Nounce* para efeitos de frescura	m_A : Relatório de evento do nó A

Figura 4.4 - Especificação do protocolo LHA-SP

No caso do LHA-SP deixa de existir o conceito de ronda e os nós participantes terão de formar a hierarquia final da rede durante o intervalo de tempo em que a chave global é válida. Devido a este facto surge a possibilidade de alguns nós morrerem e uma ligação ser quebrada após esse intervalo. Nestes casos o protocolo prevê um mecanismo de adopção de órfãos que será especificado mais à frente. Por agora apresenta-se a especificação das duas fases principais que compõem a formação hierárquica inicial da rede, assim como a inicialização dada a todos os nós participantes.

Pré-Deployment

Cada nó sensor X é carregado inicialmente com os seguintes dados: K_x, uma chave simétrica que X partilhada exclusivamente com a *base station* e K_G, uma chave global partilhada por todos os elementos da rede. Esta chave global será válida apenas durante a fase de organização pelo que deverá ser descartada após um determinado tempo, parametrizável e igual para todos os nós. A razão prende-se com a insegurança da existência de uma chave global que foi um tema já abordado no trabalho relacionado.

Além das duas chaves é também estaticamente atribuído um nível na hierarquia a que o nó pertencerá. Esse nível deverá ser atribuído conforme as capacidades do mesmo (energética, processamento, etc..).

Fase de organização

A fase de organização consiste na formação de hierarquias de *clusters* e na distribuição de chaves pelos mesmos. Essa formação ocorre em múltiplos estágios, numa aproximação de cima para baixo, ou seja, nós de nível *h+1* tentam formar um *cluster* com nós de nível *h*, os quais distribuem posteriormente chaves simétricas partilhadas par-a-par. Em seguida o mesmo protocolo é executado entre os nós de nível *h* e os de nível *h-1* e isto sucessivamente até ao nível 1. O nível superior, onde o protocolo se inicia, corresponde ao nível da *base station*. A especificação em cima apresentada demonstra um desses estágios.

Em cada estágio, nós de nível *h* fazem *broadcast* para o grupo de vizinhos (mensagem 1) de um anúncio de disponibilidade de aceitação de nós de nível inferior para formação de *cluster*.

Esta mensagem inclui o nível hierárquico do nó de nível superior, assim como o seu identificador para que os nós de nível inferior saibam que são eles os destinatários e quem fez *broadcast* do anúncio.

Durante um certo período de tempo, parametrizável, os nós de nível inferior guardam anúncios de nível superior para poderem escolher um que melhor se adapte às características da aplicação. No caso da presente dissertação foi escolhido o critério do nó mais próximo, medido através da força de sinal com que recebe o anúncio.

Um nó de nível inferior irá então depois enviar um pedido de junção (mensagem 2) para o *cluster head* escolhido e este, ao receber o pedido, vai gerar uma chave simétrica que será partilhada entre os dois para proteger o canal de comunicação criado. A chave vai ser entregue depois ao nó de nível inferior (mensagem 3).

Todas as mensagens trocadas durante esta fase deverão ser protegidas pela chave global da rede. Em cada passo um nó verifica se a mensagem recebida teve origem num nó participante legítimo, usando a mesma chave, e só reagirá se tal acontecer. Esta chave tem também uma validade devido à insegurança já discutida pelo que, após um certo tempo de operação, cada nó deverá descartar o conhecimento que tem desta.

Fase de operação

Assim que um nó termine a sua própria fase de organização, ou seja, assim que tenha formado *cluster*, o mesmo entra em modo de operação e fica pronto a enviar relatórios para a *base station*. Para tal um nó envia um relatório cifrado (mensagem 4) com a chave simétrica que foi atribuída pelo *cluster head* para proteger o canal partilhado entre os dois. Para efeitos de frescura é enviado também um *nonce*. O respectivo *cluster head* irá depois decifrar o conteúdo e verificar a autenticidade do mesmo. Este processo vai ser repetido em cada *hop* até que o relatório chegue ao último nível (*base station*) criando então um algoritmo de encaminhamento próprio que garante um número fixo de *hops* conforme o nível do nó que envia o relatório.

Fase de manutenção

Um nó de nível h+1, ao morrer, torna impossível a todos os nós de nível h fazerem chegar os seus relatórios à *base station* devido à topologia imposta durante a fase de organização. Por esse motivo é necessário haver um mecanismo tolerante a estas falhas que permita a reorganização topológica desses nós que ficaram sem *cluster head*. No caso dos protocolos baseados no Leach tal mecanismo não é necessário pois o próprio sistema de rondas garante, de uma forma indirecta, a reorganização nas rondas seguintes. Já no caso do presente protocolo foi necessário implementar a especificação do autor para a adopção de órfãos e que é em seguida apresentada.

1. $A_h \rightarrow G_{h+1}$: *orphan-ad*, *h*
2. $B_{h+1} \rightarrow G_h$: *adoption-ad*, *h+1*, id_B
3. $A_h \rightarrow B_{h+1}$: *adoption-req*, id_A, id_B, n_A, $mac_{Ka}(id_A, id_B, n_A)$
4. $B_{h+1} \rightarrow C_{h+2}$: {*key-req* | id_A | id_B | n_A | n_B | $mac_{Ka}(id_A, id_B, n_A)$ | $mac_{Kb}(id_A, id_B, n_B)$} K_{BC}
5. $C_{h+2} \rightarrow\rightarrow$ BS : {*key-req* | id_A | id_B | n_A | n_B | $mac_{Ka}(id_A, id_B, n_A)$ | $mac_{Kb}(id_A, id_B, n_B)$} K_{BC}

   Base Station autentica A e B e gera $K_{A,B}$

6. BS $\rightarrow A_h$: *key-del*, {id_B | n_A | $K_{A,B}$} K_A
7. BS $\rightarrow B_{h+1}$: *key-del*, {id_A | n_B | $K_{A,B}$} K_B

Figura 4.5 - Especificação da fase de manutenção do protocolo LHA-SP

Em primeiro lugar é necessário um nó de nível *h* aperceber-se que o nó de nível *h+1*, com quem formou *cluster*, deixou de existir. Para tal foi implementado um sistema de mensagens *Hello* que cada nó de nível *h+1* envia periodicamente dando a conhecer aos nós de nível *h* que o mesmo se encontra vivo. Assim, após um tempo parametrizável, se um nó não receber um *Hello* do *cluster head* respectivo, então pode assumir que o mesmo deixou de existir e pode iniciar depois o protocolo de adopção.

Inicialmente o nó órfão faz *broadcast* de um pedido de adopção (mensagem 1) e os nós vizinhos de nível superior respondem com a possibilidade de adopção (mensagem 2). O órfão irá depois escolher o melhor vizinho a quem se juntar e envia o pedido efectivo de adopção (mensagem 3). Como os nós envolvidos (órfão e *cluster head*) não possuem segredos partilhados

então ambos terão de se autenticar perante a *base station*, a qual ficará responsável por distribuir uma nova chave pelos dois. Essa autenticação é feita através de *macs* gerados pelas chaves par-a-par que ambos partilham com a mesma. Assim, o pedido de adopção composto pelos *macs* dos dois nós envolvidos chega até à *base station* seguindo a hierarquia do protocolo e a mesma irá gerar uma nova chave e distribuir a mesma pelos nós (mensagem 6 e 7). No final, o nó que antes era órfão, irá partilhar uma chave com um novo *cluster head* e deixará de ficar isolado, podendo fazer chegar de novo os seus relatórios à *base station*.

4.5.2. Condições de cobertura

Foram definidas as seguintes condições de cobertura para os tipos especificados pelo módulo de cobertura:

- *Cobertura Topológica*: um nó terá de ter um vizinho de nível imediatamente superior para que possa ser considerado coberto pela topologia do protocolo. Um desses vizinhos será depois escolhido para formar *cluster*.
- *Cobertura do Protocolo*: um nó está coberto pelo protocolo se estiver topologicamente coberto e, se a partir de pelo menos um dos vizinhos de nível superior, houver um caminho para a *base station* seguindo a especificação do protocolo, ou seja, se esse vizinho possuir também um nó de nível imediatamente superior ao seu e daí em diante até ao último nível.
- *Cobertura Parcial*: após a conclusão da fase de organização, um nó estará parcialmente coberto se fizer parte de um *cluster* e partilhar uma chave simétrica com o respectivo *cluster head*.
- *Cobertura Total*: devido à especificação do protocolo, um nó coberto parcialmente estará também coberto totalmente devido à formação sequencial de níveis a partir do nó de nível mais alto.

4.6. Framework de Bohge

A *framework* de Bohge prevê a existência de dois tipos de nós na rede com duas funções bem distintas. Por um lado existem os nós sensores que capturam eventos e enviam os mesmos para a

base station e, por outro, existem os nós encaminhadores que existem apenas para encaminhar os relatórios dos nós sensores. A formação da rede pode então ser vista como uma hierarquia de dois níveis (face aos *h* níveis do LHA-SP).

Ao contrário dos restantes protocolos, não existe agora a noção de fases por grupo de nós e cada um destes executa o protocolo de forma individualista. Assim, cada nó terá uma fase de organização e uma fase de operação à semelhança dos restantes protocolos.

Em seguida é apresentada a especificação da *framework* de Bohge que segue a especificação original do autor excepto no facto de não se considerar a mobilidade de nós, ou seja, não foi necessária a implementação de certos mecanismos do protocolo.

4.6.1. Especificação do protocolo

Setup phase (base station)

1. B → A : *offer*, $SIGN_{-Kb}$(*offer*), CertTTP(id_B)
2. **if** (A accepts offer) **then**
   if ($SIGN_{-Kb}$(*offer*) is valid) **then**
   A → B : *ok*, $SIGN_{-Ka}$(*ok*), $\{K_{A,B}, gK_{AP}\}+K_B$
   else
   A → B : *deny*, $SIGN_{-Ka}$(*deny*)
   else
   A → B : *LoI*, $SIGN_{-Ka}$ (*LoI*)

Setup phase (sensor node)

1. D → C : *snReq*, mac_{iKd}(*snReq*), iCertTTP(id_D)
2. C → B : *snReq*, mac_{iKd}(*snReq*), iCertTTP(id_D)
3. B → A : *snReq*, mac_{iKd}(*snReq*), iCertTTP(id_D), id_B, $mac_{Ka,b}$(*snReq*)
4. **if** (mac_{iKd}(*snReq*) is valid) **then**
   A → B : *ok*, $mac_{Ka,b}$(*ok*), $\{K_{B,D}\}K_{A,B}$, mac_{iKd}(*ok*), $\{K_{A,D},K_{B,D}\}iK_D$
   B → C : *ok*, $mac_{Kb,d}$(*ok*), mac_{iKd}(*ok*), $\{K_{A,D},K_{B,D}\}iK_D$
   C → D : *ok*, $mac_{Kb,d}$(*ok*), mac_{iKd}(*ok*), $\{K_{A,D},K_{B,D}\}iK_D$
   else

$A \rightarrow B$: *nok*, mac_{iKd}(*nok*), $mac_{Ka,b}$(n*ok*)

$B \rightarrow C$: *nok*, mac_{iKd}(*nok*)

$C \rightarrow D$: *nok*, mac_{iKd}(*nok*)

Figura 4.6 - Especificação da fase de organização da framework de Bohge

Ao contrário dos restantes protocolos, a *framework* de Bohge prevê a separação entre a *base station* (B) e a aplicação (A) e atribui-lhes diferentes funções e conteúdos. Para a *framework* a *base station* é apenas um meio de acesso à aplicação e, assim como os nós sensores (D), terá de se autenticar perante a mesma para fazer os relatórios, que por ela passem, chegarem à aplicação. A comunicação entre a *base station* e aplicação é modelada segundo um canal físico de comunicação e, portanto, fica um pouco fora do escopo dos objectivos de avaliação da presente dissertação. No entanto todas as comunicações entre estas duas entidades foram implementadas pois poderá ser interessante verificar o seu comportamento (consumo energético por exemplo). Já os nós encaminhadores (C) são classificados como nós fortes, capazes de executar criptografia assimétrica, e seguem a implementação do algoritmo de encaminhamento *multi-hop* previamente definido para encaminhar as mensagens dos nós sensores até à *base station* e vice-versa.

Pré-Deployment

Como a *framework* prevê o uso de criptografia assimétrica nos nós mais fortes, todos eles terão um par de chaves, ou seja, a aplicação será carregada com o par $\{+K_A,-K_A\}$, a *base station* com $\{+K_B,-K_B\}$ e cada um dos nós encaminhadores com $\{+K_C,-K_C\}$.

Cada nó sensor é carregado inicialmente com um certificado pessoal (*iCert*) emitido por um nó em qual todos confiam (TTP) e que é capaz de executar assinaturas RSA. O certificado será associado à aplicação e terá de ser apresentado pelo nó sensor à *base station* como garantia de autenticidade do mesmo, o que lhe permitirá juntar-se à rede.

$$\mathbf{iCert_{TTP}(D)} = (id_D, \{iK_D\}+K_A, TS_{TTP}, SIGN_{-Kttp}(\ldots))$$

O TTP emite o certificado ao nó sensor juntamente com a sua chave inicial iK_D (chave simétrica), a qual será usada para a autenticação inicial do mesmo perante a *base station* e onde

só esta poderá obtê-la no segundo campo do certificado. O resto do certificado contém o identificador do nó ao qual foi emitido, um *timestamp* que indica a validade do mesmo e a assinatura do TTP que permite verificar a sua autenticidade por todos os nós que possam executar criptografia assimétrica.

Por fim, tanto os nós encaminhadores como a *base station* recebem um certificado também assinado pelo TTP para se poderem autenticar perante a aplicação. Como ambas as classes de nós são capazes de executar criptografia assimétrica então não é preciso introduzir uma chave simétrica adicional como nos certificados dos nós sensores. Em vez disso o certificado terá a respectiva chave pública, seguindo o formato dos certificados X.509.

$$\mathbf{Cert_{TTP}(B)} = (id_B, TS_{TTP}, +K_B, SIGN_{-Kttp}(\ldots))$$

Fase de organização

A fase de organização consiste na autenticação dos nós sensores perante a aplicação, assim como da *base station*. A *base station*, sendo um nó sem qualquer tipo de restrição, pode autenticar-se com recurso a criptografia assimétrica. Assim que entra na rede envia a sua oferta de serviços (mensagem 1) que contém o seu certificado e uma assinatura da oferta. A aplicação, com recurso ao certificado e após o mesmo ser validado (admite-se o conhecimento da chave pública do TTP), verifica a assinatura e aceita ou recusa a oferta em conformidade (mensagem 2). No caso da oferta ser aceite a aplicação retorna à *base station* uma chave simétrica partilhada entre ambas e uma chave global do grupo das *base stations*. É prevista também a possibilidade de se rejeitar a oferta de serviços mesmo sendo o pedido válido.

Já a autenticação dos nós sensores é feita com recurso à chave de inicialização de cada um destes e ao mecanismo de *macs* devido à impossibilidade do uso de criptografia assimétrica. Quando um nó sensor entra na rede envia um pedido à aplicação (mensagem 1) com o seu certificado e um *mac* gerado a partir da chave de inicialização. A *base station*, ao receber o pedido, e antes de o reencaminhar para a aplicação, adiciona-lhe o seu identificador e um *mac* gerado a partir da chave que partilha com esta. A aplicação, ao receber o pedido, valida o *mac* da *base station* e, após validar o certificado, obtém a chave de inicialização para validação do *mac* do nó sensor. Finalmente, a aplicação distribui novos segredos para que o nó sensor se possa autenticar perante a *base station* ($K_{B,D}$) e a própria aplicação ($K_{A,D}$).

Fase de operação

Assim que um nó sensor esteja autenticado o mesmo terá uma chave simétrica que partilha com a *base station* e outra que partilha com a aplicação. Com base nessas chaves poderá autenticar-se perante essas duas entidades e fazer os seus relatórios serem aceites pela aplicação. A especificação da fase de operação deste protocolo é então a seguinte:

1. $D \rightarrow C$: *data*, $\{rn\}K_{B,D}$, $mac_{Kb,d}(data)$, $mac_{Ka,d}(data)$
2. $C \rightarrow B$: *data*, $\{rn\}K_{B,D}$, $mac_{Kb,d}(data)$, $mac_{Ka,d}(data)$
3. **if** ($mac_{Kb,d}(data)$ is valid) **then**

   $B \rightarrow A$: *data*, $mac_{Ka,d}(data)$, $mac_{Ka,b}(data)$

   $B \rightarrow C$: *rnplusone*, $\{rn + 1\}K_{B,D}$

   $C \rightarrow D$: *rnplusone*, $\{rn + 1\}K_{B,D}$
4. **if** ($mac_{Ka,b}(data)$ is valid) **then**

   if ($mac_{Ka,d}(data)$ is valid) **then** A processes data

   else

   $A \rightarrow B$: *dRej*, id_D, $mac_{Ka,b}(dRej, id_D)$, $mac_{Ka,d}(dRej, id_D)$

   $B \rightarrow C$: *dRej*, id_D, $mac_{Kb,d}(dRej, id_D)$, $mac_{Ka,d}(dRej, id_D)$

   $C \rightarrow D$: *dRej*, id_D, $mac_{Kb,d}(dRej, id_D)$, $mac_{Ka,d}(dRej, id_D)$

   else

   $A \rightarrow B$: *dRej*, id_B, $mac_{Ka,b}(dRej, id_B)$, $mac_{Ka,d}(dRej, id_B)$

   $B \rightarrow C$: *dRej*, id_B, $mac_{Kb,d}(dRej, id_B)$, $mac_{Ka,d}(dRej, id_B)$

   $C \rightarrow D$: *dRej*, id_B, $mac_{Kb,d}(dRej,id_B)$, $mac_{Ka,d}(dRej, id_B)$

Figura 4.7 - Especificação da fase de operação da framework de Bohge

Um nó sensor, além do relatório, envia dois *macs* gerados com as chaves que partilha com a *base station* e com a aplicação, autenticando-se perante ambas as entidades, juntamente com um número aleatório cifrado com a chave que partilha com a primeira. Os vários nós encaminhadores limitam-se a encaminhar a mensagem até à *base station* que, ao recebê-la, autentica o nó sensor e adiciona o seu próprio *mac* gerado com a chave que partilha com a

aplicação para o mesmo objectivo de autenticação. Adicionalmente responde ao nó sensor com o incremento do número aleatório, dando-lhe a conhecer o sucesso da entrega do relatório. Como só a *base station* pode obter o número devido à cifra então o nó consegue autenticar a resposta. Finalmente, após a validação dos *macs*, a aplicação processa os dados ou responde a ambos no caso de uma das verificações falhar, usando o mesmo sistema de autenticação por *macs* no sentido inverso.

4.6.2. Condições de cobertura

Foram definidas as seguintes condições de cobertura para os tipos especificados pelo módulo de cobertura:

- *Cobertura Topológica*: como um nó sensor depende dos nós encaminhadores para fazer chegar as suas mensagens à *base station* então este estará coberto topologicamente se, na sua vizinhança, tiver pelo menos um nó encaminhador ou a própria *base station*.
- *Cobertura do Protocolo*: um nó está coberto pelo protocolo se estiver topologicamente coberto e, se a partir de um dos nós encaminhadores vizinhos, houver um caminho até à *base station* dado pelo protocolo de encaminhamento *multi-hop*, assim como no sentido inverso.
- *Cobertura Parcial e total*: após o envio do pedido de autenticação à *base station*, se um nó obteve resposta da mesma então considera-se estar parcialmente e totalmente coberto visto que consegue fazer chegar as suas mensagens à *base station* e vice-versa.

5. Implementação

Neste capítulo é apresentada a arquitectura de implementação da presente dissertação, distribuída na implementação da plataforma de simulação e na implementação dos protocolos de distribuição e estabelecimento de chaves. É também apresentada uma aproximação àquela que foi a dimensão do trabalho realizado.

5.1. Arquitectura de implementação

A figura 5.1 apresenta a arquitectura geral de todo o ambiente de simulação onde se insere a plataforma cuja arquitectura foi descrita no capítulo 3:

Figura 5.1 - Arquitectura de implementação

Hardware: A implementação das soluções foi realizada numa máquina com um processador Intel Atom N270 a 1.6GHz e 3 Gbytes de memória RAM. Para a validação das soluções e avaliação das mesmas foi usada outra máquina com mais recursos, constituída por um processador Intel Core 2 Duo E6600 a 2.4GHz e com 4 Gbytes de memória RAM.

Sistema Operativo: As soluções foram implementadas, validadas e avaliadas em dois ambientes distintos. O primeiro constituído pelo sistema operativo Microsoft Windows 7 e o segundo, pela distribuição Ubuntu 10.10 do sistema operativo Linux.

Java *Framework*: Tanto a extensão da plataforma como a implementação dos protocolos de distribuição e estabelecimento de chaves foram implementados em código Java que é compilado sobre a forma de *bytecodes* que são depois interpretados pela máquina virtual do Java a correr sobre o sistema operativo.

Suite criptográfica: Para o fornecimento de primitivas e mecanismos de segurança, necessários à correcta implementação e avaliação dos protocolos, utilizou-se a colecção de APIs que compõem a suite Bouncy Castle [33] na sua versão 1.45 para o ambiente Java.

De entre as primitivas criptográficas simétricas fornecidas optou-se pelas recomendações da suite SPINS [24]. Para tal é usada a cifra em blocos RC5 no modo CTR (*counter*) devido à sua eficiência e baixos requisitos de memória o que a torna ideal para dispositivos de baixos recursos como sensores. Devido ao uso do modo CTR é necessário um *counter* para cada operação e, neste caso, volta-se a seguir a recomendação da suite SPINS: tanto o *cluster head* como um nó membro partilham um estado (*counter*) e incrementam o mesmo após cada operação mantendo o estado consistente. Para situações de uso de chaves em grupo (exemplo: fase de organização do protocolo LHA-SP) tal não é possível e os *counters* não poderão ser sincronizados da mesma maneira, visto que uma transmissão não será ouvida pela rede inteira. Nestes casos é adicionado o valor do *counter* usado às mensagens trocadas.

Finalmente, para o fornecimento de primitivas de autenticação (MAC) é usado o mecanismo CMAC com recurso à cifra por blocos RC5 e para cifras/decifras assimétricas (para os nós que o suportem) é usado RSA com chaves de 1024 bits.

5.2. Dimensão do trabalho

Partindo do núcleo base de simulação foram criadas novas classes e programadas diversas linhas de código de modo a estender o mesmo, com o objectivo de criar uma plataforma genérica de simulação capaz de simular a execução de diversos protocolos para redes de sensores sem fios aproximada ao funcionamento real sobre a pilha IEEE 802.15.4, sendo também capaz de extrair indicadores importantes da execução dos mesmos. A plataforma encontra-se descrita no capítulo 3 e a especificação dos protocolos estudados no capítulo 4. Em baixo é apresentada uma tabela que pretende realçar algumas métricas importantes do código implementado, que permitem aferir a dimensão do trabalho realizado e onde o núcleo base de simulação, integrado na plataforma final, representa cerca de 6,8% de todo o código.

	Núcleo base de simulação (jProwler)	**Plataforma final (+ núcleo base)**	**Protocolos**
Total de classes	17	113	95
Total de interfaces	1	9	1
Total de métodos	129	947	844
Total de linhas de código	976	8389	5847

Tabela 5.1 - Tabela comparativa da dimensão da solução implementada

6. Avaliação experimental

Neste capítulo descrevem-se os testes realizados e analisam-se os resultados obtidos, que serviram de base para a validação e comparação dos protocolos, assim como base de validação da plataforma sobre a qual estes foram implementados. Todos os testes foram realizados no ambiente de simulação descrito no capítulo 3 e onde foi dado uso a algumas das funcionalidades implementadas na plataforma.

Para se proceder às validações definiram-se parâmetros e condições dos testes de forma a manter a realização dos testes o mais objectiva possível. Os testes realizados dividem-se essencialmente em 5 tipos de critérios: cobertura da rede, latência na formação da mesma, gastos energéticos, fiabilidade na entrega de relatórios e impacto de ataques internos na operação dos protocolos, nomeadamente o impacto sobre os anteriores critérios. A avaliação destes critérios será dividida pelas duas grandes fases que compõem os protocolos: organização da rede e operação da mesma. Inicialmente serão apresentados os parâmetros e condições de realização dos testes e em seguida os resultados obtidos nas respectivas fases.

6.1. Parâmetros e condições de simulação

Nas condições da simulação foram simulados sensores do tipo Mica 2 Mote da Crossbow com raios de alcance de transmissão e recepção entre 30 e 50 metros. As características de comunicação dividem-se ainda na modelação MAC para comunicações IEEE 802.15.4 em modo sem coordenador e simulação de colisões rádio. A distribuição dos sensores encontra-se explicada individualmente para cada protocolo na secção abaixo e a *base station* foi sempre lançada no centro da rede pois foi onde foram obtidos melhores índices de cobertura em alguns testes prévios.

As parametrizações da plataforma passaram por usar filas de mensagens no envio de mensagens com atrasos na ordem dos 2 segundos para evitar colisões e garantir um débito

aproximado de uma mensagem a cada um segundo. Já as parametrizações usadas por defeito para cada protocolo foram as seguintes:

- **F-Leach:** 4 rondas para efeitos de média; 15% de probabilidade de um nó se auto-eleger *cluster head*;
- **SecLeach:** 4 rondas para efeitos de média; 15% de probabilidade de um nó se auto-eleger *cluster head*; *key rings* comuns (o impacto do esquema de Eschenauer foi estudado à parte);
- **LHA-SP:** 20 metros de distância entre anéis de níveis (ver secção 6.1.1); Protocolo *hello* desactivado para evitar colisões desnecessárias;
- ***Framework* de Bohge:** Entre 46% de nós encaminhadores face ao número de nós sensores de modo a fornecer um *backbone* capaz de encaminhar as mensagens dos mesmos até à *base station* através de vários caminhos;

6.1.1. Distribuição topológica

Um factor importante na avaliação experimental dos protocolos sobre a plataforma surge na distribuição dos nós participantes (de acordo com o número de sensores parametrizável) pela rede de modo a que esta seja minimamente coberta e para se poderem obter bons indicadores de avaliação experimental. A distribuição dos nós em cada protocolo segue uma das seguintes topologias de distribuição que foram implementadas:

Distribuição aleatória

Tanto no protocolo F-Leach como SecLeach os sensores são distribuídos uniformemente e de forma aleatória pela área de teste e as respectivas vizinhanças não são conhecidas previamente. Este é o modelo que mais vezes é usado na realidade e também o mais interessante de obter indicadores de consumo energético, fiabilidade, etc. Já na *framework* de Bohge os nós sensores e os nós encaminhadores são também distribuídos aleatoriamente.

Distribuição parcialmente aleatória baseada em níveis

Este tipo de distribuição foi especificamente implementada para o protocolo LHA-SP. Este protocolo tem em comum com os restantes a formação hierárquica da rede, mas é bastante

diferente na atribuição de níveis a cada nó, mais concretamente o momento em que o faz. Enquanto que os restantes protocolos definem uma rede de apenas 2 níveis e, no caso dos protocolos Leach, a atribuição ser feita de forma dinâmica e aleatória após a distribuição de nós, no LHA-SP existem vários níveis definidos e a atribuição é feita estaticamente antes da distribuição, o que limita bastante o conjunto de vizinhos com quem um nó qualquer possa formar *cluster*, reflectindo-se de forma acentuada nos índices de cobertura.

A distribuição adoptada foi então de variar o número de níveis modo a cobrir uma dada área. Para tal os níveis são particionados em áreas circulares (anéis), onde um nível de *h* engloba todo o raio do nível *h+1* mais um incremento que corresponde a uma distância parametrizável, garantindo que os mesmos conseguem comunicar com os nós de nível superior. Já a distribuição de nós pelos níveis é feita de forma aleatória e sempre garantindo que os níveis de maior área possuem um peso percentual maior de nós. A figura 6.1 exemplifica o modelo de distribuição descrito. Com este modelo de distribuição foi possível obter bons índices de cobertura topológica neste protocolo face a apenas boa cobertura física se a distribuição fosse totalmente aleatória numa rede plana.

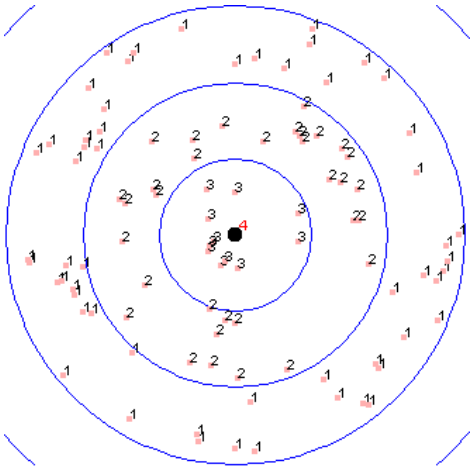

Figura 6.1 - Distribuição topológica do protocolo LHA-SP

6.1.2. Dimensão da rede

Devido às diferentes características dos protocolos e às limitações que cada um impõe sobre a vizinhança de nós, foi necessário achar uma dimensão da área de testes adequada para cada um destes em função do número de nós. Os valores obtidos são apresentados na tabela 6.1 e têm por

base um funcionamento real sobre a pilha IEEE 802.15.4 onde a cobertura estática, quer topológica quer do protocolo, foi maximizada e a cobertura dinâmica situou-se entre 75% e 80%, ou seja, a tabela pretende apresentar uma aproximação à área ideal para a referida dimensão da rede (número de nós) onde os protocolos conseguem obter boas taxas de cobertura seguindo as parametrizações já referidas. São também apresentadas aproximações para diferentes alcances de rádio (50m e 30m). Os campos da tabela representam a dimensão de um dos lados da área que é assumida como quadrada.

Num Nós:	**200**	**400**	**700**	**1000**	**1500**	**2000**	**3000**
F-Leach(50m)	286m	393m	518m	652m	813m	929m	1143m
F-Leach(30m)	172m	236m	311m	392m	488m	558m	686m
SecLeach(50m)	304m	411m	536m	679m	840m	965m	1215m
SecLeach(30m)	183m	247m	322m	408m	504m	579m	729m
LHA-SP(50m)	322m	429m	554m	679m	840m	947m	1161m
LHA-SP(30m)	193m	258m	333m	408m	504m	568m	697m
F. Bohge(50m)	293m	402m	522m	652m	813m	929m	1161m
F. Bohge(30m)	176m	242m	313m	392m	488m	558m	697m

Tabela 6.1 - Aproximação às dimensões de área ideais face à quantidade de nós

Os valores obtidos são uma aproximação a uma dimensão real de uma área onde os sensores deverão ser lançados para obter boas taxas de cobertura. Observou-se que alguns protocolos conseguem cobrir maior área que outros mas, de modo a comparar os protocolos nos critérios propostos, os valores óptimos de áreas que maximizam a cobertura dinâmica serão sempre os valores de referência para os restantes testes apresentados em função do número de nós, servindo esta tabela de referência. As parametrizações dos protocolos apresentadas anteriormente também foram ajustadas de modo a que a ordem de grandeza das dimensões das áreas fossem semelhantes entre protocolos. A título de exemplo a *framework* de Bohge conseguiria cobrir uma área bastante maior bastando para isso aumentar percentualmente o número de nós encaminhadores mas decidiu-se fixar este parâmetro em 46% pela razão apresentada.

6.2. Indicadores durante a fase de organização da rede

A fase de organização da rede define a topologia final da rede que servirá de referência para a fase de operação. Assim, torna-se importante analisar os critérios de cobertura da mesma, latência de formação e o custo necessário para atingir tal estado.

6.2.1. Cobertura

O primeiro critério avaliado está relacionado com as questões de cobertura da rede. A cobertura define o aproveitamento que é dado aos sensores lançados na rede e a área que é possível cobrir da mesma, tornando este critério num dos mais importantes para a análise da escalabilidade de um protocolo. Previamente procedeu-se à determinação da área ideal em que cada protocolo consegue cobrir com uma boa taxa de aproveitamento de nós, o qual resultou na tabela anteriormente apresentada. Assim, com a presente análise, pretende-se efectuar outro estudo: aferir o comportamento dos protocolos quando a área é muito pequena para uma larga quantidade de nós e o caso exactamente oposto.

Assim fixaram-se estrategicamente duas dimensões de áreas e nelas foram lançados um número incremental de sensores por protocolo de modo a prever os seus comportamentos quando sujeitos às condições reais de operação subjacente à pilha IEEE 802.15.4 e onde ocorrem colisões simuladas pelo modelo de colisões rádio do simulador. Os gráficos a seguir apresentados demonstram os resultados obtidos, onde os mesmos estão divididos em análise estática (a tracejado) e análise dinâmica.

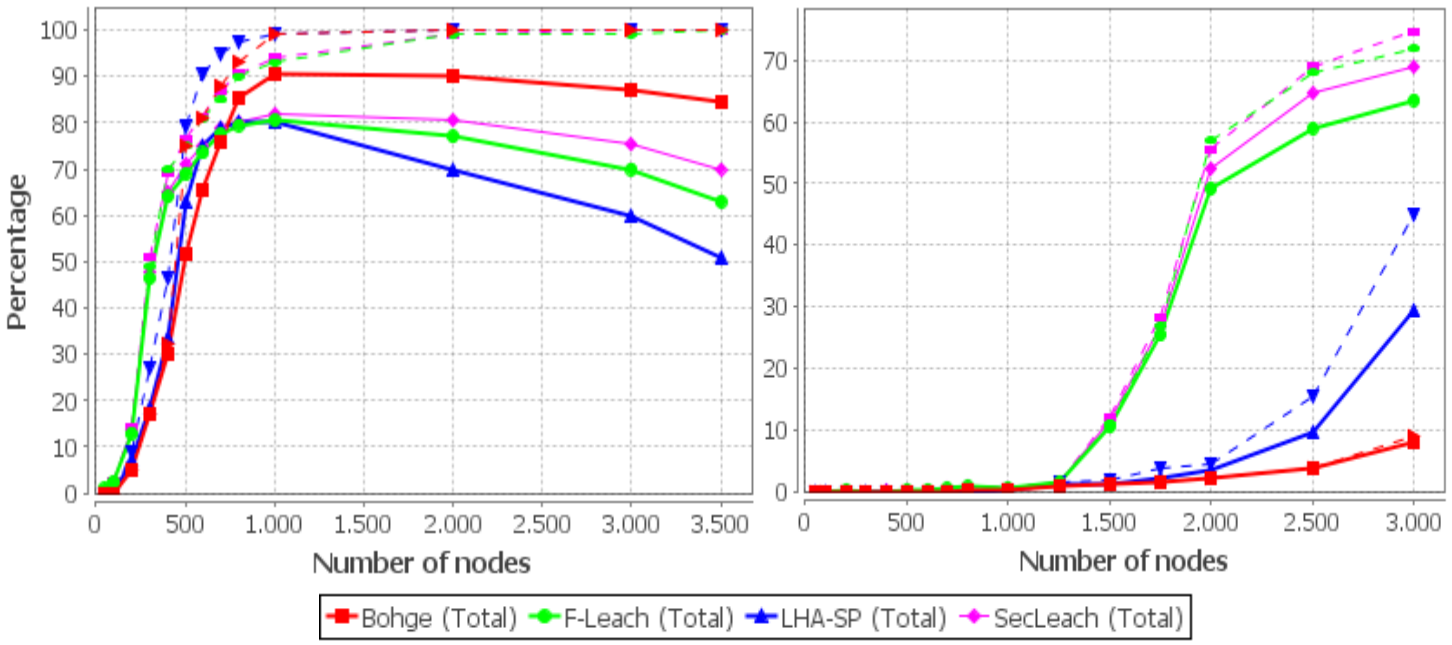

Figura 6.2 - Indicadores de cobertura numa área de 0,29km^2 (esquerda) e 2km^2 (direita)

Como se verifica para uma área de 0,29 km^2 (alcance de rádio dos sensores = 50m) os protocolos atingem a taxa óptima de cobertura (na ordem dos 80%) por volta do mesmo número de nós sendo que o protocolo LHA-SP é mais vulnerável às colisões de pacotes visto que ao aumentar a concentração de nós numa mesma área a taxa de cobertura desce mais acentuadamente neste. A razão para tal passa pela sequencialidade da formação de *clusters* entre níveis, face à paralelização como no SecLeach e F-Leach. Por outro lado a *framework* de Bohge resiste muito bem a altas concentrações de nós devido a todo o protocolo funcionar à base de *flooding* e portanto serem exploradas múltiplas rotas para todas as mensagens especificadas no protocolo. Já no F-Leach a exploração de múltiplas rotas é de facto benéfica para a autenticação dos *cluster heads* mas no final a cobertura sai sempre afectada pelas comunicações *single-hop* entre nós membros e os mesmos, assim como no SecLeach.

O gráfico da direita apresenta uma área de 2 km^2, muito grande para ser coberta por apenas 3000 nós sensores, e demonstra a mais rápida convergência para uma boa taxa de cobertura nos protocolos baseados no Leach.

6.2.1.1. Impacto do esquema de Eschenauer

Com esta análise pretende-se estudar o impacto das parametrizações relativas ao esquema de pré-distribuição de Eschenauer quando o protocolo SecLeach se encontra sujeito a condições reais de simulação. Como ponto de partida considerou-se a mesma área do teste de cobertura anterior e obtiveram-se novos indicadores quando a probabilidade de dois nós partilharem uma chave deixa de ser total (p=100%). No gráfico seguinte podem-se observar os resultados obtidos para uma dimensão da *key pool* de 10 mil chaves e *key rings* de 120 (p=76%) e 80 (p 47%).

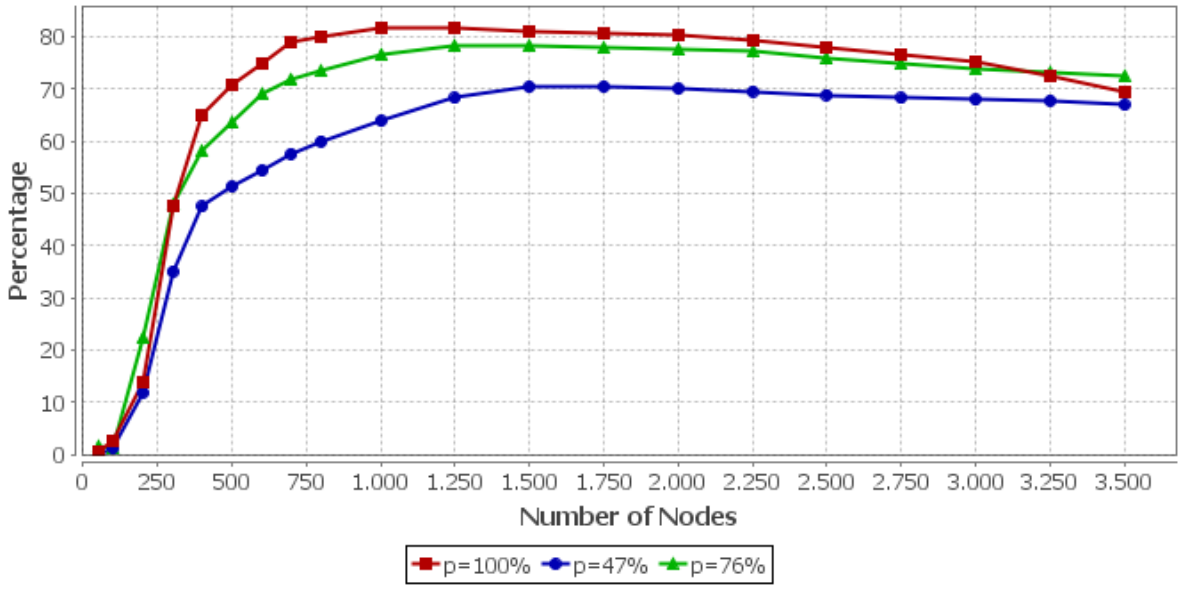

Figura 6.3 - Indicadores do esquema de Eschenauer face à cobertura

Os resultados obtidos demonstram que, quer numa área com poucos nós quer numa área com o número ideal de nós, a taxa de cobertura é sempre superior quanto maior for a dimensão do *key ring*. No entanto onde o modelo teórico falha é quando a concentração de nós é muito elevada para uma área pequena, levando ao aparecimento de muitas colisões nas comunicações. Nestes casos a tendência é exactamente a oposta do modelo teórico, ou seja, o facto de nós vizinhos começarem a partilhar cada vez menos chaves acaba por favorecer a taxa de cobertura visto que haverão menos comunicações (pela especificação do protocolo) e, consequentemente, menos colisões. Para a área estudada e dimensão da *key pool* definida verifica-se que a partir de 1250 nós (média de 20 vizinhos para cada) é quase indiferente um nó ter uma dimensão do *key ring* de 10 mil chaves ou apenas 120, sendo que a partir de 3250 nós (média de 50 vizinhos para cada) é mais vantajoso, quer em termos de segurança quer em taxa de cobertura, ter mesmo apenas 120.

6.2.2. Latência

A latência durante a fase de organização da rede define o tempo expectável que a mesma demora a organizar-se de modo a prosseguir para a fase de operação. De modo a medir e comparar a latência entre protocolos foram lançados números incrementais de nós de modo a medir o impacto que o aumento da dimensão da rede tem neste critério. As áreas onde os nós foram lançados foram as já referenciadas no início do capítulo onde os protocolos conseguem obter respectivamente taxas de cobertura entre 75% e 80% de modo a que essas taxas sejam idênticas entre os mesmos e poderem ser, portanto, comparáveis. Os tempos medidos referem o instante em que o último nó formou *cluster* com outro ou, no caso da *framework* de Bohge, se autenticou perante a *base station* tendo recebido a resposta desta. No caso específico do protocolo F-Leach foi atribuído um tempo de espera de anúncios na *base station* suficientemente grande para não se perderem anúncios devido a este parâmetro e os resultados apresentados ignoram a diferença de tempo entre a recepção do último anúncio e o instante em que a lista de *cluster heads* válidos é difundida. Com isto pretende-se apresentar a latência mínima assumindo que o tempo se encontra bem parametrizado para a dimensão da rede.

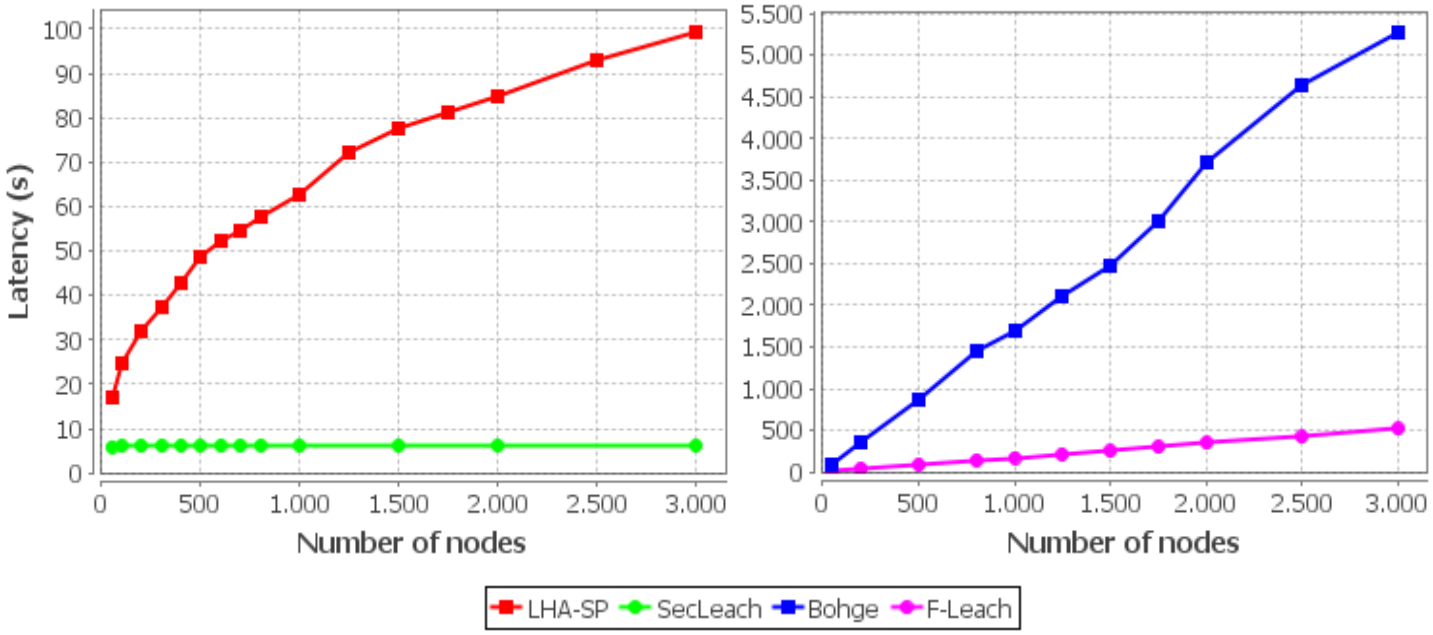

Figura 6.4 - Indicadores de latência (fase de organização) face à dimensão da rede

O gráfico da esquerda apresenta os resultados obtidos para protocolos que não recorrem ao *flooding* de dados visto que tal algoritmo de encaminhamento aumenta a latência exponencialmente. Já o gráfico da direita compara os dois protocolos que usam *flooding*.

Dos resultados obtidos no gráfico da esquerda percebe-se que o protocolo SecLeach leva clara vantagem na latência de estabilização da rede ao formar *clusters* com visões parciais da rede. A desvantagem imposta neste comportamento é que não há garantia de haver mais tarde caminho para a *base station* e os nós sensores poderem enviar dados que nunca chegarão ao destino. Por outro lado o protocolo LHA-SP apresenta sempre uma latência incremental à medida que a dimensão da rede aumenta (implicitamente o número de níveis também) visto que o protocolo é sequencial entre níveis. Os resultados obtidos para este último foram também obtidos parametrizando o protocolo para responder imediatamente a anúncios de nós de nível superior, ou seja, em condições ideais onde um nó introduz um tempo de espera para processar vários anúncios a latência seria ainda superior.

Analisando o gráfico à direita, verifica-se que a latência introduzida pelo *flooding* de mensagens quando sujeitas a operações reais sobre a pilha IEEE 802.15.4 é muito grande, especialmente na *framework* de Bohge onde tanto os pedidos de autenticação como as respostas da *base station* são difundidas por toda a rede, causando imensas colisões e aumentando consideravelmente as filas de mensagens dos nós. Observa-se também um aumento linear de latência neste último protocolo à medida que o número de nós cresce.

6.2.3. Consumo energético

A avaliação do consumo energético é relevante de forma a medir o impacto que a fase de organização tem nas baterias limitadas dos sensores permitindo analisar se esse impacto está a afectar em demasia o tempo de vida da rede. É também conveniente avaliar o gasto energético individual introduzido pelas várias operações dos protocolos para identificar onde está o maior gasto, assim como distinguir o consumo efectuado pelas diferentes classes de nós, pré-determinadas ou determinadas probabilisticamente.

Os gastos energéticos durante o processo de organização da rede envolvem todas as comunicações e computações realizadas durante a fase de auto-organização da rede, seguindo o modelo de consumo energético detalhado no capítulo 3.2.3.4. Os protocolos foram comparados ao nível da dimensão rede onde a taxa de cobertura alcançada foi na ordem dos 75%-80%, ou

seja, usando as dimensões e parametrizações de referência anteriormente estudadas. À semelhança do teste de latência serão apresentados resultados separados para protocolos que recorram ou não ao *flooding* de dados. Os resultados obtidos reflectem o consumo de toda a rede.

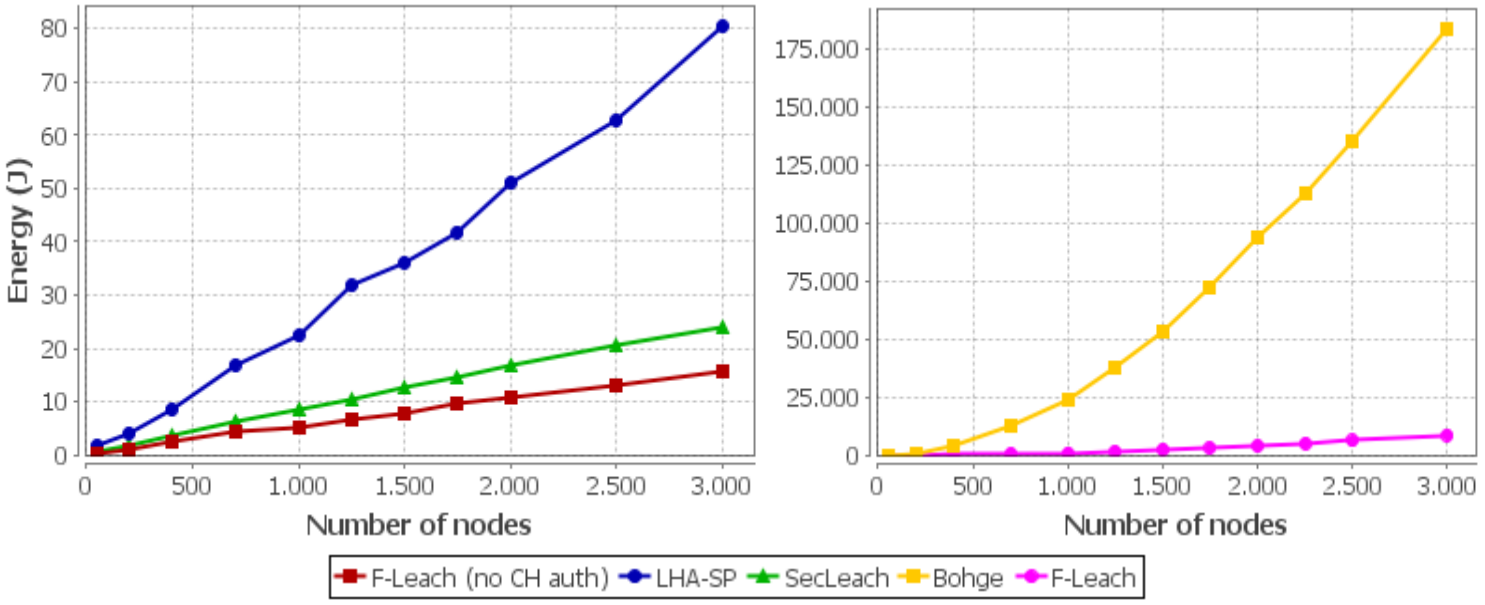

Figura 6.5 - Indicadores de consumo energético (fase de organização) face à dimensão da rede

Os resultados demonstram o elevado consumo introduzido pelo *flooding* de dados, especialmente na *framework* de Bohge onde tanto os pedidos de autenticação como as respostas são difundidas pela rede, ou seja, o equivalente ao dobro do numero de nós em termos de mensagens. Já no F-Leach o uso de *flooding* aumenta também o consumo mas numa escala muito inferior devido a serem muito menos nós a autenticarem-se (apenas *cluster heads*) e a resposta da *base station* ser única e não necessitar de encaminhamento.

Virando a atenção para os protocolos que não necessitam de encaminhamento *multi-hop* de dados, verifica-se que o protocolo LHA-SP apresenta um consumo consideravelmente superior ao SecLeach e a razão não é tanto pelo tamanho das mensagens, mas sim pela quantidade das mesmas. Ambos os protocolos apresentam uma especificação de autenticação de um nó membro perante um *cluster head* em 3 passos com mensagens de tamanho idêntico, mas o protocolo LHA-SP especifica que o nó de nível superior responde individualmente a cada pedido de junção enquanto que na especificação do SecLeach o *cluster head* envia apenas uma mensagem de resposta para todos os nós membros. Adicionalmente foi também estudada a hipótese de, no protocolo F-Leach, ignorar a autenticação dos *cluster heads* perante a *base station* e a formação

de *clusters* prosseguir sem este passo. O resultado é um impacto energético ainda inferior ao SecLeach devido ao tamanho mais reduzido das mensagens.

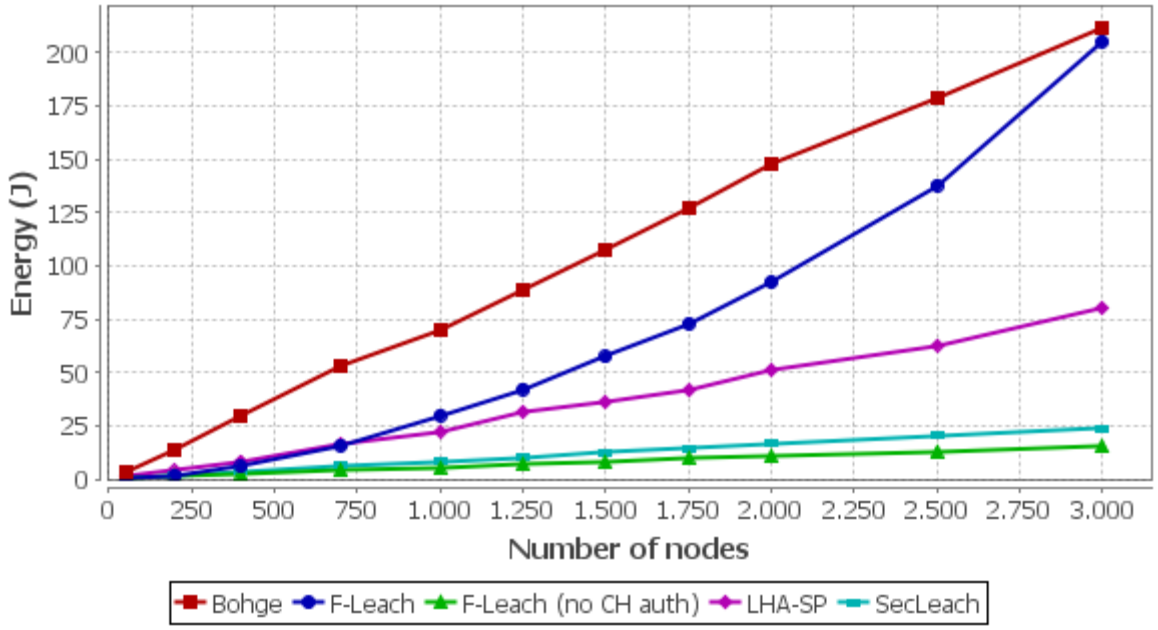

Figura 6.6 - Indicadores de consumo energético (fase de organização) sem encaminhamento multi-hop face à dimensão da rede

Em seguida foi analisado o impacto energético dos protocolos descontando os custos do encaminhamento por *flooding*. Para a análise deste gráfico é importante referir que, aos protocolos F-Leach e *framework* de Bohge, ainda é preciso então somar o custo de encaminhar as mensagens até à *base station*. Assim, tirando os custos de encaminhamento da equação, verifica-se que, para baixo de 700 nós, o protocolo F-Leach apresenta vantagens no consumo energético face ao LHA-SP. Outra análise importante é a de que, pouco acima dos 3000 nós, o protocolo F-Leach apresenta custos superiores à *framework* de Bohge com as suas mensagens de longas dimensões. A razão para este facto deve-se à divulgação da lista de *cluster heads* autenticados, que fica cada vez maior à medida que a rede cresce (4 bytes por identificador). Esta é uma conclusão que não poderia ter sido tirada dos gráficos anteriores devido ao alto consumo de encaminhamento necessário na *framework* de Bohge que ofuscava o crescimento energético acentuado no F-Leach.

6.3. Indicadores durante a fase de operação

Após a formação da topologia final da rede os nós entram em modo de operação onde seguem a especificação do protocolo para enviar relatórios de eventos até à *base station*. Como tal torna-se necessário avaliar os critérios de fiabilidade, latência e impacto energético na rede da entrega dos mesmos. Todos os testes efectuados foram realizados apenas sobre nós que, após a fase de organização, ficaram totalmente cobertos pelo protocolo pois torna-se pouco interessante extrair indicadores de nós que nunca conseguirão alcançar a *base station* e onde a fiabilidade é automaticamente 0%.

Para efeitos de clareza neste capítulo, consideram-se como correctamente entregues todos os relatórios que sejam recebidos pela *base station*, sejam eles compostos por uma ou mais mensagens. Assim, no protocolo F-Leach, um relatório tem sucesso na entrega se ambas as mensagens que o compõem foram recebidas pela *base station*, ou seja, nenhuma delas se perdeu devido a colisões. Já no caso específico da *framework* de Bohge a mensagem de confirmação (*rnplusone*) não é considerada para efeitos de sucesso na entrega do relatório em questão.

6.3.1. Fiabilidade

A taxa de fiabilidade de entrega de mensagens numa RSSF indica qual é a probabilidade de entrega de qualquer relatório, sendo por isso um fenómeno interessante de observar, não só para validar os protocolos, como para estabelecer uma base comparativa entre eles e em futuros testes que meçam essa mesma fiabilidade quando a rede se encontra sob ataque. O parâmetro que mais influencia a fiabilidade da rede está relacionado com a quantidade de mensagens em circulação, pois quanto maior for esse número, maior será a probabilidade de existirem colisões no nível físico, impedindo a sua correcta recepção.

Para a obtenção de indicadores de fiabilidade foram realizados dois grandes testes aos quais foram submetidos os protocolos. No primeiro teste escolheram-se aleatoriamente 20% de nós cobertos para gerarem um evento cada, produzirem os relatórios e depois então procederem com o envio dos mesmos para a *base station*. O segundo teste baseou-se na mesma ideia só que o número de nós aumentou para 40% de modo a medir o impacto que os protocolos têm na fiabilidade quando o número de mensagens em circulação aumenta (para o dobro neste caso).

Com a análise individual de cada gráfico também foi possível avaliar o presente critério em cada protocolo, quando os mesmos são sujeitos a redes de cada vez maior escala.

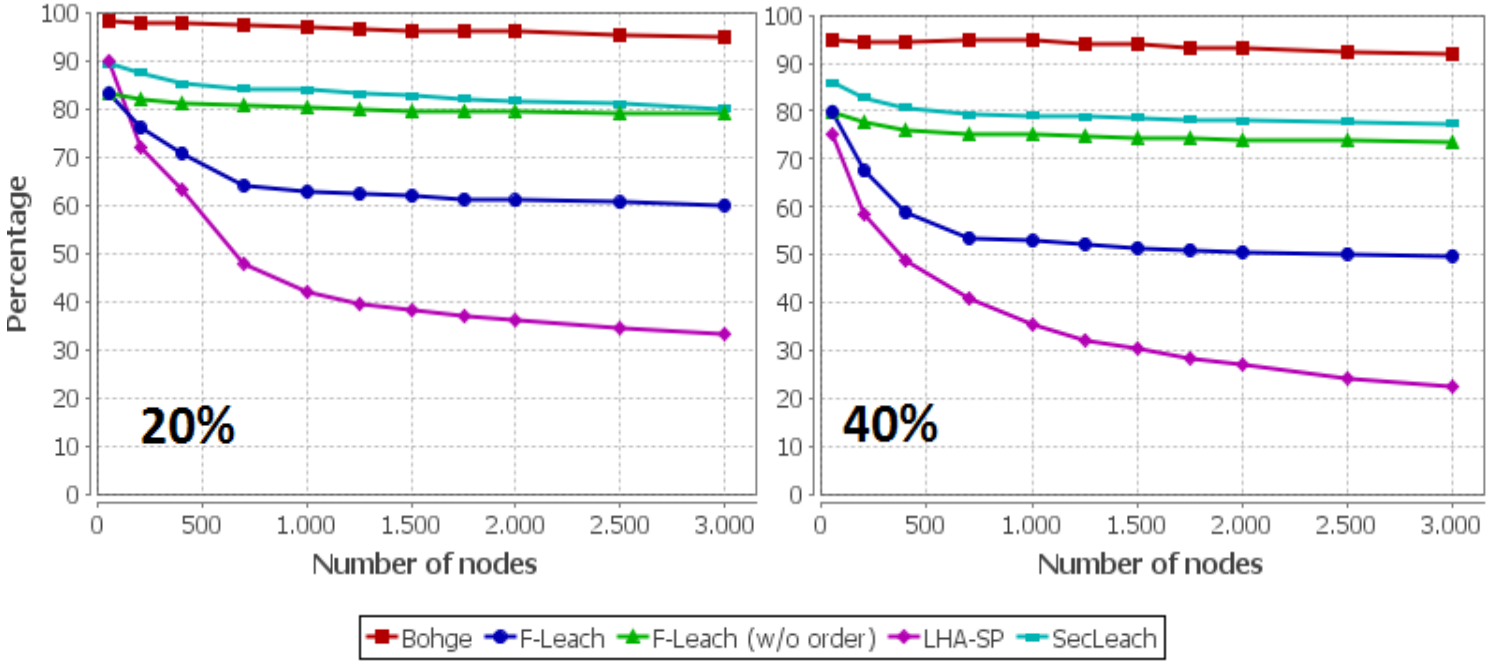

Figura 6.7 - Indicadores de fiabilidade face à dimensão da rede com geração de eventos em 20% da rede (esquerda) e 40% (direita)

Em primeiro lugar a especificação original do F-Leach previa transmissões *single-hop* entre *cluster heads* e *base station* logo, seguindo a especificação, ao enviar um relatório e em seguida a colecção de *macs* dos nós que participaram na construção do mesmo (*mac_array*) subentende-se uma ordem imposta na recepção também. Ora com a adição de encaminhamento *multi-hop* (*flooding*) tal ordem não pode ser garantida e por isso os testes ao protocolo foram divididos em 2 casos. No primeiro ("F-Leach") a ordem tem de ser preservada para a entrega de um relatório ser considerada como um sucesso e no segundo ("F-Leach w/o order") a ordem não interessa desde que as duas mensagens referentes à entrega sejam recebidas pela *base station*.

Passando à análise dos resultados, a primeira conclusão que se tira é de que a fiabilidade da entrega de mensagens é sempre superior nos protocolos que usam *flooding* como método de encaminhamento, especialmente na *framework* de Bohge onde, desde a primeira transmissão, a mensagem é replicada por diversas rotas, ao contrário dos protocolos baseados no Leach onde se observa uma redução de fiabilidade devido às comunicações *single-hop* entre nós membros e *cluster heads* respectivos. No protocolo F-Leach o critério de fiabilidade apresenta-se

ligeiramente inferior devido a terem de ser encaminhadas duas mensagens por *cluster head*, face a apenas uma como no SecLeach e consideravelmente inferior se existir a necessidade de ordem na entrega das mesmas. Já o protocolo LHA-SP é, de todos, o que apresenta indicadores de fiabilidade mais baixos devido a todas as comunicações serem *single-hop* e o aumento da dimensão da rede implica um aumento no número de *hops*, o que por sua vez implica uma maior probabilidade de colisões pelo caminho entre níveis até à *base station*.

Ao considerar um aumento no número de eventos gerados (de 20% para 40%) observa-se que, mais uma vez, os protocolos baseados em *flooding* respondem bem, sendo os índices de fiabilidade apenas ligeiramente afectados nestes. Por outro lado o número crescente de colisões geradas pelo aumento do número de mensagens na rede é muito mais prejudicial para o protocolo LHA-SP, onde o número de *hops* se mantém mas as colisões aumentam e, existindo apenas um caminho por relatório, a fiabilidade sai consideravelmente afectada.

6.3.2. Latência

A latência durante a fase de operação define o tempo expectável que um relatório de um evento demora a ser entregue à *base station* seguindo a especificação do protocolo. De modo a comparar este critério entre os protocolos foram gerados eventos em 20% dos nós e em seguida foram extraídos, dos relatórios que efectivamente chegaram à *base station*, as diferenças de tempos entre a geração dos mesmos e as respectivas entregas. No final foram calculadas médias para as diversas entregas, assim como os respectivos desvios padrão (a tracejado no gráfico). Com esta análise pretende-se principalmente saber com que ordem de grandeza média contar na entrega de relatórios já que na realidade a latência depende muito do número de *hops*, ou seja, nós mais distantes terão uma latência maior. No entanto, sendo a escolha de nós aleatória, será então possível obter uma boa aproximação a uma ordem de grandeza média.

No caso específico dos protocolos baseados no Leach a latência é também directamente afectada pela parametrização da duração de cada *slot* TDMA visto que cada nó terá de esperar pela sua vez de enviar o relatório para o *cluster head*. De modo a que este tempo de espera, introduzido em cada nó, influencie o menos possível na latência de entrega do relatório final a duração de cada *slot* foi parametrizada com o tempo expectável de envio de uma mensagem, ou seja, cada nó terá apenas o tempo necessário de enviar o seu relatório e não mais.

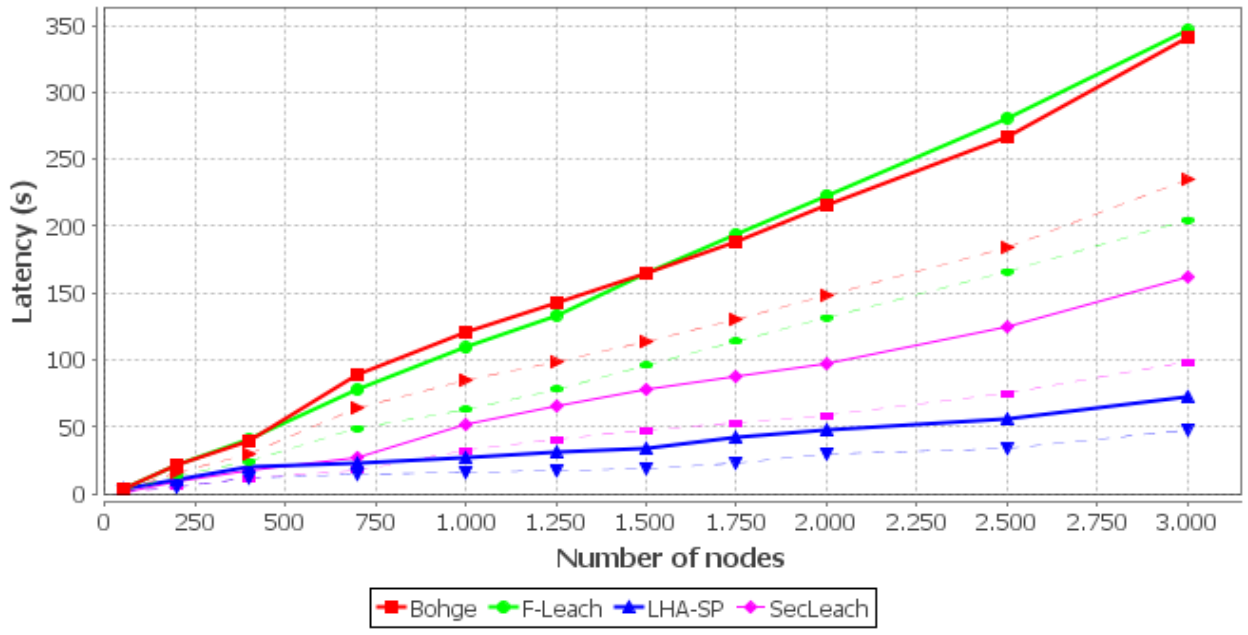

Figura 6.8 - Indicadores de latência (fase de operação) face à dimensão da rede

Observa-se pelos resultados que o protocolo LHA-SP consegue obter latências inferiores aos restantes protocolos devido a não usar *flooding* para o encaminhamento de dados e a própria topologia da rede apresentar um caminho muito directo, desde os nós sensores até à *base station*, traduzindo-se também num menor número de *hops*. Por outro lado, os restantes protocolos apresentam latências bastante superiores que começam a ser bem visíveis a partir de dimensões da rede na ordem dos 400 nós.

Entre os protocolos baseados no Leach percebe-se que a latência é sempre superior no F-Leach, normalmente um pouco acima do dobro, devido à divisão do relatório em duas mensagens onde ambas terão de ser encaminhadas separadamente, aumentando as filas de mensagens nos nós que participam no encaminhamento. No entanto foi também observado que o desvio padrão é quase idêntico, em termos percentuais, nestes dois protocolos, situando-se entre os 59% e os 61% do valor da latência em cada resultado obtido. Finalmente a *framework* de Bohge apresenta uma latência na mesma ordem de grandeza do F-Leach, o que à primeira vista parece contraditório visto que apenas uma mensagem tem de ser encaminhada. A razão passa por haverem menos nós capazes de encaminhar os relatórios até à *base station* (no F-Leach a rede toda encaminha), tornando a exploração de caminhos mais limitada e aumentando em média o número de *hops* e as filas de mensagens dos nós encaminhadores. Posteriormente a *base station* difunde também notificações de sucesso das entregas, o que agrava ainda mais a situação e,

razão também pela qual, o desvio padrão é o maior observado em todos os protocolos, que em termos percentuais face à latência obtida corresponde a uma ordem de grandeza entre os 69% e 71%.

6.3.3. Consumo energético

A avaliação do consumo energético durante a fase de operação envolve todas as comunicações e computações efectuadas para encaminhar as várias mensagens geradas durante esta fase desde qualquer nó da rede até à *base station*. Para se proceder a essa avaliação foram gerados eventos numa percentagem da população coberta após a fase de organização e mediu-se o consumo total da rede que teve de dar encaminhamento aos relatórios dos nós em questão. Definiu-se então essa percentagem como 20% da rede, seleccionando os nós aleatoriamente à semelhança do teste de latência, e mediram-se os consumos para um número incremental de nós que constituem a rede de modo a medir o impacto no consumo que um crescimento da mesma implica em cada protocolo.

Como nota adicional a dimensão de referência dos dados enviados por cada nó sensor foi de 4 bytes (valor inteiro), o qual foi somado ao *overhead* já introduzido pelos campos adicionais das mensagens segundo a especificação dos protocolos.

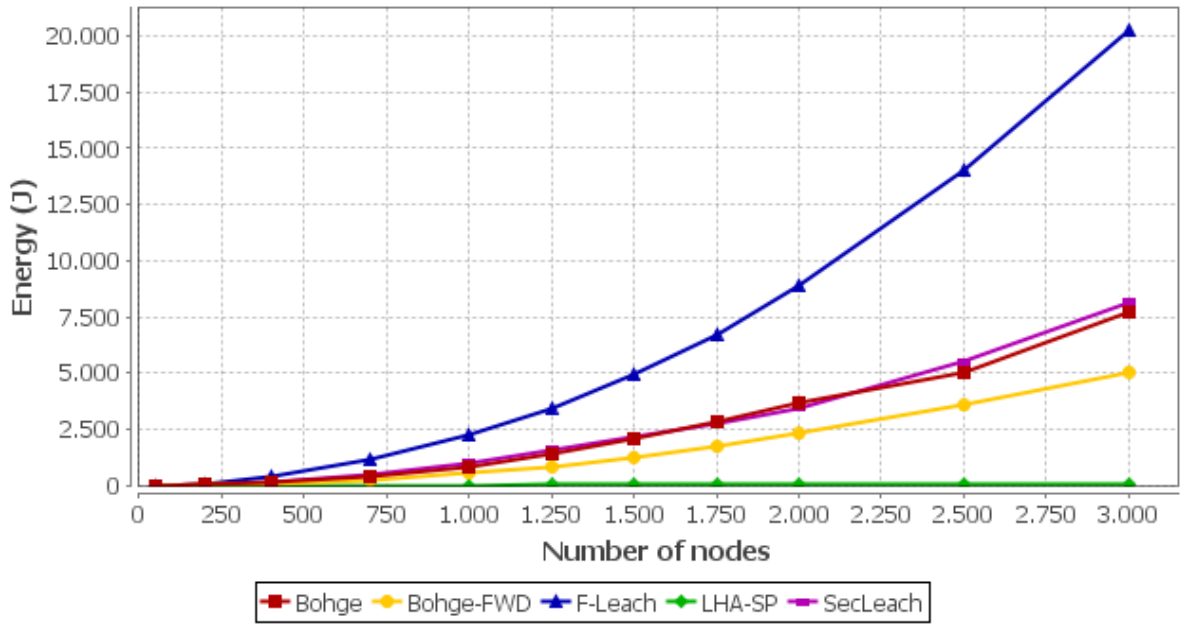

Figura 6.9 - Indicadores de consumo energético (fase de operação) face à dimensão da rede

De todos os protocolos o que menos consumo energético gerou durante a fase de operação foi obviamente o LHA-SP devido a usar uma solução de encaminhamento bastante directa até à *base station*, onde o número de *hops* é fixo em cada nível e as próprias mensagens têm tamanho reduzido (9 bytes). A aproximação deste protocolo implicou um maior consumo de processamento em cada nó (cifra e decifra) o qual é, no entanto, insignificante ao ser comparado com o custo observado dos envios e recepções das mensagens.

No lado oposto, observou-se um consumo energético bastante elevado no protocolo F-Leach, o que vai de encontro à especificação implementada que prevê o uso de *flooding* para encaminhar um conjunto de duas mensagens (por relatório) de tamanho considerável até à *base station*. De entre as duas mensagens destaca-se a segunda que serve o propósito de autenticar os nós sensores perante a *base station* de modo a que os seus relatórios sejam considerados pela mesma. Este mecanismo causa um aumento considerável da mensagem final (*array* de *macs*) e constitui a principal razão do consumo elevado nesta fase. Em contrapartida o protocolo SecLeach apresenta um consumo bastante inferior ao F-Leach devido a ter garantido a autenticação dos nós sensores durante a fase de organização da rede, tendo pago o custo adicional do *overhead* introduzido pelas mensagens durante essa fase, que é muito baixo comparado com a alternativa de gerar *mac arrays* durante a fase de operação. Assim, é fácil argumentar que a introdução de um mecanismo de autenticação entre nós membros e respectivos *cluster heads* com recurso a chaves aleatoriamente pré-distribuídas (esquema de Eschenauer por exemplo) seria benéfico, em termos energéticos, de se introduzir no F-Leach.

Por fim, a *framework* de Bohge apresenta um consumo bastante idêntico ao protocolo SecLeach, o que parece contraditório visto que as mensagens de relatório neste último (~21 bytes) são inferiores ao primeiro (~25 bytes) e o próprio SecLeach não prevê nenhum mecanismo de resposta que agrave ainda mais o consumo. A razão para tal prende-se no facto de que, no SecLeach, todos os nós da rede são encaminhadores de pacotes enquanto que na *framework* de Bohge apenas alguns nós estão encabidos de tal tarefa e os mesmos não são contabilizados no consumo geral dos nós sensores (porque não pertencem a esta classe), tornando a comparação um pouco injusta quando os dados são encaminhados por *flooding*. Assim, foram também tirados indicadores energéticos para os nós encaminhadores na *framework* de Bohge que, ao somar aos indicadores dos nós sensores, se obtêm custos energéticos

consideravelmente superiores à fase de operação do SecLeach e, no entanto, ainda inferiores à mesma fase no F-Leach.

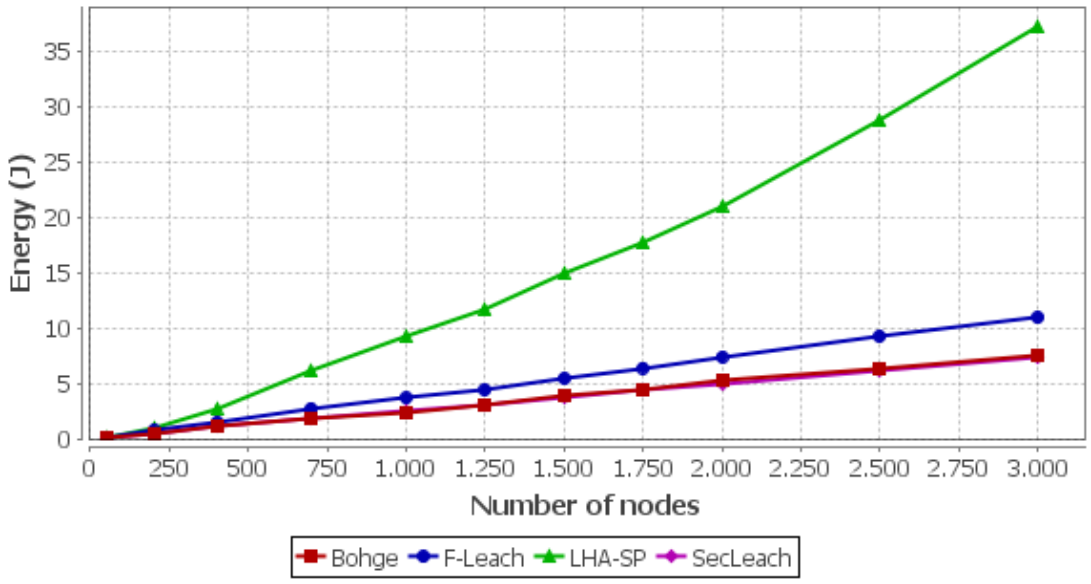

Figura 6.10 - Indicadores de consumo energético (fase de operação) sem encaminhamento multi-hop face à dimensão da rede

À semelhança do teste de consumo energético na fase de organização da rede, foram também extraídos indicadores durante a fase de operação descontando os custos energéticos que envolvem o encaminhamento por *flooding*. Em primeira análise dá-se uma melhor visão comparativa entre o LHA-SP e o resto dos protocolos, onde os indicadores deste primeiro já contemplam o encaminhamento dos relatórios sendo que no gráfico anterior eram completamente ofuscados pelos custos elevados dos restantes protocolos. A conclusão a tirar desta visão é que tanto o F-Leach, como o SecLeach e a *framework* de Bohge estão em condições de superar o LHA-SP em termos de consumos durante a fase de operação se usarem outra abordagem de encaminhamento mais eficiente em termos energéticos, o que já foge do escopo da presente dissertação.

Pelos resultados obtidos também se pode observar que os protocolos baseados no Leach têm consumos idênticos à *framework* de Bohge (no caso do F-Leach um pouco acima) mas levam vantagem no número de mensagens que têm de ser encaminhadas devido ao agrupamento de relatórios nos *cluster heads*, resultando também num menor custo de encaminhamento.

6.4. Indicadores face a ataques

Por fim foram retirados indicadores em situações em que a rede se encontra sob ataque. Para o âmbito dos testes realizados apenas se consideraram ataques internos, pois os ataques externos estão à partida protegidos pelas garantias de autenticidade e por vezes confidencialidade dos protocolos. A única excepção provém do SecLeach como já foi explicada anteriormente e portanto não será alvo de análise devido a não ser possível fazer comparação com os restantes protocolos.

Para a extracção de indicadores foram então simulados ataques que seguem o padrão estudado no modelo de adversário do capítulo 2. Nem todos os ataques são aplicáveis a certos protocolos pelo que os indicadores extraídos apenas representam aqueles que são vulneráveis.

6.4.1. Sinkhole

Na simulação deste ataque apenas foram considerados os protocolos SecLeach, F-Leach e LHA-SP pois são os únicos onde a aplicação desta topologia de ataque é válida. O objectivo do atacante é de atrair tráfego para si de maneira a controlar o maior número possível de fluxos de mensagens podendo depois dar o tratamento que bem entender aos mesmos e, no pior caso, descartar as mesmas de modo a causar impacto na cobertura e fiabilidade do protocolo.

Nos protocolos baseados no Leach a simulação do ataque passou por, ainda na fase de organização da rede, o atacante capturar uma percentagem de nós aleatoriamente e forçar a decisão de se tornarem *cluster heads* de modo a atraírem o tráfego de nós membros vizinhos que decidam formar *cluster* com estes. Já no LHA-SP a única possibilidade do atacante é capturar aleatoriamente sensores e continuar com a execução do protocolo normalmente nestes, comprometendo todos os nós abaixo na hierarquia de níveis que tenham formado *cluster* com estes e controlando então todas as mensagens que fluam na mesma.

Considerou-se como estudos de interesse nesta classe de ataque a análise da cobertura da rede após a fase de organização, assim como a percentagem da mesma que ficou comprometida no final da mesma fase. Adicionalmente verificou-se também a percentagem de mensagens que um atacante consegue controlar durante a fase de operação sendo portanto um indicador de referência para a eficácia do ataque.

6.4.1.1. Indicadores de cobertura

No gráfico abaixo é representado o impacto na cobertura quando o atacante captura, aleatoriamente, 5% e 15% dos nós da rede durante a fase de organização. A tracejado encontra-se discriminada a percentagem da mesma que ficou directamente comprometida, ou seja, todos os nós que foram capturados mais aqueles que, apesar de não o terem sido, dependem pela especificação do protocolo, dos directamente capturados para fazer chegar os seus relatórios até à *base station*. No SecLeach e F-Leach um nó é considerado comprometido quando forma *cluster* com um nó capturado pelo atacante e no LHA-SP consideram-se comprometidos todos aqueles que, algures no caminho da hierarquia de níveis até à *base station* formada durante a fase de organização, encontram um nó capturado que terá controlo sobre a mensagem.

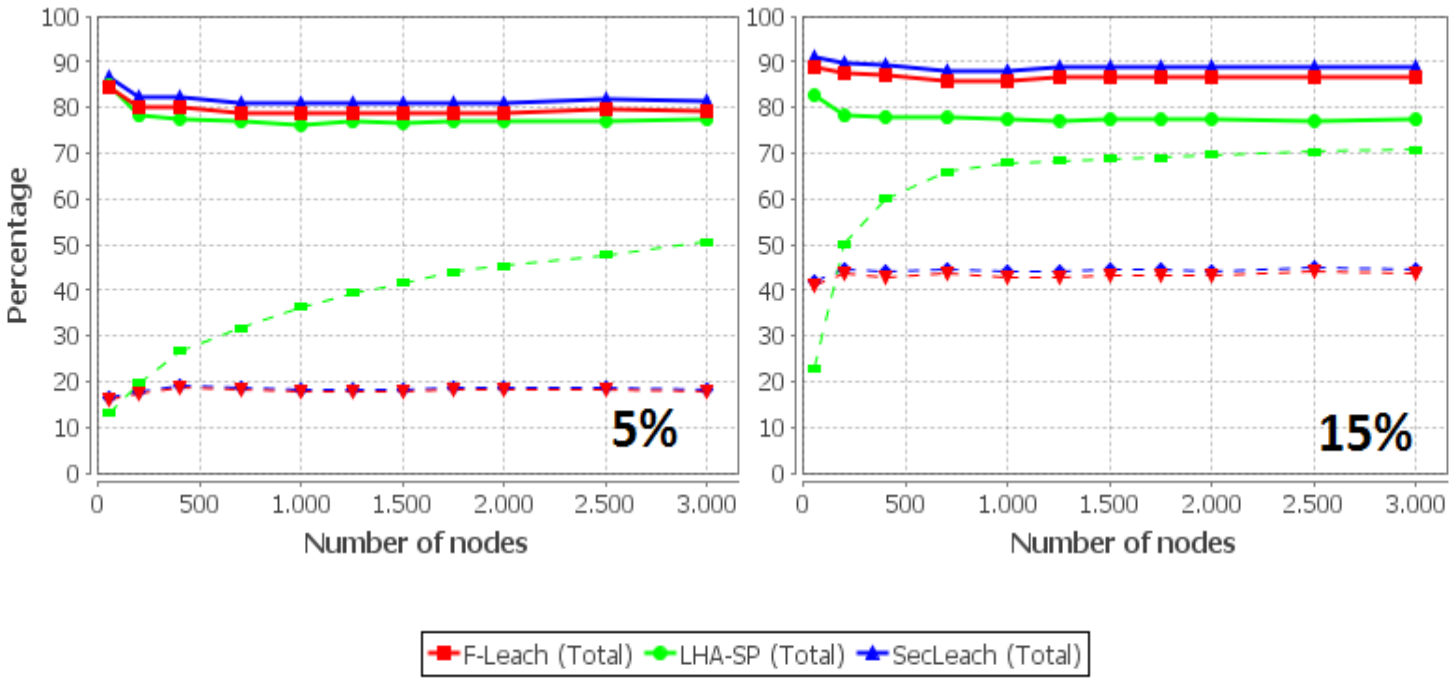

Figura 6.11 - Indicadores de cobertura na presença de ataques sinkhole

À partida pode-se observar um facto interessante: nos protocolos baseados no Leach a captura de nós segundo um ataque *sinkhole* melhora a cobertura da rede, onde sem captura de nós se obtinham resultados na ordem dos 75%-80% para as áreas de referência. Isto deve-se ao facto de o atacante forçar os nós capturados a tornarem-se *cluster heads*, o que aumenta o número destes distribuídos pela rede e, consequentemente, a cobertura da mesma visto que o objectivo do

atacante não é descartar mensagens (*blackhole*) mas apenas ganhar controlo sobre uma percentagem da rede. Já no LHA-SP a cobertura fica exactamente a mesma visto que o atacante apenas ganha controlo sobre certos nós e não origina novos *links* e/ou *clusters* como nos anteriores protocolos.

Pelo gráfico pode-se também observar que, com 5% e 15% de nós capturados, um atacante consegue comprometer em média 19% e 44% da rede, respectivamente, nos protocolos baseados no Leach e uma percentagem crescente no LHA-SP. A razão deste crescimento reside no facto que se um nó capturar um nó de nível alto vai conseguir comprometer todos os nós de nível inferior que formem *clusters* sequenciais na hierarquia de níveis o que, em termos percentuais, se reflectem em mais nós quando existem mais níveis. Nos restantes protocolos tal não se verifica visto que a formação de *clusters* é paralela entre várias zonas da rede. Assim, no LHA-SP, quanto maior for a percentagem de nós capturados, maior será o crescimento da percentagem de nós comprometidos face ao crescimento da rede.

Adicionalmente, tendo-se verificado uma taxa de cobertura idêntica para as várias dimensões da rede, foram seleccionadas duas das mesmas (1000 nós e 3000 nós respectivamente) para servirem de referência a um estudo que apresenta uma visão da cobertura da rede (e percentagem da mesma comprometida) face ao incremento de nós capturados pelo atacante. Pelos resultados observa-se, de forma mais explícita, o crescimento da cobertura face à percentagem de nós capturados e, consequentemente, o crescimento do número de nós comprometidos que é mais acentuado no LHA-SP, especialmente quando a rede tem maiores dimensões.

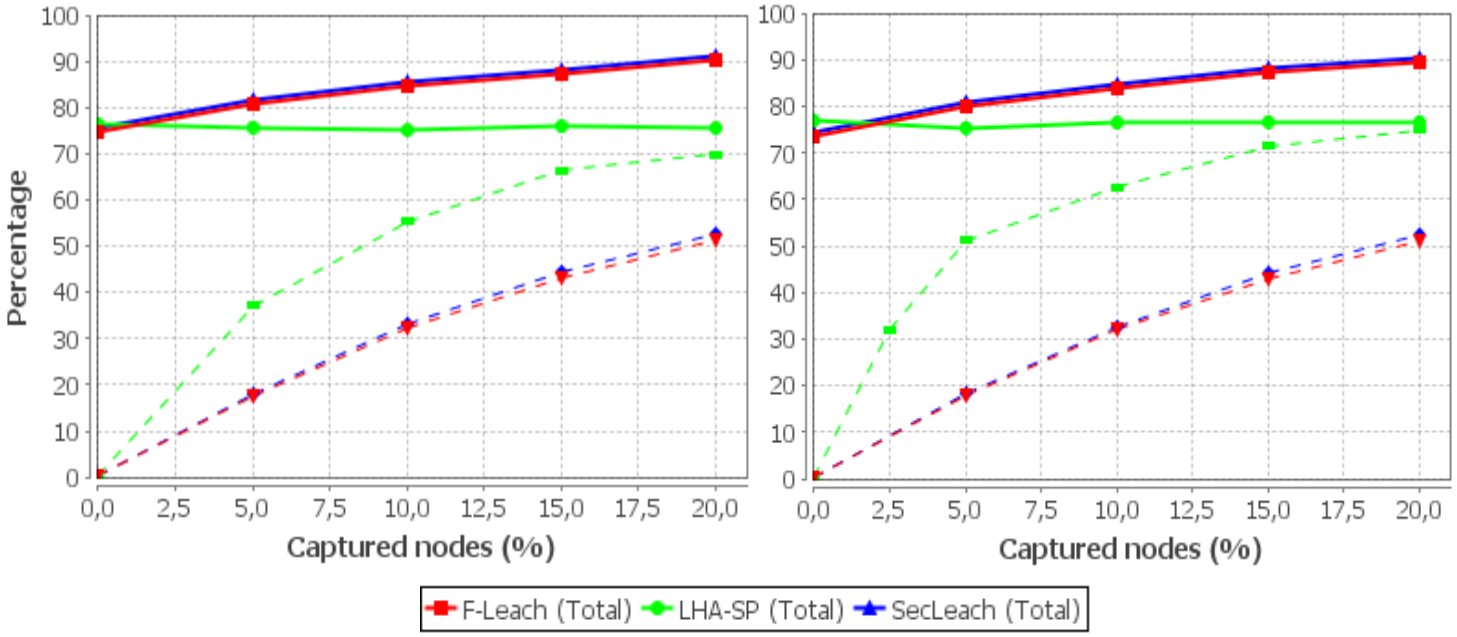

Figura 6.12 - Indicadores de cobertura face à percentagem de nós atacantes segundo um ataque sinkhole

6.4.1.2. Indicadores de fiabilidade

Com este estudo pretendeu-se medir o impacto na fiabilidade que um ataque *sinkhole* causa. Foi seguida a mesma topologia de testes da fase de operação, medindo a taxa de entrega de mensagens à *base station* quando são gerados eventos em 20% da rede e onde o atacante capturou 5% e 15% dos nós respectivamente. De modo a medir correctamente o impacto foram obtidos indicadores, de entre as mensagens que chegaram à *base station*, da percentagem das mesmas que ficaram sobre o controlo do atacante, podendo este atacar a sua autenticidade e/ou confidencialidade.

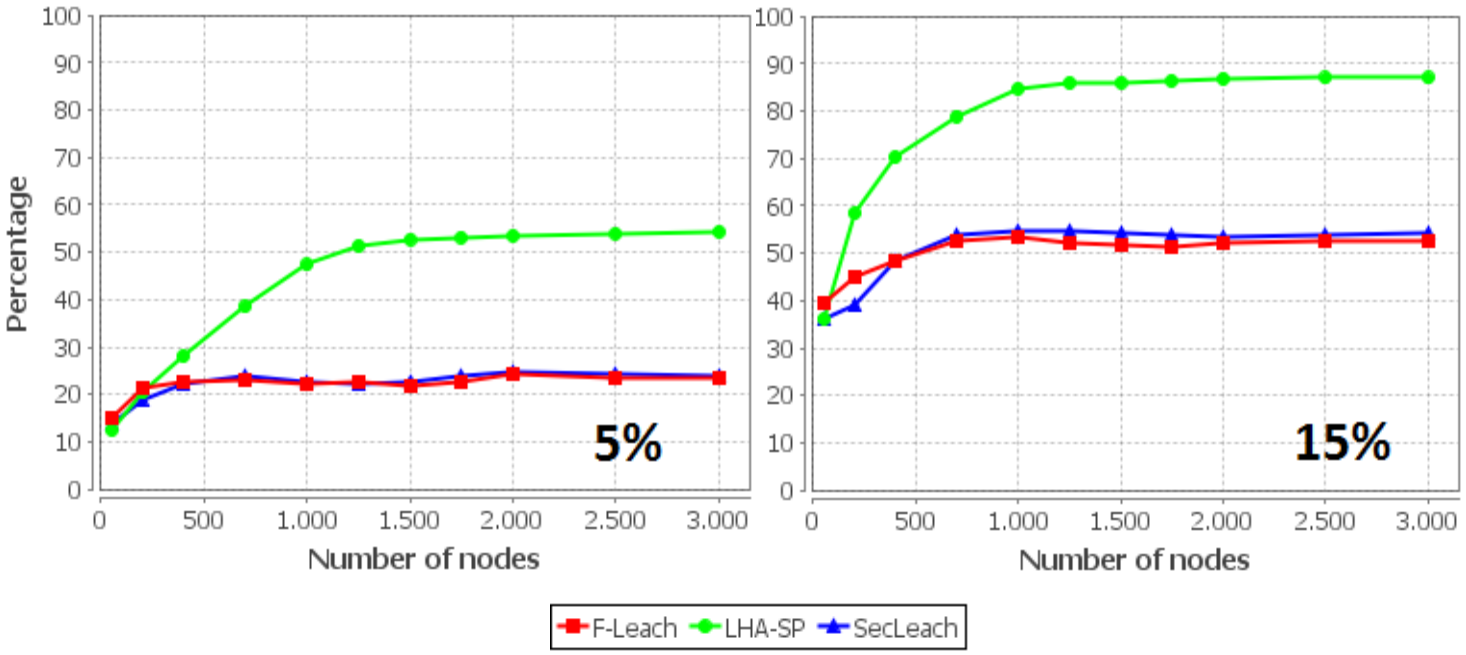

Figura 6.13 - Indicadores de percentagens de mensagens comprometidas durante um ataque sinkhole

Os resultados obtidos demonstram que um ataque *sinkhole* é bastante mais eficaz no LHA-SP devido às razões já enumeradas face à percentagem da rede que fica comprometida durante a fase de organização. Pode-se também verificar que os protocolos baseados no Leach apresentam indicadores mais constantes de percentagem de mensagens comprometidas devido à paralelização da formação de *clusters*.

6.4.2. Blackhole

Um ataque do tipo *blackhole* é previsto ter lugar após a fase de organização da rede e do estabelecimento de todos os *links*. O objectivo do atacante é novamente a captura de nós para apenas descartar todas as mensagens *multi-hop* que por ele passem de modo a evitar que estas cheguem ao destino, o que aplicado às especificações dos protocolos estudados nesta dissertação representa o descarte de mensagens dirigidas à *base station* e vice-versa.

Para a simulação deste ataque foram geradas capturas aleatórias de nós da rede de modo a incumbir-lhes o comportamento de descarte de mensagens durante a fase de operação. Poder-se-ia também considerar que o atacante tinha conhecimento prévio da topologia da rede e atacava os nós mais sensíveis mas a extracção de indicadores nestas situações seria pouco interessante

devido à previsibilidade dos resultados. Por exemplo, no LHA-SP seriam capturados os nós de nível mais alto o que comprometia imediatamente toda a rede. Já no SecLeach o atacante apontaria para os *cluster heads*, comprometendo novamente toda a rede (15% de nós capturados = 15% de *cluster heads* segundo as parametrizações de referência). Como tal os indicadores em seguida apresentados consideram uma captura de nós totalmente aleatória em todos os protocolos estudados e reflectem o impacto na fiabilidade da entrega de mensagens de modo a avaliar a eficácia do ataque. Novamente foram utilizadas como referência as percentagens de 5% e 15% de nós capturados e geração de eventos em 20% da rede.

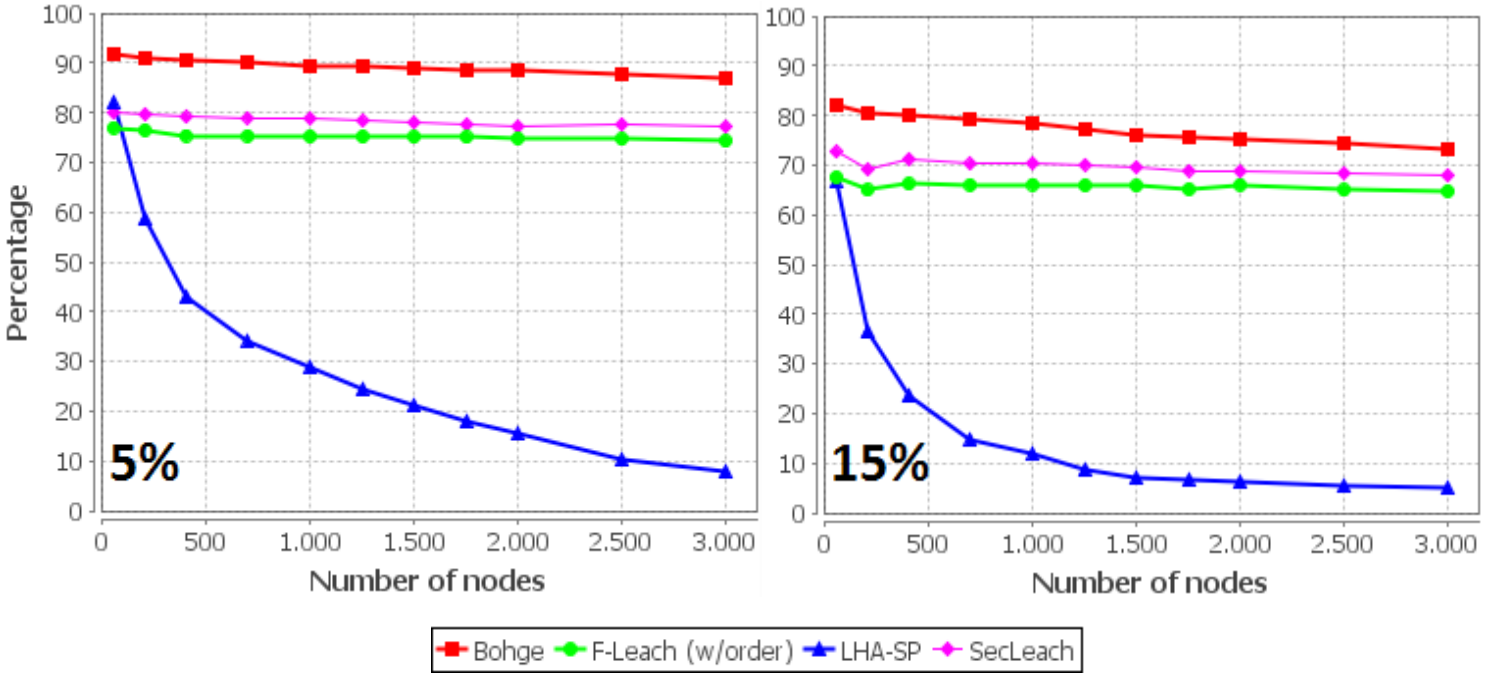

Figura 6.14 - Indicadores de fiabilidade face à dimensão da rede na presença de ataques blackhole

Verifica-se mais uma vez que o protocolo LHA-SP é bastante mais sensível à captura de nós do que os restantes protocolos. De entre todos, a *framework* de Bohge apresenta uma melhor resiliência devido à não formação de uma estrutura hierárquica bem definida, o que impossibilita o atacante de comprometer, mesmo aleatoriamente, certos nós mais sensíveis como os *cluster heads* nos protocolos baseados no Leach.

Observou-se também uma descida na taxa de fiabilidade nos protocolos que recorrem ao *flooding* de dados à medida que a rede aumenta de dimensão. A razão para tal deve-se ao

comprometimento de certas rotas no algoritmo de *flooding* que necessita também de um maior número de *hops* para alcançar a *base station* à medida que a rede aumenta. Estes factores, aliados à colisão de pacotes, forçam a taxa de fiabilidade a descer, especialmente na *framework* de Bohge onde o número de rotas é, por si só, já inferior devido à existência de menos nós, percentualmente, que encaminham mensagens.

Adicionalmente, foi novamente realizado um estudo que mede a taxa de fiabilidade de entrega de mensagens, para uma dimensão da rede fixa, face ao incremento de nós capturados pelo atacante que se comportam segundo um ataque *blackhole*. O objectivo foi o de apresentar uma visão, mais clara, da resiliência de cada protocolo face ao aumento de nós capturados. Os resultados demonstram mais uma vez a menor resistência do protocolo LHA-SP que usa uma estrutura hierárquica bem definida para o encaminhamento de mensagens até à *base station*.

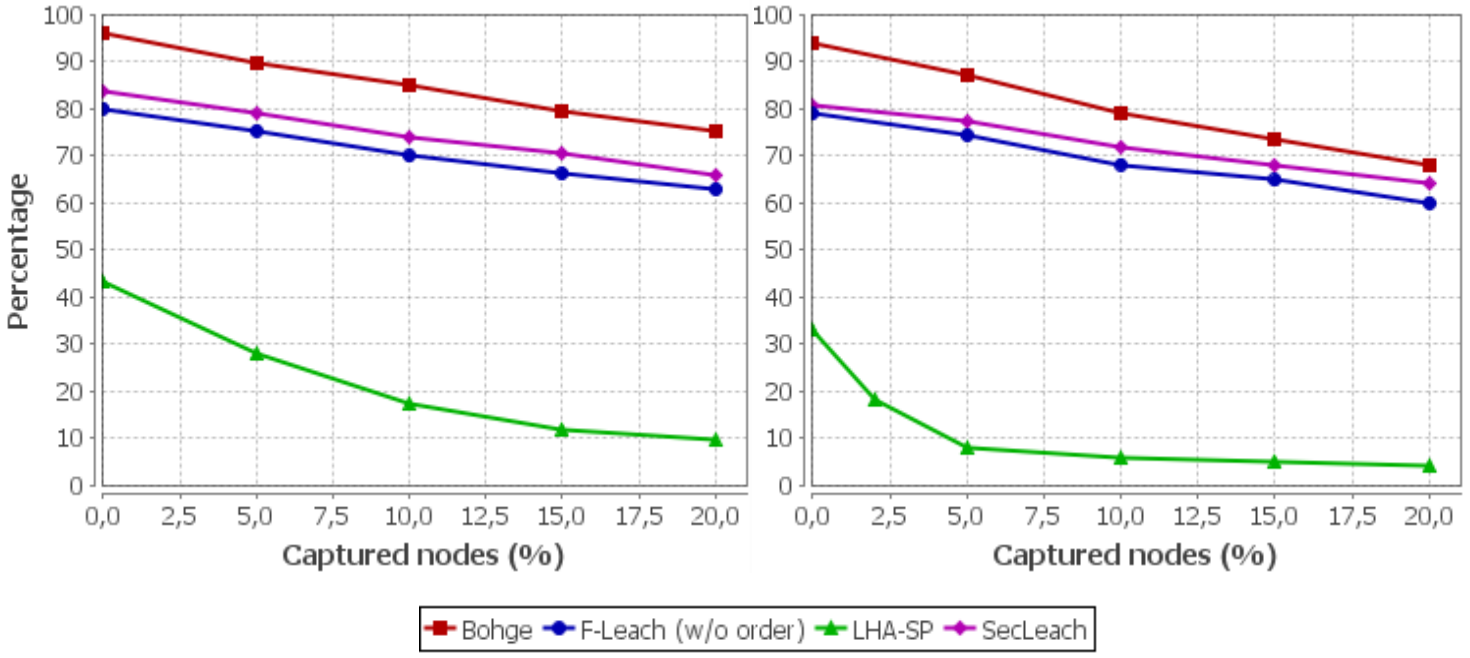

Figura 6.15 - Indicadores de fiabilidade face à percentagem de nós atacantes segundo um ataque blackhole

Finalmente, foi considerada a hipótese de o ataque *blackhole* ter início durante a fase de organização com recurso a uma estratégia de *hello flood* de modo a atrair tráfego para os nós atacados. Como tal os protocolos baseados no Leach serão preferencialmente mais vulneráveis pois o atacante pode formar novos *clusters* como foi visto no ataque anterior. O gráfico abaixo apresentado reflecte a diferença das taxas de fiabilidade, entre o ataque ter início na fase de

organização e na fase de operação, quando a percentagem de nós capturados aumenta. Pode-se observar que o impacto na fiabilidade é bastante maior se o ataque tiver início ainda durante a formação da rede e mais eficaz à medida que o número de nós capturados aumenta. Para o protocolo F-Leach foram observados valores na mesma ordem de grandeza pelo que o gráfico apenas apresenta o protocolo SecLeach.

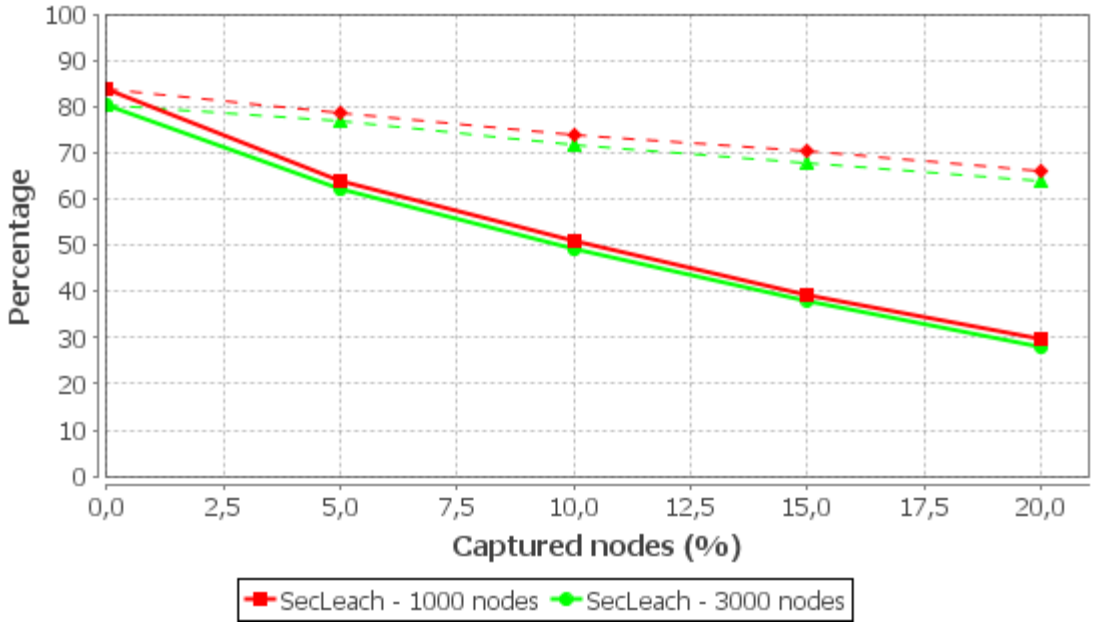

Figura 6.16 - Indicadores de fiabilidade comparativos entre um ataques blackhole com início na fase de organização e na fase de operação

7. Conclusões e trabalho futuro

Neste capítulo são apresentadas as conclusões que se podem retirar da elaboração de todo o trabalho, apresentando-se também os aspectos em aberto e trabalho futuro que se julga importante em consequência da elaboração da presente dissertação.

7.1. Conclusões

Na presente dissertação foi desenvolvida uma plataforma genérica de estudo e avaliação de protocolos de distribuição e estabelecimento de chaves em redes de sensores sem fios (RSSF) de larga escala. A plataforma é capaz de aferir e comparar, em base experimental aproximada a condições reais de funcionamento e de forma sistemática, o comportamento de diferentes protocolos. Foram também implementados, avaliados e comparados, sobre a plataforma desenvolvida, diversos protocolos orientados para formações em estruturas hierárquicas baseadas em *clusters*, sendo eles: F-Leach, SecLeach, LHA-SP e *framework* de Bohge.

A distribuição e estabelecimento de chaves criptográficas para RSSF é uma dimensão muito relevante para garantias de segurança nestas redes. A investigação recente propôs diversos modelos de pré-distribuição e estabelecimento de chaves criptográficas de entre os quais se destacam os protocolos probabilísticos e de auto-organização dinâmica e aleatória adaptados para o estabelecimento de topologias hierárquicas em *clusters* que permitem explorar possíveis heterogeneidades de nós sensores. No entanto as propostas realizadas não abarcam um estudo e discussão do comportamento dos protocolos face à operação em ambientes reais subjacentes ao funcionamento da pilha IEEE 802.15.4, limitando-se na grande maioria das vezes a apresentar apenas estudos de índole teórico, pelo que tal facto constituiu a principal motivação da presente dissertação. A partir deste ponto o trabalho teve duas vertentes: desenvolvimento de uma plataforma de simulação genérica de estudo experimental aproximado às condições reais de funcionamento das RSSF e, posteriormente, a avaliação de protocolos de referência na área da distribuição e estabelecimento de chaves criptográficas.

A plataforma desenvolvida partiu do núcleo base de simulação do jProwler que ofereceu um modelo de rádio e colisões bastante robusto e, no geral, ofereceu as melhores características face aos objectivos propostos. Ao núcleo base foram feitas várias extensões com o objectivo de tornar a plataforma o mais genérica possível e facilmente extensível, destacando-se o mecanismo de marcação e subscrição de eventos que serviu de base à implementação de módulos adicionais ao simulador. O resultado do trabalho desenvolvido nesta vertente definiu a contribuição de disponibilização de uma plataforma genérica de simulação, capaz de antecipar o comportamento dos protocolos em execução quando sujeites a condições reais de operação sobre a pilha IEEE 802.15.4, sendo extraídos indicadores importantes durante as várias fases dos mesmos em tempo de simulação para efeitos de validação e comparação. Os módulos adicionais implementados permitiram então extrair indicadores dos diversos tipos de cobertura da rede, tempo de estabilização da mesma, consumos energéticos tanto para a organização como para o envio de mensagens até à *base station*, latência e fiabilidade no envio/entrega das mesmas e, finalmente, foi também especificado um módulo que atribui a possibilidade de injecção de ataques internos na rede, tendo sido implementadas duas classes dos mesmos para efeitos de validação e análise dos protocolos.

A par da plataforma de simulação foram implementados quatro protocolos de distribuição e estabelecimento de chaves com o objectivo de providenciar um estudo dos mesmos no ambiente de simulação implementado e disponibilizado. Em comum têm o mecanismo de pré-distribuição de chaves, uma formação hierárquica baseada em *clusters* e o encaminhamento *multi-hop* de dados até à *base station*, mas diferem entre si no comportamento em condições de operação real, evidenciando-se pelas diferenças nos critérios enunciados. Foram também detectadas algumas limitações nos mesmos referentes ao escalonamento da rede, nomeadamente no encaminhamento *multi-hop* de dados. Como consequência, foi definida uma camada universal de encaminhamento com a implementação do algoritmo de *flooding* com detecção de duplicados, fornecendo primitivas de comunicação *multi-hop* quando as especificações originais dos protocolos falharam em fazê-lo ou simplesmente omitiram a especificação do encaminhamento.

O protocolo LHA-SP prevê a formação da rede em níveis de nós conforme as capacidades dos mesmos. É garantida a autenticação e confidencialidade dos dados tanto na fase de organização como de operação da rede através de uma chave global inicialmente pré-distribuída e partilhada por todos e, posteriormente, através de chaves geradas durante a primeira fase. O estudo em

ambiente de simulação demonstra que é um protocolo com boas taxas de cobertura, desde que os nós não sejam distribuídos aleatoriamente segundo uma rede totalmente plana, mas é bastante vulnerável a colisões e a ataques por intrusão. Além disso mostrou-se ser um protocolo com consumos e latências relativamente baixos durante a fase de organização e bastante baixos durante a fase de operação.

Os protocolos que estendem o Leach em termos de segurança distribuem a rede, gerada segundo uma topologia aleatória totalmente plana, em *clusters* que são formados durante a fase de organização. Conseguem garantir a autenticidade dos dados perante a *base station* mas diferem entre si na autenticação entre *cluster heads* e nós membros, sendo que no F-Leach os *cluster heads* autenticam-se perante os possíveis membros e no SecLeach ocorre o oposto. A extracção de indicadores dos mesmos demonstra boas taxas de cobertura, especialmente no SecLeach, mas consumos e latências elevados devido ao encaminhamento por *flooding*, que é bastante mais prejudicial no F-Leach devido às mensagens de maior dimensão. Em termos de resiliência face a ataques internos mostram-se relativamente resistentes e bastante mais que o protocolo anterior.

Finalmente, a *framework* de Bohge, apresenta-se como um protocolo que distribui a rede entre nós sensores e nós encaminhadores que constituem o *backbone* da rede, encaminhando dados. O protocolo apenas garante autenticação mútua entre *base station* e nós sensores e a premissa é bastante simples: os nós autenticam-se inicialmente, com recurso a certificados previamente distribuídos, e novos segredos são gerados que garantem a autenticação durante o resto do protocolo. Como contribuição da presente dissertação foi usado o algoritmo de *flooding* para o encaminhamento *multi-hop* já que a especificação original não refere a implementação de um algoritmo em específico. Em consequência dessa decisão foram observadas altas taxas de cobertura, assim como de fiabilidade de entrega de mensagens e resistência a ataques internos. Por outro lado, foi neste protocolo que se observaram latências mais elevadas, assim como consumos energéticos que, por si só já eram previsivelmente elevados devido às mensagens de maior dimensão, especialmente na fase de organização da rede quando os nós têm de se autenticar.

7.2. Aspectos em aberto e trabalho futuro

Como trabalho futuro prevê-se, na plataforma de simulação, uma revisão do núcleo de eventos de modo a paralelizar mais o despacho dos mesmos, com o objectivo de atingir uma melhor performance e ser possível a simulação de redes de ainda maior escala. Para tal seria necessário um estudo aprofundado das dependências entre eventos e até que ponto os mesmos poderiam ser processados em paralelo. A motivação para esta tarefa surge na sequência do peso computacional observado durante as simulações realizadas no capítulo de testes, especialmente na *framework* de Bohge, o que impediu de alargar a rede além dos 3000 nós. Adicionalmente a plataforma poderá também ser estendida com a implementação de módulos adicionais, entre os quais foi inicialmente pensado num que permitisse a geração de diversas topologias de *deployment*, como por exemplo em *grid*.

Em relação aos protocolos fica em aberto a possibilidade do uso de outra implementação de um algoritmo de encaminhamento que não o *flooding*. Devido ao seu uso foi observada uma alta taxa de cobertura e fiabilidade nos protocolos que mais dependem do mesmo, mas em contra-partida observou-se também o enorme crescimento no consumo energético e nas latências de estabilização da rede e entrega de mensagens. Como os sensores são dispositivos com baixas capacidades energéticas surge naturalmente o impulso da implementação de outro algoritmo de encaminhamento *multi-hop*, sendo também interessante a medição do *tradeoff* entre o mesmo e o *flooding* de dados no que respeita aos critérios enunciados. Já referindo protocolos em específico, seria interessante a extracção de indicadores referentes à fase de manutenção do protocolo LHA-SP que, apesar de implementada (capítulo 4.5.1), não chegou a ser avaliada no capítulo de testes.

Finalmente, e não menos interessante, seria também benéfico a implementação de mais protocolos de distribuição e estabelecimento de chaves de modo a avaliar melhor a generalização que foi dada à plataforma de simulação e aos seus módulos.

8. Bibliografia

[1] IEEE 802.15.4 Standard, Wireless MAC and PHY Specifications for Low Rate Wireless Personal Area Networks (WPANs), IEEE Standard, 802.15 IEEE Group, 2006

[2] Z. Alliance, "Zigbee specification," Technical Report Document 053474r06, ZigBee Alliance, June 2005.

[3] Yixin Jiang, Chuang Lin, Minghui Shi, and Xuemin (Sherman) Shen, “Key Management Schemes for Wireless Sensor Networks”, in Security in Sensor Networks, ISBN 9780849370588, 2006

[4] L. Eschenauer and V.D. Gligor, “A Key-Management Scheme for Distributed Sensor Networks”, in Proceedings of the 9th ACM Conference on Computer and Communications Security, Washington, DC, USA, November 18-22, 2002

[5] H. Chan, A. Perrig, and D. Song, “Random Key Predistribution Schemes for Sensor Networks”, in Proceedings of the 2003 IEEE Symposium on Security and Privacy, pp.197-213, Washington, DC, May 11-14, 2003

[6] D. Liu and P. Ning, “Location-based pairwise key establishments for static sensor networks”, in Proceedings of the 1st ACM Workshop on Security of Ad Hoc and Sensor Networks, pp. 72 –82, Fairfax, Virginia, 2003

[7] W. Du, J. Deng, Y.S. Han, S. Chen, and P.K.Varshney, “A Key Management Scheme for Wireless Sensor Networks Using Deployment Knowledge”, in Proc. of IEEE INFOCOM04, Vol. 1, pp. 586-597, 2004.

[8] A. C. Ferreira, M. A. Vilaça, L. B. Oliveira, E. Habib, H. C. Wong, and A. A. F. Loureiro, “On the security of cluster-based communication protocols for wireless sensor networks”, in 4th IEEE International Conference on Networking (ICN’05), volume 3420 of Lecture Notes in Computer Science, pp. 449–458, Reunion Island, April 2005.

[9] Leonardo B. Oliveira, Hao C. Wong, M. Bern, Ricardo Dahab, A. A. F. Loureiro, “SecLEACH – A Random Key Distribution Solution for Securing Clustered Sensor

Networks", in Proceedings of the Fifth IEEE international Symposium on Network Computing and Applications, Washington, DC, July 24-26, 2006

[10] L. B. Oliveira, H. C. Wang, A. A. Loureiro, "LHASP: secure protocols for hierarchical wireless sensor networks", in 9th International Symposium on Integrated Network Management, pp. 31-44, 2005.

[11] M. Bohge and W. Trappe, "An authentication framework for hierarchical ad hoc sensor networks", in Proceedings of the 2003 ACM workshop on Wireless security, pages 79–87. ACM Press, 2003

[12] TOSSIM, part of TinyOS, http://www.eecs.berkeley.edu/~pal/research/tossim.html PowerTossim, http://www.eecs.harvard.edu/~shnayder/ptossim/

[13] Freemote homepage, http://mote.tic.eia-fr.ch/freemote/

[14] JProwler homepage, http://w3.isis.vanderbilt.edu/projects/nest/JProwler/index.html

[15] Ana Sofia Querido Rito. Redes de sensores sem fios. Dissertação de Mestrado, Departamento de Informática da Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa, supervisionada pelo Prof. Doutor José Legatheaux Martins, 2006

[16] Jussi Haapola, "NanoMAC: a distributed MAC protocol for wireless sensor networks", in Proc. 18th Convention on Radio Science & IV Finnish Wireless Communication Workshop (FWCW'03), pp 17-20, Oulu, Finlândia, Outubro 2003

[17] Tijs van Dam, Koen Langendoen, "An adaptive energy-efficient MAC protocol for wireless sensor networks", in Proceedings of the 1st international conference on Embedded networked sensor systems, Los Angeles, California, USA, November 05-07, 2003

[18] Joseph Polastre, Jason Hill, David Culler, "Versatile low power media access for wireless sensor networks", in Proceedings of the 2nd international conference on Embedded networked sensor systems, Baltimore, MD, USA, November 03-05, 2004

[19] Injong Rhee, Ajit Warrier, Mahesh Aia, Jeongki Min, "Z-MAC: a hybrid MAC for wireless sensor networks", in Proceedings of the 3rd international conference on Embedded networked sensor systems, San Diego, California, USA, November 02-04, 2005

[20] Vojislav B. Miŝić, Jun Fung, and Jelena Miŝić, "MAC Layer Security of 802.15.4-Compliant Networks", Department of Computer Science, University of Manitoba Winnipeg, Manitoba, Canada, November 2007

[21] D. Dolev and A. Yao, “On the security of public key protocols,” IEEE Transactions on Information Theory, vol. 29, no. 2, pp. 198–208, 1983, iSSN: 0018-9448

[22] W. Stallings, Cryptography and Network Security, 4th edition.: Prentice Hall, 2006

[23] C. Karlof, N. Sastry, and D. Wagner, “TinySec: a link layer security architecture for wireless sensor networks”, in Proceedings of the 2nd international conference on Embedded networked sensor systems, Baltimore, MD, USA, 2004

[24] A. Perrig, R. Szewczyk, V. Wen, D. Culler, and J. D. Tygar. “SPINS: Security protocols for sensor networks”, in Wireless Networks, volume 8, number 5, pp. 521–534, September 2002

[25] M. Luk, G. Mezzour, A. Perrig et al., “MiniSec: a secure sensor network communication architecture”, in Proceedings of the 6th international conference on Information processing in sensor networks, Cambridge, Massachusetts, USA, 2007.

[26] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, “Energy-efficient communication protocol for wireless microsensor networks”, in Proceedings of the 33rd Hawaii international Conference on System Sciences, Washington, DC, January 04-07, 2000

[27] A. Perrig, R. Canetti, B. Brisco, D. Song, and D. Tygar, “TESLA: Multicast source authentication transform introduction” IETF working draft, draft-ietf-msec-tesla-intro-01.txt, October 2002

[28] Miguel Castro and Barbara Liskov, “Practical Byzantine Fault Tolerance”, in Proceedings of the Third Symposium on Operating Systems Design and Implementation, pp. 173-186, New Orleans, Louisiana, USA, 1999

[29] Germano Guimaraes, Eduardo Souto, Djamel Sadok, and Judith Kelner. 2005. “Evaluation of Security Mechanisms in Wireless Sensor Networks”, in Proceedings of the 2005 Systems Communications (ICW '05), August 2005

[30] G. de Meulenaer, F. Gosset, F. X. Standaert, and L. Vandendorpe, “On the Energy Cost of Communications and Cryptography in Wireless Sensor Networks”, in (extended version), IEEE International Workshop on Security and Privacy in Wireless and Mobile Computing, Networking and Communications (SecPriWiMob'2008), pages 580-585, 10 2008.

[31] M. Athanassoulis, I. Alagiannis, and S. Hadjiefthymiades, “Energy Efficiency in Wireless Sensor Networks: A Utility-Based Architecture,” in *Proc. 13th European Wireless Conf. (EW '07)*, Apr. 2007.

[32] A. Wander, N. Gura, H. Eberle, V. Gupta, and S.C. Shantz, "Energy Analysis of Public-Key Cryptography for Wireless Sensor Networks", in Proc. PerCom, 2005, pp.324-328.

[33] Bouncy Castle homepage, http://www.bouncycastle.org/

Bibliografia específica do Anexo

[34] A. Bharathidasan, V. Armand, and S. Ponduru, “Sensor networks: An overview”, Department of Computer Science, University of California, 2001

[35] H. Karl and A. Willig. A short survey of wireless sensor networks. Technical Report TKN-03-018, Telecommunication Networks Group, University of Berlin, October 2003

[36] You-Chiun Wang and Yu-Chee Tseng, “Attacks and Defenses of Routing Mechanisms in Ad Hoc and Sensor Networks”, in Security in Sensor Networks, ISBN 9780849370588, 2006

[37] Rodrigo Roman, Jianying Zhou, and Javier Lopez, “Applying intrusion detection systems to wireless sensor networks”, In Proceedings of the IEEE Consumer Communications and Networking Conference, Las Vegas, USA, 2006.

ANEXO

Introdução às Redes de Sensores sem Fios e abordagem da problemática da segurança nestas redes

A.1 Introdução às redes de sensores sem fios

Neste anexo pretende-se fazer uma introdução geral às redes de sensores sem fios (RSSF) apresentando-se uma visão geral das mesmas. Com este anexo não se pretende abarcar uma visão muito detalhada sobre todas as características e, para esse efeito, pode-se recorrer a bibliografia amplamente existente desde o início da investigação nesta área ([34], [35], [15]).

Redes de Sensores sem Fio (RSSF) são uma tecnologia emergente que promete uma funcionalidade avançada para monitorar e, eventualmente, controlar o mundo físico. RSSF são formadas por dispositivos de pequenas dimensões (cm3 ou mm3), a que chamamos sensores, que podem monitorizar fenómenos físicos tais como a pressão, temperatura, humidade, luminosidade, vibração, etc.. Estes sensores são dotados de capacidades computacionais limitadas, distribuídos por uma determinada área e que comunicam entre si sem fios por radiofrequência onde utilizam normas estabelecidas para a estruturação dos níveis físicos e de ligação de dados (IEEE 802.15.4 [1], Zigbee [2]).

Uma das grandes motivações para a criação destas redes prende-se com o avanço tecnológico que se tem verificado nas áreas de micro-processadores e micro-sistemas. Estima-se que no futuro o uso destas tecnologias aplicadas na criação de sensores venha a reduzir o custo destes para cerca de 1 euro cada. O baixo custo pode então propiciar a criação de redes com milhares ou até milhões destes sensores, tornando cada vez mais real o conceito de computação pervasiva.

Os nós sensores funcionam nestas redes como dispositivos autónomos, dotados de recursos locais de computação e de comunicação e por sensores específicos associados a processamento de sinais para monitorização de fenómenos físicos da área onde estão inseridos. Assim pode-se dizer que os sensores interagem com o meio ambiente, medindo ou monitorando indicadores associados a grandezas físicas e cooperam entre si para a disseminação e processamento dos

valores obtidos. Uma característica muito importante nos sensores é que geralmente possuem uma baixa capacidade energética e capacidades computacionais e de comunicação muito limitadas devido às suas pequenas dimensões. Como tal é normal que sejam propostos imensos protocolos diferentes tendo em vista essas limitações e os requisitos específicos de cada aplicação que vai ser suportada pela rede. Pode-se então dizer que uma rede de sensores é *application-driven*, ou seja, a pilha de protocolos e serviços pode sofrer grandes alterações de rede para rede devido às características já enunciadas.

Em seguida são apresentados dois exemplos de sensores bastante usados nestas redes que demonstram as suas baixas capacidades, limitações e respectivos consumos energéticos:

- MICAz: Micro-processador ATMEL Atmega128L de 8 bits com uma frequência de relógio 7.37 Mhz. Tem 128 Kbytes de memória dedicada e 512 Kbytes para armazenamento de dados. Usa duas baterias AA.
- TelosB: Micro-processador TI MSP430F1611 de 16 bits com uma frequência de relógio 8 Mhz. Tem 10 Kbytes de memória RAM, 48 Kbytes de memória dedicada e 1024 Kbytes para armazenamento de dados. Usa duas baterias AA.

Power consumption	MICAz	TelosB
Transmit	65 mW (@ Ptx = -5 dBm)	54 mW (@ Ptx = -5 dBm)
Listen	68 mW	60 mW
Receive	72 mW	61 mW
Compute	26 mW (@ 7.37 MHz)	4.8 mW (@ 4 MHz)
Sleep	25 μW (power down mode)	35 μW (low-power mode 3)

A.2 Aplicações das redes de sensores sem fios

Devido ao baixo custo dos sensores e à possibilidade de criar redes com milhares destes, povoando vastas áreas geográficas (dependendo da topologia) a sua possibilidade de uso em aplicações inovadoras tem-se mostrado bastante favorável. Assim, as redes de sensores têm sido testadas e exploradas em diversos cenários de aplicações, podendo-se destacar as seguintes áreas: (1) Militar - funções de monitoramento, rastreamento, segurança, controle e manutenção; (2) Engenharia – monitoramento de estruturas; (3) Aviação - substituindo as redes com fio, que são usadas hoje em dia; (4) entre muitas outras.

A.3 Aspectos de organização das redes de sensores sem fios e sua operação

A visão que um observador externo tem sobre uma rede de sensores, é de que se trata de um sistema distribuído composto por uma quantidade considerável de nós, responsável por medir e enviar informações, provenientes do ambiente em seu redor, a um nó especial denominado *Base Station* (BS) ou *Sink-Node*. Esse nó pode ser visto como mais um nó da rede ou um nó especial com mais recursos que os restantes, mas em todo o caso a informação flui para o mesmo. Já o método usado para a informação atingir o nó especial vai depender do algoritmo que é usado ao nível de rede, mais concretamente de encaminhamento.

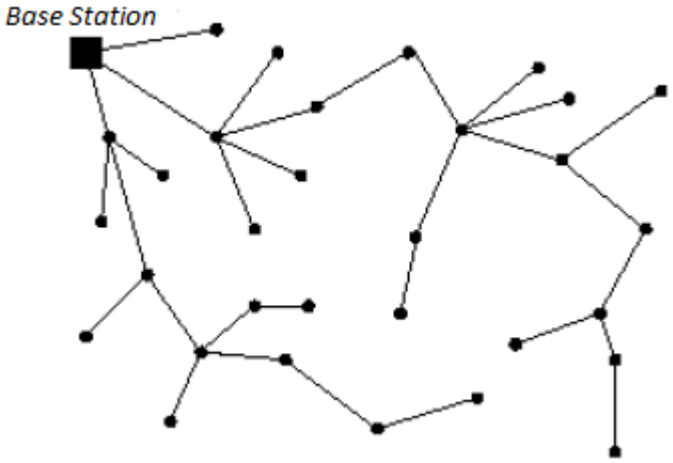

Figura A.1 – Rede ad hoc de sensores

No geral uma RSSF pode ser vista como uma rede *ad-hoc* sem fios, isto porque partilha características idênticas como auto-configuração, dinamicidade, comunicação directa entre nós sem presença de um ponto de acesso, entre outras. No entanto as RSSF possuem outras características que as diferenciam, destacando-se as seguintes:

- Capacidade energética e computacional limitada: Os sensores têm geralmente uma quantidade de energia finita e bastante baixa comparada com redes *ad-hoc* tradicionais que também possuem capacidade computacional (processamento e memória) geralmente superior;
- Grande quantidade de nós: O número de nós numa RSSF é numa ordem de grandeza bem maior que uma rede *ad-hoc*;

- Facilidade de ocorrência de falhas: Devido à quantidade de nós, à sua fragilidade e aos seus recursos limitados é normal e bastante comum que existam mais falhas, seja por omissão de dados (energia gasta) ou por falhas de comunicações motivadas por factores externos;
- Aplicações com fins específicos: RSSF costumam já ter a sua aplicação pré-definida no momento inicial da sua utilização, não possuindo tanta flexibilidade como redes *ad-hoc* tradicionais.

Tendo em vista estas diferenças, entende-se que a maior limitação das RSSF é mesmo a baixa capacidade energética e computacional. Devido a isso mesmo, os protocolos propostos para estas redes têm sempre de ter em conta esse factor e os melhores geralmente optimizam esses parâmetros.

Finalmente, o modelo de comunicação nestas redes passa pelo encaminhamento *multi-hop* de dados por diversos nós até chegarem à *base station*, o que permite alargar a rede para áreas fora do alcance rádio de um simples sensor. No entanto, em certos casos, são usados outros modelos, como *broadcast* ou *multicast*. A título de exemplo o *broadcast* pode ser aproveitado pela *base station* que geralmente possui maiores recursos.

A.4 Aspectos associados a requisitos de escala e auto-organização

Um conceito muito importante em RSSF é que as mesmas sejam constituídas por uma quantidade muito grande de sensores de baixo custo e que no geral toda a rede possa cobrir uma vasta área geográfica. No entanto a área inicialmente coberta pode ter de ser expandida para englobar outros pontos de interesse e essa extensão deve ser tratada de uma forma tão simples como a simples adição de sensores à rede. Outro factor que propicia a adição de nós à rede é as falhas que os que pertencem de momento podem sofrer, tendo portanto de ser substituídos por novos.

Assim, torna-se interessante e quase obrigatório que a rede seja capaz de se auto-organizar e a respectiva manutenção topológica seja pouco exigente, de preferência sem qualquer tipo de intervenção humana. Não obstante, as soluções propostas devem sempre ter em conta as características limitadas dos sensores que populam a rede.

A.5 Aspectos sobre topologia

Um factor muito importante em RSSF é a topologia assumida pela rede pois pode ser graças a essa organização que a rede será mais ou menos eficaz. De entre as possíveis topologias destacam-se: organização hierárquica ou orientada a grupos (*clusters*), em estrela ou ainda em malha (*meshed*). Em certas condições podem-se também formar estruturas híbridas. As topologias assumidas por cada rede influenciam então a sua eficácia, ou seja, têm de estar fortemente dependentes das características dos seus nós, pelos requisitos necessários de cobertura e ainda pelos requisitos e necessidades das aplicações.

Topologia em estrela

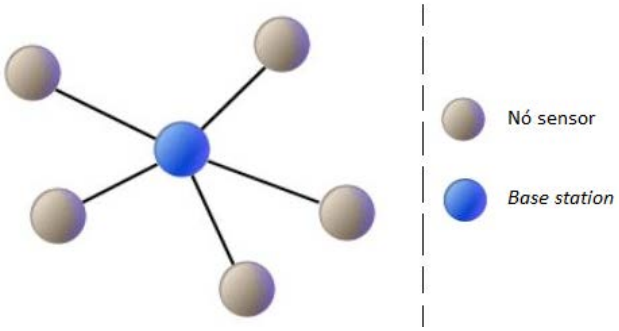

Figura A.2 – Topologia em estrela

É a topologia mais básica, onde cada sensor pode comunicar directamente com a *base station* mas as distâncias geográficas que a rede no geral pode atingir são muito baixas devido à limitação do sinal de rádio nos sensores, que não se podem colocar muito longe da BS.

Topologia em malha

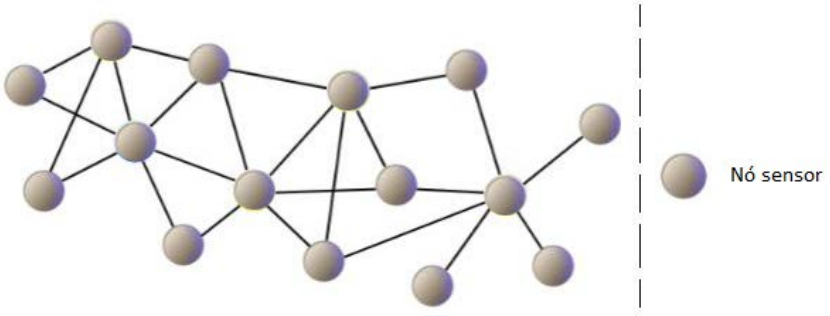

Figura A.3 – Topologia em malha (meshed)

Numa topologia em malha todos os nós desempenham o mesmo papel e têm de se coordenar entre si através de algoritmos de sistemas distribuídos. Uma grande vantagem destas topologia é o longo alcance alcançado pela rede devido à possibilidade de quaisquer dois nós se poderem ligar entre si, como não acontece na topologia em estrela. Se a rede for dinâmica e a posição dos seus nós variar ao longo do tempo esta organização pode-se mostrar então benéfica. No entanto há que ter em conta que muitas ligações deverão ser estabelecidas, o que terá impacto na memória e no consumo energético. Assim pode-se considerar um *tradeoff* entre cobertura e tempo de vida útil da rede.

Topologia hierárquica ou em grupo (clusters)

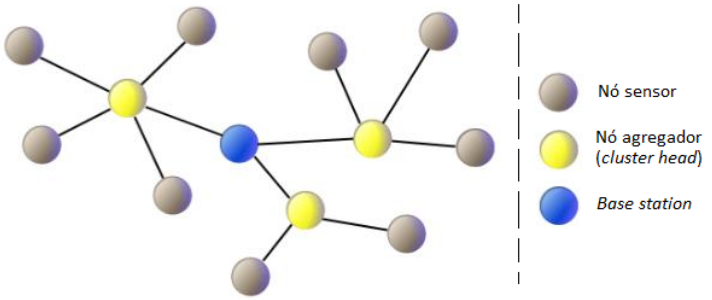

Figura A.4 – Topologia hierárquica ou em grupo

Uma topologia hierárquica cria, como o nome indica, uma hierarquia entre os nós. O caso mais ilustrativo é a organização por grupos onde cada grupo contém um nó agregador, chamado também *cluster head*, que é seleccionado de forma aleatória ou com base em critérios como as características físicas, energia disponível, entre outros. Geralmente este tipo especial de nós recolhe informações dos nós mais simples, agregam tudo e depois enviam para a *base station*. Relativamente aos nós mais simples, os mesmos procuram juntarem-se a um agregador e daí formar a hierarquia. Para uma rede estática esta topologia pode trazer muitas vantagens, entre as quais: (1) número reduzido de ligações entre os nós o que implica menos memória usada e consequentemente um consumo energético também menor; (2) simplificação de processos de coordenação, de entre os quais destaco o processo de distribuição e refrescamento de chaves, onde a responsabilidade pode ficar a cargo do agregador que depois notifica os nós perante os quais é responsável.

Ainda sobre a topologia em grupos é importante referir que nem sempre os nós agregadores podem ser nós especiais com mais recursos que os outros e normalmente são iguais a eles, com

os mesmos recursos. Como os agregadores geralmente são os que mais recursos consomem e se nada os diferenciar dos restantes então geralmente aplica-se um mecanismo que elege novos agregadores periodicamente, distribuindo a carga por todos.

A.6 Pilha de serviços numa arquitectura de RSSF

Redes de sensores são muito *application-driven*, ou seja, muito orientadas à aplicação o que implica que as arquitecturas de *software* para suporte das mesmas sejam específicas de cada uma. Essa observação implica também que a pilha seja normalmente definida com base nos requisitos de cada aplicação. No entanto não deixa de ser importante a estruturação dos serviços em camadas *middleware* que facilitem o desenvolvimento das aplicações. Na figura A.5 pode-se observar uma aproximação de como deverá ser uma pilha de suporte típico em redes de sensores sem fios. A camada de segurança é aquela que mais dúvidas levanta quanto à sua posição na pilha já que segurança pode ser implementada a vários níveis, mas para uma estruturação base de uma pilha assume-se que se coloca ao mesmo nível da camada de rede, ou até acima desta.

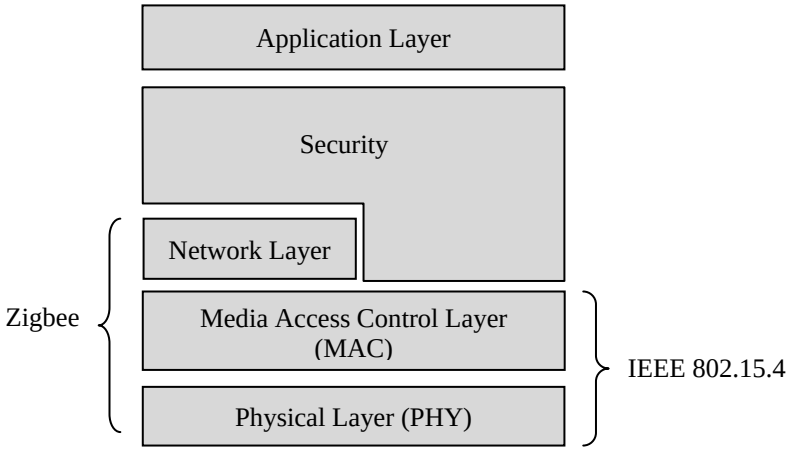

Figura A.5 – Pilha de suporte em RSSF

A.7 Introdução ao estudo da segurança em RSSF

As redes de sensores sem fios, devido a operarem durante longos períodos sem intervenção humana estão, portanto, expostas a vários ataques sem que se tenha conhecimento dos mesmos. O facto de operarem sobre comunicações *wireless* torna-as um alvo fácil pois estão abertas a

toda a gente, incluindo atacantes que nem sequer precisam de capturar fisicamente um nó para causar danos em toda a rede (atacantes externos). Como já foi referido, o consumo energético é também um factor essencial na construção destas redes e pode muito bem ser aproveitado como vantagem para os atacantes, que tentam gastar os poucos recursos energéticos disponíveis. Não obstante, também é uma limitação na escolha de algoritmos criptográficos para tornar a rede mais segura. Assim a segurança pode, e deve, ser vista como um critério-chave numa rede de sensores.

A investigação na área de segurança nestas redes tem abarcado diversas dimensões, de onde se podem destacar:

- Dimensão associada a uma nova perspectiva e definição do modelo de adversário. Esta dimensão pretende prever toda uma topologia de ataques que podem ser desencadeados por atacantes, tanto externos como internos. Um modelo de adversário para RSSF tem geralmente por base o modelo de Dolev-Yao [21] que é insuficiente pois não contempla a captura física e consequentemente atacantes internos;
- Dimensão associada a ataques a nível MAC que geralmente têm o objectivo de gastar recursos energéticos da rede com mensagens inúteis, de criar colisões, ou de lançar ataques de negação de serviço [20];
- Dimensão associada a ataques ao mecanismo de encaminhamento de dados na rede que podem comprometer os mesmos e levam muitas das vezes a rede a encaminhar pacotes para os nós do adversário, o qual tomará controlo sobre o destino dos mesmos [36];
- Agregação segura de dados e processamento dos mesmos em redes de sensores;
- Tolerância a falhas e mecanismos de detecção de intrusões em RSSF [37]. Dimensão que visa explorar protocolos tolerantes a falhas, especialmente arbitrárias que visam também detectar ataques internos por intrusões;
- Dimensão de estudo de algoritmos criptográficos que têm em conta as limitações e baixos recursos energéticos e computacionais dos sensores;
- Dimensão que visa estudar mecanismos de distribuição e estabelecimento de chaves entre sensores. Os mecanismos geralmente são optimizados para certas topologias de rede e para requisitos particulares da aplicação. Actualmente a vertente mais estudada é o mecanismo de pré-distribuição de chaves simétricas que é aquele abordado também na presente dissertação.