Universidade Nova de Lisboa
Faculdade de Ciências e Tecnologia
Departamento de Informática

Dissertação de Mestrado

Mestrado em Engenharia Informática

Simulação e Estudo Experimental de Protocolos de Encaminhamento Seguro com Tolerância a Intrusões em Redes de Sensores sem Fios de Grande Escala

Tiago Araújo (28047)

Lisboa
(Fevereiro de 2011)

Universidade Nova de Lisboa
Faculdade de Ciências e Tecnologia
Departamento de Informática

Dissertação de Mestrado

Simulação e Estudo Experimental de Protocolos de Encaminhamento Seguro com Tolerância a Intrusões em Redes de Sensores sem Fios de Grande Escala

Tiago Araújo (28047)

Orientador: Prof. Doutor Henrique Domingos

Trabalho apresentado no âmbito do Mestrado em Engenharia Informática, como requisito parcial para obtenção do grau de Mestre em Engenharia Informática.

Lisboa
(Fevereiro de 2011)

À minha família

Agradecimentos

Esta dissertação é pela sua finalidade académica um trabalho individual, contudo não posso deixar de expressar os meus sinceros agradecimentos às pessoas que contribuíram em certa forma na sua realização:

Em primeiro lugar, os meus agradecimentos ao Prof. Doutor Henrique João Lopes Domingos, orientador desta dissertação, por todo o apoio prestado, bem como a orientação e sugestões dadas, o que considero ter sido determinante na elaboração da dissertação.

Uma palavra de apreço para a Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa, na qual tive o prazer de realizar com toda a satisfação esta minha caminhada académica.

À minha família gostaria de expressar a minha profunda gratidão, por terem sido o contínuo apoio durante todos estes anos, e por serem a base da construção e coerência dos meus próprios valores, um muito obrigado.

Por fim, deixo ainda uma palavra a todos os meus amigos que compreenderam as horas em que privei da sua companhia, e que sempre me apoiaram e incentivaram.

Resumo

As redes de sensores sem fios (RSSF) apresentam desafios complexos de segurança, sobretudo quando suportam aplicações críticas e quando operam em cenários de grande escala. Neste tipo de ambientes, as RSSF podem operar geralmente sem supervisão ou com supervisão muito reduzida, podendo estar expostas a várias tipologias de ataques, sejam ataques às comunicações por rádio-frequência, sejam ataques por intrusão com comprometimento dos seus nós.

O objectivo da presente dissertação visa o estudo experimental por simulação de protocolos de encaminhamento seguro com mecanismos de tolerância a intrusões, que sejam adequados a suportar RSSF de grande escala e que tenham que operar sem supervisão. Neste estudo é importante que a eficácia e efectividade dos mecanismos de tolerância a intrusões sejam vistos de forma complementar a outras avaliações de impacto, nomeadamente em relação a critérios de fiabilidade, latência, cobertura e consumo energético da rede no seu conjunto e dos seus nós, individualmente.

A dissertação insere-se nesta direcção de investigação, realizando o estudo de protocolos de encaminhamento com mecanismos de tolerância a intrusões a partir de uma aproximação experimental por simulação, avaliando o impacto de tipologias de ataques caracterizados na terminologia e literatura das RSSF face à injecção de ataques ou falhas ao nível de nós que possam ter sido comprometidos por um atacante. Nestas tipologias consideram-se ataques por *Hello-Flooding*, *Sinkhole*, *Wormhole*, *Sybil*, *Blackhole* e *Selective-Forwarding*.

Os resultados das análises e simulações são confrontados com análises teóricas preliminares da arquitectura, concepção e operação desses protocolos, tendo em conta que aqueles ataques podem ser concretizados através do potencial de ameaça às comunicações, bem como à possível captura e comprometimento dos nós sensores. Os resultados

da dissertação incidem sobre a análise de topologias aleatórias de redes de grande escala, entre 1000 a 10.000 nós, e sensores com características similares a sensores do tipo *MicaZ* ou *TelosB* da *Crossbaw*, modelando e simulando condições de operação de sensores em ambiente TinyOS e que usam TinySec como camada primária de segurança das comunicações por rádio-frequência.

As simulações e análises realizadas ao longo da dissertação incidiram sobre os sistemas de encaminhamento INSENS e CleanSlate que são referências relevantes na investigação da área dos protocolos de encaminhamento tolerantes a intrusões para RSSF. Dos resultados da dissertação é possível retirar conclusões relevantes sobre o desempenho, operação e condições de segurança esperados nesses protocolos quando operam em condições aproximadas a condições reais. Estes resultados complementam as análises teóricas prévias apresentadas na literatura, e que não cobrem todas as dimensões de estudo desses protocolos nas hipóteses da presente dissertação. Deste modo, a dissertação apresenta uma avaliação mais completa e rigorosa desses protocolos, a partir de um ambiente de simulação para redes em cenários de grande escala, com modelação e injecção de ataques por intrusão.

Palavras-chave: Redes de Sensores sem Fios (RSSF); Protocolos de encaminhamento; Segurança: autenticação, confidencialidade e integridade de dados; Indicadores de desempenho: cobertura e conectividade da rede, latência, consumo energético e fiabilidade; Tolerância a intrusões; Protocolos de encaminhamento tolerantes a intrusões.

Abstract

Wireless Sensor Networks (WSA) have complex security challenges, specially when are used in support of critical applications and at large scale environments. In this kind of environments, WSN's usually operate with reduced supervision, or without supervision at all, which leads to an exposure to several attack typologies, namely radio-frequency attacks to the communications or intrusion attacks that can compromise the network nodes.

The purpose of this dissertation aims the experimental study by simulation of secure routing protocols with intrusion-tolerant mechanisms, which are suitable for supporting large scale WSN's and for working without supervision. In this study it is important that efficiency and effectiveness of intrusion-tolerant mechanisms can be seen as a complement to the other impact evaluations, namely in criteria regarding reliability, latency, coverage and energy consumption in the whole network and in its nodes individually

The dissertation follows this research direction, performing the study of routing protocols with intrusion-tolerant mechanisms from an experimental approach by simulation. It also evaluates the impact of typologies attacks characterized in the WSN's terminology and literature, against injection attacks or failures at the nodes that might been compromised by an attacker. The attacks considered are *Hello-Flooding*, *Sinkhole*, *Wormhole*, *Sybil*, *Blackhole* and *Selective-Forwarding*.

The analysis and simulations results are confronted with theoretical preliminary analysis of the architecture, design and operation of these protocols, bearing in mind that those attacks can be performed by potential threat to the communications as well as by compromising sensor nodes. The results of the dissertation focus on the analysis of random topologies for large scale networks, around 1000 to 10.000 nodes, and on

sensors with characteristics similar to MicaZ or TelosB by Crossbaw, which model and simulate conditions of operation for sensors in TinyOS environment and use TinySec as primary layer to security in radio-frequency communications.

The simulations and analysis performed throughout the dissertation were focused on INSENS and Clean-Slate routing systems, which are relevant references in research about intrusion-tolerant routing protocols for WSN. Following the results from this dissertation, it is possible to get relevant conclusions about performance, operation and expected security conditions on protocols when operating on conditions approximated to the real ones. These results complement the previous theoretical analysis presented in the literature, and does not cover all aspects of study to such protocols in this dissertation. Thus, the dissertation presents a more complete and rigorous evaluation of these protocols, by using a simulation environment for large scale networks, with modeling and injection attacks by intrusion.

Keywords: Wireless Sensor Networks (WSN); Routing protocols; Security: authentication, confidentiality and data integrity; WSN Performance and operation criteria: connectivity, latency, energy and reliability; Intrusion tolerance; Intrusion-tolerant routing.

Conteúdo

Lista de Figuras

Lista de Tabelas

Listagens

1

Introdução

1.1 Introdução geral

Uma rede de sensores sem fios (RSSF) é constituída por vários nós sensores, que são dispositivos de dimensões bastante reduzidas e distribuídos espacialmente, comunicando entre si através de um meio de comunicação desprovido de fios. Os nós sensores têm como principal objectivo monitorizar determinados fenómenos físicos e consequentemente encaminhar os respectivos dados até ao observador externo à rede. No entanto, estes dispositivos apresentam recursos bastante limitados (por exemplo ao nível do processamento, memória e da energia) que condicionam, de certo modo, o funcionamento da rede e, assim sendo, importa que os sensores actuem de forma colaborativa. Portanto, numa RSSF o modelo de comunicação da própria rede passa essencialmente pelo encaminhamento da informação ao longo dos vários nós, sendo determinante a existência de sistemas de encaminhamento que dêem suporte a esse mecanismo.

Vários sistemas de encaminhamento têm sido propostos, contudo o facto de estas redes tenderem a ser orientadas para um destinado tipo de aplicação faz com que alguns sistemas se apresentem mais vantajosos que outros em determinadas circunstâncias. Exemplo disso é o caso de sistemas de encaminhamento que se revelam mais apropriados para redes de larga escala, ao contrário de outros. Embora o encaminhamento se enquadre ao nível lógico da pilha de camadas de um nó sensor, ele tem influência directa nos resultados provenientes das medições aplicadas à rede: cobertura,

latência, fiabilidade e consumo energético.

Porém, como qualquer outro tipo de rede, as RSSF também incluem a problemática da segurança, pois podem ser alvo de ataques que têm em consideração as características que lhes são específicas. O facto dos nós sensores serem dispositivos com recursos limitados impõe algumas barreiras na aplicação dos tradicionais mecanismos de segurança. As questões de segurança estendem-se a vários níveis, inclusive no que diz respeito ao nível do encaminhamento, sendo aqui que os sistemas de encaminhamento seguro denotam a sua importância. Nestas redes as hipóteses do atacante são muito diversificadas, o atacante pode mesmo actuar facilmente por intrusão, uma vez que são dispositivos vulneráveis a esse tipo de ataques, o que consequentemente aumenta as exigências na concepção de sistemas de encaminhamento seguro.

Nesta secção apenas foi feita uma preliminar e sucinta abordagem à introdução das RSSF, contextualizando os objectivos pretendidos que seguidamente são referidos. Contudo, encontra-se em anexo uma visão, ainda que introdutória, um pouco mais detalhada sobre este tipo de redes.

1.2 Motivação e enquadramento

As redes de sensores sem fios são alvo de grande interesse por parte da comunidade de investigação, tendo esta cada vez mais demonstrado que, devido às características e aos requisitos de segurança que as aplicações exigem, a segurança é um dos aspectos de maior importância neste tipo de redes. Como tal, podemos ter como exemplo o caso das RSSF estarem muitas vezes distribuídas em locais inacessíveis e com uma reduzida vigilância.

Outro aspecto a ter em conta é o facto de as técnicas, mecanismos e serviços de segurança presentes nas redes convencionais não poderem simplesmente ser transpostos para estas redes, uma vez que as limitações ao nível dos dispositivos que as constituem são bastante penalizadoras. Assim sendo, surge a necessidade de encontrar soluções de segurança que se adeqúem e que demonstrem ser eficientes face às possíveis tipologias de ataques a considerar, nos vários níveis da pilha de estruturação de serviços. Ao nível rede/encaminhamento, alguns sistemas de encaminhamento seguro foram propostos, e é verdadeiramente um dos temas nesta área que está permanentemente sob investigação, tendo em vista a obtenção de melhores soluções de forma a realizar o encaminhamento tão seguro quanto possível (considerando as limitações da rede) perante várias hipóteses do atacante. Torna-se bastante importante avaliar e comparar alguns dos sistemas de encaminhamento que têm sido propostos, na medida em que se consiga obter conclusões válidas acerca da qualidade do respectivo sistema quando

implementado num determinado tipo de rede e exposto a determinadas tipologias de ataques. Para que estes sistemas possam ser implementados em ambientes reais eles requerem um estudo prévio tendo por base ambientes de simulação, o que permitirá dar a ideia do verdadeiro comportamento ainda antes de ser transposto para o ambiente real. Todos estes aspectos mencionados acabam assim por constituir as motivações que surgem como ponto de partida para os objectivos e contribuições delineados na presente dissertação.

Relativamente ao enquadramento exigido para a elaboração desta dissertação, foi necessário realizar um estudo preparativo que incidiu na introdução às RSSF e suas aplicações. Este estudo abarcou as características gerais destas redes, nomeadamente os princípios e fundamentos associados à tipologia de *hardware*, sistema operativo e estruturação da pilha de serviços de *software* dos dispositivos usados como nós das RSSF (vulgarmente designados por sensores ou *motes* – na terminologia de língua inglesa), a organização e operação das redes, e ainda uma abordagem às aplicações. A preparação envolveu ainda um enfoque na problemática da segurança nos diversos níveis da referida pilha, bem como a familiarização com alguns ambientes de simulação e emulação que existem para suporte de desenvolvimento neste tipo de redes. Este estudo preparativo obrigou também a entender os princípios de funcionamento do suporte básico de comunicação das RSSF na pilha IEEE802.15.4 [41503], o que inclui o conhecimento de normas e princípios de funcionamento dos protocolos de ligação de dados (nível MAC), bem como algumas variantes e melhorias introduzidas a este nível pela investigação recente para suporte de topologias de redes de grande escala e de características *multi-hop*.

De modo a focar o estudo nos objectivos e contribuições pretendidas, foi necessário o levantamento e elaboração de uma síntese de trabalho relacionado a alguns dos protocolos de encaminhamento de RSSF que têm sido propostos na investigação recente. Os protocolos estudados apresentam diferentes pressupostos e propriedades de segurança, sendo portanto elaborada uma abordagem analítica e comparativa.

O trabalho relacionado visa, também, particularmente o estabelecimento de uma análise crítica de hipóteses associadas a modelos de adversário e tipologias de ataques ao encaminhamento, nomeadamente nas diferentes fases de operação de um protocolo de encaminhamento para uma RSSF: a fase de descoberta de rotas, a fase de selecção de rotas e a fase de controlo do encaminhamento (após estabelecimento das rotas). Nesta abordagem inclui-se uma análise de técnicas e condições de resiliência a partir de contramedidas que podem ser usadas como defesas contra diferentes tipologias de

ataques. São consideradas técnicas e mecanismos de resiliência pró-activa ou de tolerância a intrusões, e ainda aspectos complementares a serviços básicos de segurança das comunicações e que asseguram, em geral, propriedades de autenticação, confidencialidade, integridade, defesa contra retransmissão ilícita de mensagens e mecanismos primários de controlo de acessos.

Por fim, realizou-se também um estudo a alguns ambientes de simulação de redes *ad-hoc* e RSSF, sendo proposto um conjunto de critérios que orientaram a escolha de uma base de simulação para criação da plataforma objectivada pela dissertação.

1.3 Objectivos e contribuições

A problemática da concepção e avaliação de condições de segurança de sistemas de encaminhamento seguro para RSSF reside, sobretudo, no facto de como se comportariam estes sistemas de encaminhamento sem e com a presença de ataques enquadrados em determinadas tipologias de ataques. A questão que se pode colocar, passa por questionar qual é realmente a eficácia das contramedidas inerentes ao sistema perante certos ataques (ao encaminhamento) numa rede com determinadas características.

Nesse sentido, um dos objectivos desta dissertação passa por dar uma resposta, visando a concepção, implementação e avaliação experimental de uma plataforma genérica, tendo como suporte primário a utilização de um ambiente de simulação já previamente desenvolvido e apropriado para simular RSSF. Esta plataforma genérica deverá simular o comportamento de RSSF em condições aproximadas ao funcionamento real das redes IEEE 802.15.4 [41503], com sensores do tipo *TelosB* ou *Micaz* (pertencentes à gama *Mica motes*), ambos da *Crossbaw*, e que executam o sistema TinyOS, suportando uma pilha de comunicações seguras baseada na norma TinySec [KSW04]. Estes sensores foram escolhidos por serem utilizados actualmente e disporem de melhores recursos. Serão consideradas redes com um elevado número de nós (até cerca de 10.000 nós), ou seja, redes de grande escala. A plataforma genérica terá de ser constituída por módulos adicionais ao ambiente base de simulação, de modo a oferecer a possibilidade de efectuar medições (cobertura, latência, fiabilidade e energia) na rede, e ainda programar e parametrizar a injecção de tipologias de ataques específicos ao encaminhamento.

Relativamente aos protocolos de encaminhamento, tem-se também como objectivo a implementação dos protocolos de encaminhamento seguro: INSENS [JD02] e CleanSlate [PLGP06]. A estes protocolos pretende-se que possam ser modeladas e desencadeadas diversas tipologias de ataques. Nesta dissertação são considerados, para efeitos de avaliação, os ataques: *Hello-Flooding*, *Wormhole*, *Sinkhole*, *Sybil*, *Blackhole* e *Selective*

Forwarding, cujos ataques podem surgir através de hipóteses de captura física de nós.

Portanto, estabelecem-se as condições necessárias para se realizarem medições e avaliações experimentais aos sistemas de encaminhamento referidos, sobretudo quando estão sujeitos a ataques, sendo então avaliado o impacto provocado e a eficácia das suas contramedidas para redes com as características acima mencionadas (essencialmente de alta escalabilidade). Os efeitos verificados podem reflectir-se a diversos níveis, como são os casos da cobertura, latência, fiabilidade de entrega das mensagens encaminhadas, ou ainda ao nível do consumo energético, os quais constituem assim critérios base para efectuar uma avaliação.

No que diz respeito às principais contribuições pretendidas nesta dissertação, tem-se como contribuição inicial a concepção, implementação e avaliação da plataforma genérica que permitirá a análise e avaliação do comportamento dinâmico de sistemas de encaminhamento em redes de larga escala, bem como avaliar o impacto de tipologias de ataques. De notar que esta plataforma é baseada num ambiente de simulação previamente desenvolvido e que será estendido de forma a incorporar módulos parametrizáveis para a modelação e injecção de ataques, assim como para realizar medições.

A seguinte contribuição é obtida através da anterior, isto é, através da referida plataforma genérica será possível realizar testes e análises de forma a avaliar o comportamento dos protocolos CleanSlate [PLGP06] e INSENS [JD02], bem como o impacto e as consequências que cada uma das tipologias de ataques consideradas provoca na rede, tendo esta um elevado número de nós. Estes protocolos são protocolos de referência que abordam a problemática da defesa pró-activa contra intrusões e uma avaliação e comparação entre ambos torna-se bastante interessante. Assim, permite-se retirar conclusões concretas a partir das avaliações efectuadas, utilizando critérios de medição que possibilitam a comparação dos resultados experimentados com resultados teoricamente esperados, relativamente ao comportamento destes protocolos quando sujeitos a ataques.

Portanto, as contribuições desta dissertação podem ser representadas por duas dimensões principais, que se apresentam da seguinte forma:

1. Plataforma genérica de simulação de RSSF:
    - Análise e avaliação de sistemas de encaminhamento;
    - Injecção de tipologias de ataques;
    - Simulações com sensores *TelosB* e *Micaz* em IEEE 802.15.4.
2. Avaliação dos protocolos de encaminhamento Clean-Slate e INSENS:

- Estudo em ambientes de larga escala;
- Estudo de protocolos pró-activos contra intrusões;
- Avaliação dos seus comportamentos sem ataques e face a ataques;
- Comparação entre os protocolos e entre resultados experimentais e teoricamente previstos.

1.4 Estrutura do documento

Os diversos capítulos deste documento de dissertação encontram-se organizados da seguinte forma:

- **Capítulo 1 (Introdução) -** Destaca os objectivos e contribuições delineadas para a dissertação, fazendo-se um posterior enquadramento associado à motivação para a realização dos objectivos, bem como para as contribuições esperadas após a elaboração da dissertação;

- **Capítulo 2 (Trabalho Relacionado) -** Apresenta uma visão do estado de arte e do trabalho relacionado com os objectivos desta dissertação, compreendendo três dimensões principais: o modelo de adversário considerado, estudo dos protocolos de encaminhamento seguro, e por fim uma abordagem aos simuladores de modo a escolher o simulador a utilizar.

- **Capítulo 3 (Modelação e Arquitectura) -** Elabora uma especificação da plataforma de simulação desenvolvida, oferecendo uma visão arquitectural de como é implementada. Inicialmente é feita uma abordagem aos aspectos de maior foco do JProwler [JPr] e à versão WiSeNet Simulator [dS11], sendo de seguida descrita a arquitectura da plataforma por explicação dos vários módulos que a constituem.

- **Capítulo 4 (Implementação da plataforma de simulação) -** Destacam-se os aspectos de implementação mais relevantes que vão ao encontro da especificação. Apresentam-se as interfaces gráficas criadas, os detalhes de implementação dos módulos desenvolvidos, bem como uma referência aos detalhes da versão WiSeNet Simulator, e por fim as alterações necessárias na configuração de base.

- **Capítulo 5 (Implementação dos protocolos de encaminhamento) -** Apresentam-se os detalhes de implementação dos protocolos Flooding, Clean-Slate e INSENS numa visão mais algorítmica, e por fim são descritos os vários ataques aos protocolos ao nível da implementação.

- **Capítulo 6 (Testes e Avaliação) -** Este capítulo apresenta os vários resultados obtidos para o conjunto de testes efectuados aos protocolos, sendo as medições realizadas sobretudo no que diz respeito à cobertura, fiabilidade, latência e energia.

- **Capítulo 7 (Conclusões) -** Apresentam-se as conclusões finais desta dissertação, levando em consideração os aspectos introdutórios. No final, são também apresentados os assuntos em aberto e o possível trabalho futuro.

- **Bibliografia**

- **Anexo A (Introdução às RSSF) -** Elabora uma breve introdução às redes de sensores sem fios. A leitura deste anexo não é obrigatória para a compreensão dos restantes capítulos, sendo apenas recomendada por oferecer uma visão geral e introdutória aos leitores que não possuam conhecimento prévio sobre este tipo de redes.

2

Trabalho Relacionado

2.1 Modelo de adversário

Os modelos de adversário são importantes na medida em que permitem definir o possível comportamento de um atacante perante um determinado sistema, pois um sistema sem uma especificação de adversário não pode ser considerado seguro. Porém, o avanço das tecnologias originou novas vulnerabilidades nos sistemas o que, por sua vez, leva à necessidade de considerar novos modelos de adversário. O modelo de atacante Dolev-Yao [DY83] é caracterizado pelo facto de um atacante apenas ter a capacidade de aceder e manipular as mensagens entre dois principais legítimos ao nível das comunicações, pois os ataques são externos o que não considera pressupostos de ataques por intrusão a nós. Este modelo é limitado face a novos cenários que têm surgido, como é o caso das próprias RSSF que estão fortemente sujeitas a ataques internos, e onde ainda ocorrem diferentes pressupostos das redes convencionais. Por conseguinte, surge a necessidade de aplicar uma nova visão do modelo de adversário quando se trata deste tipo de redes.

Para o efeito dos resultados finais pretendidos nesta dissertação especificou-se um modelo de adversário a ser considerado, contudo, anteriormente à especificação é exigível um entendimento acerca de como estes ataques podem constituir ameaças à segurança nos diferentes níveis da pilha. Isto é, os ataques podem incidir, essencialmente, ao nível físico/MAC e ao nível rede/encaminhamento.

2.1.1 Nível MAC

Os nós sensores contêm na sua pilha de camadas um nível designado por *Medium Access Control* (MAC), que se assemelha ao nível *Data-Link* da convencional pilha TCP/IP e se encontra sobre o nível físico. O nível MAC mantém o protocolo de controlo de acesso ao meio, que pelas particularidades das RSSF, como é o caso da baixa capacidade energética, deverá adequar-se a tais características. Isto dificilmente seria conseguido por intermédio dos protocolos tradicionais.

2.1.1.1 IEEE 802.15.4

A necessidade de uma especificação *standard* levou a que o IEEE adopta-se a norma IEEE 802.15.4 [41503]. Esta norma especifica o nível MAC e o nível físico para as LR-WPAN (*Low Rate Wireless Personal Area Networks*), sendo apropriada à construção de RSSF.

Ao nível físico, entre as várias frequências especificadas, a banda de frequências de 2.4 GHz é a adoptada para RSSF, permitindo uma taxa de transmissão de 250 Kbps. Tendo especialmente em foco o nível MAC, o protocolo de acesso ao meio proposto pela norma IEEE 802.15.4 define duas classes de dispositivos: *Full Function Devices* (FFD) e *Reduced Function Device* (RFD). Os primeiros desempenham um conjunto completo de operações, enquanto os RFD são dispositivos com funcionalidades restritas. A cada dispositivo poderá ser atribuído um dos seguintes papéis:

- coordenador da *Personal Area Network* (PAN): dispositivo de classe FFD que actua como controlador central de uma PAN e como *gateway* para outras redes;
- coordenador: dispositivo de classe FFD que executa encaminhamento de dados e funções de organização da rede;
- participante: dispositivo de classe FFD ou RFD que apenas comunica com os coordenadores.

A norma suporta ainda dois tipos de topologias da rede: a topologia em estrela e a topologia ponto-a-ponto. Na primeira, todas as comunicações usam o coordenador PAN como intermediário que se localiza no centro da rede. A segunda topologia permite formar uma rede *multi-hop*, uma vez que os coordenadores podem comunicar com outros que se encontrem no seu alcance de rádio ou que se encontrem fora, embora neste último caso a comunicação seja feita por coordenadores intermédios.

O protocolo MAC especificado pela norma define também dois modos de acesso ao canal que ficam ao critério do coordenador PAN, isto é, podem ser utilizados os modos com ou sem *beacon frame*. O modo com *beacon frame* sincroniza as comunicações e

inclui o mecanismo *slotted* CSMA/CA, o outro modo transmite os dados por *unslotted* CSMA/CA. O *beacon frame* descreve a estrutura de um *superframe*, o qual representa um determinado intervalo de tempo separado por um período activo, onde são permitidas transmissões, e por um período inactivo, em que os nós entram num estado de baixo consumo energético. O período activo divide-se em *slots*, os primeiros constituem o *Contention Access Period* (CAP) onde se disputa o acesso ao meio, e os restantes *slots*, embora opcionais, constituem o *Contention Free Period* (CFP) que é reservado pelo coordenador PAN para o acesso dedicado, ou seja, livre de contenção.

2.1.1.2 Ataques ao nível MAC (IEEE 802.15.4)

Enquanto ao nível físico os ataques consistem maioritariamente em perturbar o meio através do envio de sinais que causam interferências (ataques *jamming*), já os ataques ao nível MAC [MFM05] consistem tipicamente em perturbar as comunicações entre nós legítimos por outros meios. Ataques à própria comunicação por via externa podem ocorrer, como é o caso dos ataques: *eavesdropping*, *masquerading*, *spoofing*, *tampering* e replicação de mensagens. Para evitar alguns destes tipos de ataques, a norma fornece alguns serviços de segurança, embora ainda apresentem algumas fraquezas.

Os ataques por intrusão, específicos ao funcionamento do protocolo MAC no IEEE 802.15.4, repartem-se em duas principais categorias: os que actuam em concordância com a especificação da própria norma e os que recorrem a modificações na especificação.

Relativamente aos primeiros, um atacante que capture um nó facilmente realiza um ataque por inundação (ataque DoS), enviando uma grande quantidade de pacotes desnecessários que comprometem a disponibilidade e o bom desempenho da rede. Este ataque, embora não seja muito eficiente, dá a possibilidade ao atacante de alvejar os destinatários pretendidos e reduzir a sua capacidade energética. Outro possível ataque consiste em um atacante parametrizar o nó, em sua posse, para o uso do modo de extensão de vida da bateria. Este modo leva o algoritmo CSMA/CA a reduzir o expoente inicial de *backoff* desse nó, provocando assim uma probabilidade de acesso ao meio superior à dos nós legítimos, o que os obriga a esperar e a desperdiçar energia. Outra possibilidade, consiste no nó do atacante ser parametrizado de modo a efectuar o *Clear Channel Assessment* (CCA) do CSMA/CA uma só vez, ao invés das duas inicialmente propostas pela norma.

Quanto aos ataques por modificação da especificação, o atacante poderá controlar um dispositivo, manipulando totalmente o mecanismo CSMA/CA do protocolo, de forma a ter uma clara vantagem no acesso ao meio. Por exemplo, através do não incremento do expoente de *backoff*, gerador de números aleatórios dar preferência a

menores períodos de *backoff* ou simplesmente ignorá-los, e da redução do número de CCAs ou até da sua omissão. Por conseguinte, várias colisões podem surgir e algumas delas eventualmente ocorrerem com *acknowledges*, logo sucedem conflitos que levam ao desperdício de energia e largura de banda.

Importa ainda referir a existência de ataques que requerem o desenvolvimento de *hardware* dedicado permitindo não respeitar a especificação da norma. Estes ataques são potencialmente mais perigosos, no entanto, a relação custo/efeito não visa ser a melhor opção por parte dos atacantes.

2.1.2 Nível de rede/encaminhamento

Ao nível rede/encaminhamento os protocolos são desenvolvidos no sentido de facilitarem a comunicação entre dois nós sem uma ligação directa, ou seja, de forma a suportar uma comunicação *multi-hop*. A participação de uma grande parte dos nós da rede nos protocolos de encaminhamento é essencial para assegurar a existência de caminhos entre eles. No entanto, isto remete para o aumento das vulnerabilidades [YCW06], que podem ser exploradas pelos atacantes provocando uma frágil segurança destas redes. O meio de comunicação sem fios, a falta de uma infra-estrutura centralizada para serviços de segurança, os problemas do elevado processamento associado à utilização de criptografia assimétrica e a possível mobilidade dos nós, são alguns dos desafios que dificultam a resolução das questões de segurança.

Os protocolos de encaminhamento classificam-se em pró-activos e reactivos. Os protocolos pró-activos requerem uma regular troca de informação entre os nós, no sentido de cada nó manter a sua tabela de encaminhamento actualizada, cuja tabela contém a informação suficiente para esse nó alcançar qualquer outro nó da rede.

Relativamente aos protocolos reactivos, estes diferem dos anteriores porque a eventual troca de informação de encaminhamento apenas ocorre quando existem comunicações solicitadas. O procedimento base tipicamente considerado por este tipo de protocolos resume-se do seguinte modo: quando um nó origem pretende comunicar com um determinado nó de destino, ele envia um pacote *route request* (RREQ) para os nós ao seu alcance, e cada nó ao receber esse pacote propaga-o no caso de desconhecer como atingir o destino, caso contrário envia um pacote *route reply* (RREP) em direcção à origem. Cada pacote RREQ pode incluir o caminho desde a origem até ao nó que o recebeu, de modo a que o destino possa obter o caminho final e o possa entregar à origem através dos pacotes RREP.

2.1.2.1 Ataques ao nível de encaminhamento

Os ataques ao nível do encaminhamento [YCW06] podem ser desempenhados por duas classes de atacantes: *mote* e *laptop*. Atacantes da classe *mote* têm capacidades semelhantes a outros nós legítimos na rede, enquanto os atacantes da classe *laptop* têm dispositivos com recursos mais poderosos, permitindo porventura cobrir toda a rede de sensores. Cada ataque destina-se a uma das seguintes fases do protocolo de encaminhamento: descoberta de rotas, selecção de rotas e controlo de encaminhamento (após o estabelecimento de rotas).

1. Ataques à descoberta de rotas

   Relativamente aos ataques na fase de descoberta de rotas, estes ocorrem quando os nós se encontram a comunicar uns com os outros a fim de se conhecerem e permitirem descobrir e estabelecer caminhos entre eles. Como tal, resultam os seguintes ataques:

   - `Falsificação da informação de encaminhamento`: No caso de protocolos pró-activos, o nó sob a posse do atacante fornece informação de encaminhamento falsa que irá alterar incorrectamente as tabelas de encaminhamentos dos respectivos nós. Em protocolos reactivos, o atacante poderá manipular os pacotes RREQ e RREP por utilização, modificação ou remoção indevidamente;
   - *`Rushing`*: É normalmente destinado a protocolos reactivos e consiste, por exemplo, em o atacante, assim que receba um pacote RREQ, modificar e dispersá-lo por outros nós intermédios antes de novos pacotes RREQ (com a mesma origem e destino) chegarem a esses nós por outros caminhos. Isto constitui um ataque porque habitualmente qualquer nó apenas aceita o primeiro pacote RREQ recebido, descartando os restantes que neste caso seriam pacotes legítimos;
   - *`RREQ-Flooding`*: Este ataque é também habitualmente destinado a protocolos reactivos, tendo como objectivo permitir o atacante inundar a rede com pacotes RREQ desnecessários, o que acaba por originar um ataque DoS.

2. Ataques à selecção de rotas

   Estes ataques têm a finalidade de influenciar os nós legítimos a seleccionarem o nó malicioso para as suas rotas, permitindo o atacante receber uma grande parte das mensagens que circulam na rede. Apresentam-se os vários ataques em seguida:

- `Hello-Flooding`: Alguns protocolos requerem pacotes *Hello* para determinarem a relação de vizinhança entre os nós. Porém, se estamos perante um atacante de classe *laptop* esse pressuposto da relação de vizinhança é quebrado, pois o seu raio de alcance é muito maior e, como tal, os nós erradamente consideram-no vizinho;

- `Sinkhole`: Este ataque influencia os nós circundantes ao nó malicioso de modo a que estes o vejam como um nó bastante atractivo, pois ele tem a capacidade de anunciar, por exemplo, uma alta qualidade de encaminhamento para qualquer destino ou até mesmo indicando que é seu vizinho. Assim, os nós vizinhos terão preferência neste para estabelecerem as suas rotas, podendo mesmo, posteriormente, aplicar outro tipo de ataques;

- `Wormhole`: Este ataque consiste na cooperação entre dois nós maliciosos, os quais utilizam um canal disponível entre ambos e independente da rede, funcionando como um túnel, por onde são transmitidas as mensagens em condições superiores às da própria rede. Portanto, os nós vizinhos erradamente poderão seleccionar estes nós para encaminhar mensagens por acreditarem ser a melhor escolha;

- `Sybil`: O nó do atacante pode anunciar aos seus vizinhos várias identidades que correspondem a falsos nós. Assim, os nós legítimos tratam os falsos como nós distintos e válidos, sendo então considerados para o estabelecimento de rotas, quando na verdade essas rotas utilizam sempre o próprio nó atacante.

3. Ataques ao controlo do encaminhamento

   Se um nó contém uma rota estabelecida para um determinado destino, a qual inclui um nó malicioso, alguns ataques podem agora ser realizados por esse nó:

   - `Blackhole`: O nó do atacante descarta todos os pacotes recebidos de uma determinada origem, impedindo que eles sejam propagados para outros nós;

   - `Selective-Forwarding`: À semelhança do ataque anterior o nó atacante descarta pacotes, embora de forma selectiva.

   - `Spam`: O atacante gera um elevado número de mensagens totalmente desnecessárias, que visam consumir e desperdiçar os recursos dos nós da rede.

2.1.3 Especificação do Modelo de adversário

No que diz respeito à especificação do modelo de adversário a ser utilizado, tendo em vista os objectivos desta dissertação, são considerados adversários que poderão desempenhar ataques externos e internos. Relativamente aos ataques externos, admite-se que um atacante terá capacidade para desempenhar ataques contemplados pelo modelo de Dolev-Yao, sendo estes tratados pela camada de segurança TinySec [KSW04] subjacente ao protocolo de encaminhamento.

Em relação aos ataques internos, consideram-se ataques que ocorrem por intrusão, que são ataques incluídos no modelo bizantino, no entanto excluem-se os ataques por replicação. A detecção de ataques por replicação [PPG05] de nós incorrectos envolve alguma complexidade e, tipicamente, os sistemas de encaminhamento têm alguma dificuldade em lidar com eles, necessitando frequentemente de mecanismos adicionais.

Tendo como referência os ataques anteriormente mencionados e enquadrando-os na pilha de camadas, importa referir que são descartados ataques que incidam ao nível MAC e físico, uma vez que não é relevante para o objectivo final avaliar a repercussão de tais ataques a esse nível. Portanto, o principal foco será sobre o nível de rede/encaminhamento.

Mais concretamente, nesta especificação de modelo de adversário a ser considerado abarcam-se as várias tipologias de ataques ao encaminhamento já apresentadas anteriormente, as quais são tidas em conta pelos protocolos de encaminhamento seguro apresentados em 2.2, incluindo a elaboração de uma análise comparativa das suas contramedidas suportadas. Porém, somente os ataques *hello-flooding*, *wormhole*, *sinkhole*, *sybil*, *blackhole* e *selective-Forwarding* são utilizados na fase de teste e avaliação, devido ao facto dos restantes ataques não serem aplicáveis aos protocolos pretendidos para a análise e avaliação (por exemplo porque apenas se aplicam a protocolos reactivos) ou por terem características muito específicas quando aplicados a um determinado protocolo, dificultando a utilização de um critério de comparação válido.

2.1.4 Sumário

Inicialmente abordou-se o nível MAC das RSSF, analisando-se particularmente a norma IEEE 802.15.4 para as LR-WPANs. O protocolo MAC tem como objectivo suportar as ligações entre nós através da transmissão e recepção por radiofrequência e coordenar as condições de acesso ao meio para que seja livre de colisões. A norma permite os tipos de topologia em estrela e ponto-a-ponto, e o modo de acesso ao canal com ou sem *beacon frame*.

Quanto às hipóteses de adversário ao nível MAC, incluem-se ataques às comunicações por via externa e ataques internos por intrusão aos nós provocando a incorrecção ao funcionamento do protocolo MAC (IEEE 802.15.4). Estes ataques por intrusão contemplam os que seguem a especificação (resumem-se a ataques do tipo DoS ou por parametrização incorrecta do protocolo CSMA/CA), e os que não a seguem, sendo o protocolo CSMA/CA manipulado totalmente sem respeitar as condições de acesso ao meio. Ao nível rede/encaminhamento, os respectivos protocolos também são susceptíveis de ataques, e podem ser classificados como protocolos pró-activos ou reactivos. Cada ataque insere-se numa das possíveis fases de funcionamento do protocolo: fase de descoberta de rotas (falsificação da informação de encaminhamento, *rushing* e *RREQ-flooding*), fase de selecção de rotas (*sinkhole*, *wormhole*, *hello-flooding* e *sybil*) e a fase de controlo do encaminhamento (*blackhole*, *selective-forwarding* e *spam*).

O modelo de adversário considerado admite ataques externos (contemplados pelo modelo Dolev-Yao) e internos, ou seja, por intrusão mas exceptuando ataques por replicação de nós incorrectos. No âmbito da presente dissertação consideram-se especificamente os ataques que incidem ao nível do encaminhamento, os quais são tidos em conta pelos protocolos de encaminhamento seguro posteriormente analisados. Para efeitos da fase de testes e avaliação os seguintes ataques serão utilizados: *hello-flooding*, *wormhole*, *sinkhole*, *blackhole* e *selective-forwarding*.

Este modelo de adversário apresentou um estudo breve acerca da norma IEEE 802.15.4, o que vai ao encontro da contribuição que diz respeito ao facto de a plataforma permitir simular dispositivos que operam neste tipo de ambientes. O estudo dos ataques é também relevante para a injecção de tipologias de ataques na plataforma de simulação, pois é necessário um conhecimento dos ataques ao nível do encaminhamento que permita conceber o módulo responsável por injectá-los e definir os respectivos comportamentos. Para as contribuições relativas à avaliação dos comportamentos dos protocolos e comparação entre resultados experimentais e teóricos, é determinante a especificação do modelo de adversário, pois ele define os ataques quem devem ser considerados na análise e avaliação dos protocolos.

2.2 Protocolos de encaminhamento seguro

Tendo em conta o modelo de adversário anteriormente especificado deparamo-nos com a necessidade de acrescentar mecanismos de segurança que tornem os protocolos de encaminhamento resilientes face aos ataques. Nesse sentido existem vários protocolos de encaminhamento que dispõem de contramedidas e que asseguram o correcto encaminhamento de dados, independentemente da tentativa de certos ataques.

Os protocolos de encaminhamento seguro analisados são: o Clean-Slate [PLGP06], o INSENS [JD02] (*Intrusion-tolerant routing protocol for Wireless Sensor Networks*) e, por fim, o SIGF [WFSH06] (*Secure Implicit Geographic Forwarding*). Estes protocolos são distintos entre si, pela forma como estão implementados e, sobretudo, pelo seu desempenho e garantias face a possíveis tipologias de ataques por eles consideradas. A concepção de protocolos de encaminhamento seguro é uma tarefa muito difícil, e estes protocolos mencionados apresentam-se como referências actuais que se preocupam em aliar as exigências de segurança às limitações dos dispositivos, razões pelas quais foram seleccionados como alvo de estudo.

Importa referir que nesta dissertação pressupõe-se que existe um esquema de distribuição de chaves previamente implantado e disponível para qualquer um dos protocolos de encaminhamento que são apresentados de seguida.

2.2.1 Clean-Slate

O Clean-Slate [PLGP06] é um protocolo de encaminhamento que oferece segurança e eficiência, sendo altamente resiliente perante adversários activos, pois preocupa-se em garantir a entrega de mensagens transmitidas entre nós. Este protocolo foi concebido tendo por base três princípios: prevenção, detecção/recuperação e resiliência.

Como pressupostos iniciais, o protocolo assume a presença de uma autoridade sobre a rede que possui uma chave pública e privada, sendo que essa autoridade entrega uma identidade certificada a cada um dos nós da rede, bem como um conjunto de valores de desafio também certificados. Assume-se ainda que os nós têm pouca mobilidade.

Tendo em consideração estes pressupostos iniciais, numa visão geral, o protocolo atribui dinamicamente um endereço único a cada nó da rede e constrói as respectivas tabelas de encaminhamento (entradas da tabela mapeiam prefixos de endereços para um determinado *next-hop*) através de um algoritmo de agrupamento recursivo que actua deterministicamente. Este algoritmo previne (*princípio da prevenção*) ataques à informação de encaminhamento e limita um nó comprometido de actuar ilegitimamente. Após estabelecida esta fase, os nós passam a encaminhar pacotes por intermédio de um mecanismo de encaminhamento resiliente que inclui técnicas para detecção/recuperação e eliminação de nós maliciosos (*principio da detecção/recuperação*). A resiliência (*princípio da resiliência*) é obtida, uma vez que o emissor pode controlar o caminho por onde o pacote segue. De facto, todos os passos do protocolo preocupam-se em proteger ataques internos e externos, de modo a garantir uma alta disponibilidade para efectuar o encaminhamento.

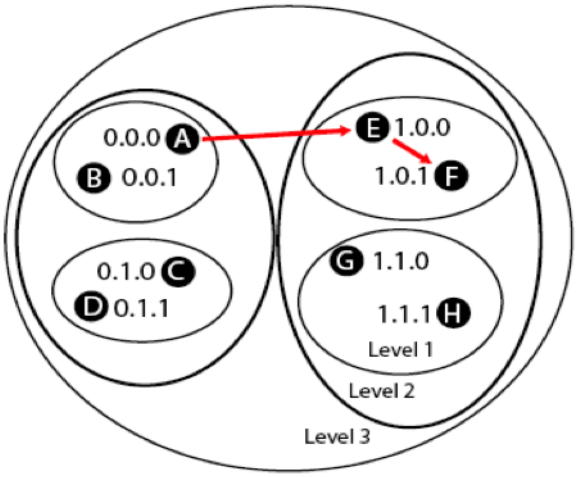


Figura 2.1: Visão do agrupamento recursivo.

No que diz respeito ao processo de configuração para realizar o encaminhamento, o algoritmo de agrupamento recursivo, ilustrado na figura 2.1, enquadra-se na ideia da rede poder ser vista em grupos de nós, onde inicialmente cada nó é o próprio grupo. Por conseguinte, cada grupo envia uma proposta de junção para o grupo de menor tamanho (grupo com menor número de nós), e se ocorrer igualmente uma proposta no sentido inverso reúnem-se as condições para ambos os grupos formarem um novo e único grupo, o processo repete-se até convergir num só grupo para toda a rede. Por cada junção entre os grupos G e G', os endereços dos nós são revistos e as tabelas de encaminhamento actualizadas, qualquer nó do grupo G passa a ter pelo menos um *next-hop* para atingir o grupo G' e vice-versa. Por questões de segurança, em cada estado do algoritmo os grupos devem ser autenticados usando uma GVT (*Group-Verification Tree*), a qual se assemelha a uma *hash tree*. Para que os nós conheçam correctamente os vizinhos em seu redor, o algoritmo requer ainda que estes, inicialmente, executem um protocolo seguro para a descoberta de vizinhos, por utilização dos certificados de identificadores pré-adquiridos pelos nós, impedindo assim um nó externo injectar um novo identificador ou que um nó comprometido o altere.

Quanto ao encaminhamento das mensagens, este é realizado através dos endereços e tabelas de encaminhamento determinados anteriormente, no entanto, o mecanismo de encaminhamento resiliente, que consiste em manter vários caminhos para os destinos, vem oferecer maiores garantias de entrega das mensagens. Os múltiplos caminhos são obtidos através do próprio algoritmo de agrupamento recursivo, que passa a estender as tabelas de encaminhamento com vários *next-hops* possíveis para cada grupo de destino, e assim o emissor poderá seleccionar o caminho pretendido. Uma outra optimização consiste em o nó seleccionar como *next-hop* aquele que está mais próximo

do grupo de destino, ou seja, baseado na distância.

Para a detecção e recuperação de comportamentos maliciosos recorre-se às seguintes técnicas: utilização de uma GVT para a detecção de inconsistências ou desvios ao correcto comportamento do algoritmo de agrupamento recursivo; esquema de detecção de duplicados que impede os nós de se apresentarem com múltiplas identidades (por exemplo utilizando certificados legítimos de outros nós) ou tentarem agrupar-se com vários grupos simultaneamente; e por fim a técnica *Honeybee* que remove nós maliciosos que sejam detectados na rede.

O protocolo fica, assim, de certo modo protegido face aos seguintes ataques: falsificação da informação de encaminhamento e ataques DoS generalistas (aplicados ao encaminhamento) são resolvidos pelo determinismo do algoritmo de agrupamento recursivo; ataque de *sybil* é tratado pelo protocolo seguro de descoberta de vizinhos; ataque de *sinkhole* não ocorre se o *next-hop* é seleccionado não tendo por base a distância, e mesmo que seja seleccionado por distância o protocolo inclui um mecanismo de segurança para o anúncio das distâncias; ataque de *blackhole* é, de certo modo, impedido desde que as rotas não sejam seleccionadas por um critério baseado na distância; ataque por *wormhole* é parcialmente tratado através do encaminhamento resiliente, no entanto exige outro tipo de técnicas adicionais para o impedir na totalidade. Um ataque por replicação é resolvido por um algoritmo de detecção de réplicas que não está incluído na concepção de base, como tal esse ataque não é tido em conta nesta dissertação.

2.2.2 INSENS

O INSENS [JD02] é um protocolo para encaminhamento seguro que visa tratar, de certa forma, o problema das limitações dos recursos e das intrusões numa RSSF com uma topologia assimétrica. As limitações dos nós sensores são contornadas através de um forte contributo dado pela estação-base, pois esta tem recursos com menores limitações. Nesse sentido, é a estação-base que irá processar a informação acerca da topologia da rede, de modo a criar as tabelas de encaminhamento e distribuí-las respectivamente por cada nó.

O INSENS para tratar os ataques por intrusão, não se baseia em nenhum mecanismo de detecção de intrusões, uma vez que este é um processo complexo para aplicar neste tipo de redes. A solução passa simplesmente por tolerar intrusões através da introdução de caminhos redundantes, pelos quais se enviam as mensagens, bem como oferecer as condições necessárias para suportar o princípio de que um nó comprometido afecta apenas uma pequena porção da rede, permitindo a rede funcionar independentemente de ocorrerem tais ataques.

De facto, pretende-se tolerar intrusões mas requer-se ainda que o protocolo proteja contra ataques DoS que inundam pacotes por toda a rede, e ataques de falsificação da informação de encaminhamento. O ataque DoS é impedido porque apenas a estação-base tem a possibilidade de fazer *broadcast* a toda a rede, os nós sensores não o poderão fazer porque para enviarem mensagens deverão primeiro comunicar à estação-base que funciona assim como um filtro, impedindo que os nós inundem a rede com mensagens. O outro ataque é resolvido através da autenticação das mensagens, em que cada nó partilha apenas uma chave com a estação-base. Devido à limitação dos recursos é utilizada criptografia simétrica.

Ao usar mecanismos de criptografia, em que cada nó apenas contém a sua chave partilhada com a estação-base, e limitando a possibilidade de *flooding* reduzem-se os danos provocados por intrusões e a porção da rede que é afectada. A estrutura geral da configuração do protocolo para a descoberta de rotas está dividida em três fases:

1. *Route Request* – A estação-base provoca uma inundação limitada de mensagens, solicitando a informação de todos os nós da rede alcançáveis;

2. *Route Feedback* – Todos os nós enviam a sua informação local para a estação-base, em resposta à solicitação;

3. *Routing Table Propagation* – A estação-base calcula as tabelas de encaminhamento para cada nó e envia-as respectivamente.

Relativamente à primeira fase, os seguintes ataques são susceptíveis de serem executados por um nó atacante: fazer passar-se por estação-base; inserir um falso caminho na mensagem; e não enviar a mensagem ou fazer um ataque DoS com esta. Para impedir estes ataques utiliza-se o mecanismo *one-way sequences* e o algoritmo criptográfico MAC para a integridade. Um ataque de *rushing*, em que o atacante envia para os nós uma mensagem *request* antes da mensagem válida enviada pela estação-base, apenas actua numa zona local próxima do intruso. Na verdade, esta fase não deixa de ter algumas vulnerabilidades, contudo não põem em causa a rede na sua totalidade.

Quanto à segunda fase, utiliza-se também o algoritmo MAC para garantir a integridade da mensagem ao longo do caminho inverso. É ainda possível criar alguns ataques, tais como o atacante: enviar múltiplas mensagens de *feedback* para o mesmo nó, o que constitui um ataque DoS; adquirir informação topológica; e desviar as mensagens *feedback* para nós errados. Para limitar o ataque DoS são utilizados dois mecanismos de defesa, no primeiro não são propagadas mensagens de *feedback* duplicadas, no segundo é controlada a taxa de emissão das mensagens de forma a não congestionar o caminho a partir do nó malicioso até à estação-base. Já o ataque para adquirir informação topológica evita-se através da cifra de dados que garante confidencialidade. O desvio

de mensagens nesta fase é encarado de tal modo que não se considera problemático o caminho pelo qual a informação chega à estação-base.

Na última fase, através dos mecanismos de segurança integrados nas fases anteriores, permite-se que a estação-base possa recolher a informação correcta acerca da topologia da rede, sendo ainda verificada a consistência dos dados.

2.2.3 SIGF

Este protocolo [WFSH06] de encaminhamento seguro procura oferecer uma solução que tem uma forte consideração pelas restrições associadas aos recursos num nó de uma RSSF, desempenhando assim mecanismos de segurança que se adequam às condições de hostilidade da rede. Os seus objectivos, teoricamente esperados, passam por garantir uma boa taxa de entrega de mensagens ao destino, baixo *overhead*, baixa latência, visando assim um bom desempenho da rede face a eventuais ataques. Para tal, a adaptabilidade do protocolo SIGF às condições é conseguida através de uma família de configurações baseadas num *trade-off* entre segurança e estado (SIGF-0, SIGF-1, SIGF-2), sendo o SIGF-0 o extremo em que o protocolo praticamente não mantém estado e oferece menor segurança, e o SIGF-2 o extremo oposto onde os nós partilham informações, permitindo maior segurança.

Qualquer configuração do SIGF tira partido de um protocolo de encaminhamento que está na base, este designado por *Implicit Geographic Forwarding* (IGF) e que não tem qualquer preocupação prévia com questões de segurança. O IGF é um protocolo totalmente reactivo, pois efectua as decisões de forma não determinística e dinâmica no momento em que é desempenhado o protocolo de nível MAC, eliminando assim a necessidade de mensagens extra para o encaminhamento e da manutenção de informação de encaminhamento, o que impede logo à partida alguns tipos de ataques. Portanto, um ataque do tipo falsificação de informação de encaminhamento nunca ocorre, já os ataques *wormhole* e *HELLO-flooding* também não são perturbadores devido ao encaminhamento dinâmico que é efectuado. Contudo, o IGF não protege contra certos ataques na vizinhança, pois o nó emissor não tem capacidade de distinguir a legitimidade dos seus nós vizinhos. Um atacante pode provocar o nó emissor a escolhê-lo como *next-hop*, tal que o atacante perturbaria o encaminhamento. É nestes termos que a extensão SIGF se insere, preservando as propriedades do IGF e oferecendo segurança nas vizinhanças dos nós. De seguida, serão analisados ataques de encaminhamento que não são protegidos pelo IGF mas que são tratados por algum dos protocolos SIGF.

O ataque de *rushing* é um dos ataques que o IGF não protege, o atacante poderá utilizá-lo com o objectivo de ver o seu nó ser seleccionado, que por conseguinte permite-lhe efectuar um ataque de *blackhole*. No entanto, este ataque pode ser protegido através

do primeiro protocolo com segurança, o SIGF-0, o qual se baseia em contramedidas probabilísticas, pois determina o *next-hop* de forma dinâmica e não determinista. Assim, o protocolo SIGF-0 não dá garantias de segurança, mas em contrapartida minimiza o consumo de recursos.

Já o ataque de *sybil* não é tratado pelo SIGF-0, no entanto o protocolo SIGF-1, que herda as propriedades do SIGF-0, consegue resolvê-lo. O SIGF-1 vem reduzir ainda mais as possibilidades de ser seleccionado um nó atacante para *next-hop*, o que se deve ao facto de ser armazenado localmente o estado corrente e as estatísticas dos vizinhos, de forma a derivar as suas reputações e assim seleccionar o melhor nó para enviar dados.

Quanto a um ataque do tipo DOS, considera-se que o atacante realiza-o sobretudo de uma forma menos intrusiva e sem recorrer a ataques *jamming*, apenas por execução parcial do protocolo IGF (em particular trata-se do *RREQ-Flooding*) de modo a provocar aos nós vizinhos o desperdício de energia. Os protocolos referidos anteriormente são ainda vulneráveis, uma vez que se baseiam em probabilidades. Para tal, o protocolo SIGF-2 vem solucionar essas limitações, oferecendo agora alguma protecção para um ataque DoS deste tipo, exigindo um maior esforço. Isto é alcançado por se manter um estado partilhado entre vizinhos recorrendo ao uso de operações criptográficas que garantem autenticação, confidencialidade, integridade e frescura dos segredos. Assim, o facto de o atacante não possuir as devidas chaves criptográficas atenua a possibilidade deste injectar mensagens consideradas válidas para o resto da rede. De notar que as medidas de segurança existentes não previnem completamente ataques por intrusão, no entanto as técnicas de SIGF-0 e SIGF-1 limitam-nos de certa forma.

2.2.4 Análise comparativa

Com o estudo dos três protocolos de encaminhamento seguro, anteriormente descritos, foi possível clarificar as diferenças entre eles, no que diz respeito ao modo como abordam a tipologia de ataques ao encaminhamento, sobretudo considerando a possibilidade de intrusão. Desta forma podemos agora efectuar uma visão geral e uma análise comparativa entre os três protocolos em função dos ataques presentes no modelo de adversário proposto na secção 2.1.3, sendo indicado o mecanismo utilizado para defender cada ataque. Ressalte-se que o ataque por replicação encontra-se nessa análise comparativa, contudo apenas em termos de completude, pois não está incluído no modelo de adversário considerado.

A análise comparativa está representada através da tabela 2.1. Note-se que na tabela estão destacados os principais ataques específicos ao encaminhamento, os quais podem ser conseguidos por um atacante externo ou interno, contudo, nem todos os

protocolos estão completamente focados e preparados para as hipóteses dos ataques serem despoletados através de intrusão ou replicação, como é o caso do SIGF. Assim, a tabela inclui adicionalmente a informação sobre qual o comportamento geral do protocolo perante um ataque por intrusão ou replicação.

Tabela 2.1: Comparação dos protocolos de encaminhamento seguro em função dos ataques no modelo de adversário considerado.

	Ataques/Protocolos	**Clean-Slate** (pró-activo)	**INSENS** (pró-activo)	**SIGF** (reactivo)
Ataques específicos ao encaminhamento	**Falsificacação da informação de encaminhamento**	Protege com o algoritmo de agrupamento recursivo	Protege através de *one-way authentication* e mecanismos de integridade	Nunca ocorre (sem tabelas de encaminhamento)
	Rushing	Protege se o *edge node* for escolhido por distância	Tolerado (efeitos são locais)	Protege com o SIGF-0
	RREQ-Flooding	Protocolo não se baseia neste tipo de mensagens	Protege através de *one-way sequences*	Protege com o SIGF-2
	Hello-Flooding	Protege com o protocolo seguro de descoberta de vizinhos	Dificuldade em proteger este tipo de ataque	Ataque não ocorre (encaminhamento dinâmico)
	Sinkhole	Protege se as rotas não se baseiam em distâncias	Protege através de caminhos redundantes	Protege com o SIGF-0
	Wormhole	Protege parcialmente através do encaminhamento resiliente	Protege através do cáculo de rotas pela estação-base e dos caminhos redundantes	Ataque não ocorre (encaminhamento dinâmico)
	Sybil	Protege com o protocolo seguro de descoberta de vizinhos	Protege através do par de chaves entre nó e estação-base	Protege com o SIGF-1
	Blackhole ***Selective Forwarding***	Protege se as rotas não se baseiam em distâncias	Protege através de caminhos redundantes	Protege com o SIGF-0
	Ataque por intrusão	Usa algoritmos para detecção e recuperação	Tolera sobretudo por redundância de caminhos	Segurança do SIGF2 equivale ao SIGF1
	Ataque por replicação	Usa detecção de replicações	Não é tratado particularmente	Não é tratado

De entre os protocolos de encaminhamento estudados, os sistemas Clean-Slate e INSENS são representativos de protocolos pró-activos, sendo relevantes como sistemas de referência que abordam a problemática de defesa pró-activa contra intrusões. Estes sistemas foram estudados com base em implementações específicas e foram objecto de análises teóricas publicadas por vários autores. Uma avaliação completa com as tipologias de ataques abordadas em 2.1.2.1, tendo por base uma avaliação experimental das contramedidas para tolerância a intrusões não está porém coberta na bibliografia existente. Esta avaliação experimental numa base de simulação onde essas

tipologias de ataques possam ser configuradas por injecção de código malicioso revela-se importante, não apenas para confirmar ou verificar de forma sistemática a eficácia das referidas contramedidas, mas também para avaliar o seu impacto em condições de operação da rede que se possam aproximar a condições reais.

O SIGF sendo um protocolo reactivo foi também estudado para que não fosse restringida a visão dos protocolos de encaminhamento seguro apenas a protocolos pró-activos. No entanto, por não seguir a linha de orientação dos protocolos pró-activos este protocolo acaba por não constituir critérios válidos de comparação, logo não foi alvo de implementação.

2.2.5 Sumário

Os protocolos admitidos para estudo foram o Clean-Slate, INSENS e o SIGF. O Clean-Slate foi concebido para cumprir três princípios: prevenção, detecção/recuperação e resiliência. O primeiro princípio através de um algoritmo de agrupamento recursivo que se baseia na ideia de a rede ser vista em grupos de nós, efectuando a junção desses grupos e simultaneamente estabelecendo o endereçamento e as tabelas de encaminhamento dos nós. O segundo princípio através de algoritmos de detecção e recuperação. Com encaminhamento resiliente, o emissor pode controlar o caminho pelo qual deseja enviar o pacote, garantindo o princípio da resiliência.

O protocolo INSENS considera intrusões e as limitações dos recursos através da tolerância a intrusões, por redundância de caminhos e através do forte contributo oferecido pela estação-base, respectivamente. Neste protocolo um nó comprometido afecta apenas uma mínima parte da rede, e as restrições à comunicação evitam que os nós façam *broadcast* para toda a rede. A descoberta de rotas está dividida em três fases: *route request*, *route feedback* e *routing table propagation*.

O SIGF é um protocolo reactivo que foi concebido sobre o IGF e consiste numa família de configurações baseadas num *tradeoff* entre segurança e estado (SIGF-0, SIGF-1, SIGF-2), onde o SIGF-0 não mantém estado e oferece menor segurança e o SIGF-2 contém estado e permite maior segurança. O IGF por si só resolve já alguns tipos de ataques, no entanto o SIGF-0 resolve os ataques *rushing* e *blackhole*, o SIGF-1 trata o ataque de *sybil* e o SIGF-2 trata ataques DoS.

Por fim, é feita uma análise comparativa utilizando o modelo de adversário proposto, o que expõe uma visão geral sobre como cada protocolo dá resposta a cada ataque.

Este estudo dos sistemas de encaminhamento seguro é importante para as contribuições delineadas e que dizem respeito à avaliação experimental dos protocolos de encaminhamento Clean-Slate e INSENS. Dá-se a entender o comportamento geral de

protocolos que abordam a defesa pró-activa contra intrusões, ao estudar o protocolo SIGF, que segue uma abordagem reactiva, consegue-se diferenciar as características que distinguem os pró-activos dos reactivos. Por outro lado, permite-se também obter uma noção das implicações que podem ocorrer quando os protocolos operam em redes de larga escala, o que será experimentalmente verificado. O estudo dos mecanismos de segurança destes protocolos e a análise comparativa da resiliência que foi elaborada serão suporte para realizar a comparação entre os resultados experimentais e os teoricamente previstos, tal como pretendido nas contribuições da dissertação.

2.3 Simuladores

A tarefa de análise e avaliação dos comportamentos previstos de uma determinada prototipagem numa RSSF torna-se bastante complicada quando estamos perante um ambiente experimental que na sua totalidade é realista. Um dos aspectos que mais pesa é nomeadamente a escalabilidade, na medida em que num ambiente de grande escala será extremamente indesejável efectuar várias alterações comportamentais durante a fase de desenvolvimento.

Nestes pressupostos, estão disponíveis várias ferramentas de simulação e emulação que são fundamentais durante a fase anterior à implementação. Estas ferramentas criam os ambientes apropriados para testar e validar qualquer tipo de protocolo ou aplicação. As ferramentas distinguem-se entre si pelas suas características e na forma como contribuem para oferecer uma boa qualidade de simulação ou emulação do comportamento de uma rede real. Naturalmente, o utilizador destas ferramentas pretende um comportamento que se assemelhe tanto quanto possível ao comportamento de uma RSSF real.

Com vista à escolha de um simulador de base com as características adequadas aos objectivos propostos foi desempenhada uma abordagem a várias ferramentas de simulação, de modo a compreender as diversas características envolventes e a obter uma base de comparação crítica entre as várias. De seguida, são descritos os simuladores aos quais foi dada uma especial atenção: TOSSIM, Freemote e JProwler.

2.3.1 TOSSIM

TOSSIM [Lev],[LL] é um simulador e emulador de eventos discretos direccionado para RSSF, e que suporta o sistema TinyOS. Uma aplicação desenvolvida em TinyOS para executar num sensor real pode ser compilada directamente para o TOSSIM. A sua funcionalidade de emulação está limitada a um determinado tipo de *hardware*, e o tipo de

sensores em questão são os *Micaz Motes*. Associada ao sistema TinyOS está a linguagem NesC, porém o TOSSIM suporta duas linguagens para interacção dinâmica com a simulação, são elas as linguagens Python e C++.

O seu principal objectivo passa por oferecer uma forte fiabilidade de simulação, permitindo aos utilizadores fazerem testes e análises por intermédio de um ambiente de controlo sobre a simulação de uma topologia de rede, que pode ser estática ou dinâmica. O simulador permite topologias de rede com escalabilidade um pouco limitada devido à elevada fiabilidade que se espera em cada nó. Quanto ao modelo de consumo energético, este é possível mas numa outra versão, designada PowerTOSSIM. Relativamente aos modelos de rádio, existe o modelo simples, onde não ocorrem perdas, e o modelo de perda de dados que recorre a operações maioritariamente probabilísticas.

A inexistência de heterogeneidade das aplicações nos nós, ou seja, todos os nós terão de desempenhar o mesmo papel, executando o mesmo código; a simplicidade dos modelos de rádio e a complexa usabilidade são algumas das desvantagens presentes nesta ferramenta. Importa ainda referir que uma possível extensão da ferramenta pode tornar-se um processo algo complexo, e que existe uma interface gráfica adicional ao TOSSIM, designada TinyViz, que facilita o processo de interacção entre o utilizador e o simulador/emulador.

2.3.2 Freemote

O Freemote [MKKW08] é uma ferramenta para simulação e emulação baseada em Java e destinada a RSSF, com a particularidade de permitir criar um ambiente que envolva nós simulados e nós reais numa única rede. A sub-rede simulada e a sub-rede real estão interligadas através de um *bridge node*, e desta forma é possível criar uma base de experimentação que suporta ambas as dimensões. O emulador está preparado para suportar *Java Motes*, contudo os nós reais podem ser quaisquer sensores baseados na norma IEEE802.15.4 (*JMotes*, *Micaz*, etc.).

Os nós suportam heterogeneidade das aplicações e ainda dividem-se em três camadas independentes: aplicação, encaminhamento e a camada física. Relativamente à escalabilidade, o Freemote suporta larga escala, permitindo acima de 10.000 nós, os quais podem apresentar características de mobilidade.

Alguns problemas bem presentes nesta ferramenta são a ausência de um modelo de consumo de energia, modelo de rádio demasiado simples, não assumindo a existência de obstáculos, e o comportamento da simulação é relativamente limitado comparativamente ao comportamento esperado numa experimentação real. De facto, a fiabilidade da simulação é moderada, levando a que o Freemote não seja uma ferramenta com as características desejadas para uma análise crítica ao nível do desempenho.

O Freemote incorpora uma interface gráfica que torna visível a topologia da rede e oferece uma simples usabilidade e extensibilidade da ferramenta.

2.3.3 JProwler

O JProwler [JPr] é um simulador de eventos discretos implementado em Java para RSSF que podem apresentar uma topologia de características dinâmicas e de alta escalabilidade. Os nós simulados estão estruturados por camadas (não na totalidade) e podem desempenhar diversos papéis, pois possuem heterogeneidade ao nível da camada aplicação. Este nível aplicação baseia-se em eventos como na *framework* TinyOS. Quanto ao nível da camada MAC, o simulador fornece apenas um protocolo centrado nos sensores *Mica Motes* apropriados a aplicações TinyOS.

Em relação aos modelos incluídos, são mantidos importantes aspectos ao nível da comunicação e aplicação, nomeadamente o modelo de rádio simula a transmissão, propagação e recepção de dados assumindo a eventual existência de colisões através de operações em modo determinístico e probabilístico. A ausência de um modelo de consumo energético é um aspecto significativo, no entanto, a fácil extensibilidade da ferramenta permite que esse problema possa ser resolvido. O JProwler oferece uma rápida e acessível forma de prototipagem, possuindo uma simples interface gráfica que permite visualizar a topologia da rede. As capacidades deste simulador são muito adequadas a análises de desempenho ao nível da comunicação, encaminhamento ou da aplicação.

2.3.4 Escolha do Simulador

Os simuladores têm características bem definidas que servem como fonte de entrada para uma análise comparativa, tendo em conta os objectivos propostos. Uma vez que o ambiente estudado envolve uma larga escala e não tem como objectivo aplicar uma base de experimentação real, pretende-se uma ferramenta que no essencial seja de simulação, mantendo uma alta fiabilidade. Isto porque serão feitas análises de desempenho ao nível do encaminhamento, onde é importante obter comportamentos próximos dos reais. Um modelo de consumo energético também será um factor relevante, bem como a heterogeneidade da aplicação nos nós. São ainda valorizados outros critérios, tais como a usabilidade e extensibilidade.

Na tabela 2.2 verifica-se que a ferramenta JProwler enquadra-se nestes requisitos, contudo, o modelo de energia não está presente, o que não será um problema tão significativo, devido à fácil extensibilidade. Quanto às outras ferramentas, o Freemote tem a grande desvantagem de ter uma baixa fiabilidade, sobretudo, devido ao seu simples

Tabela 2.2: Visão comparativa das três ferramentas.

Critérios	TOSSIM	Freemote	JProwler
Simulação	Sim	Sim	Sim
Emulação	Sim	Sim	Não
Escalabilidade	Média(com limitações)	Alta	Alta
Modelo de consumo energético	Sim (PowerTOSSIM)	Não	Não
Modelo de rádio	Maioritariamente probabilístico	Simples(não admite obstáculos)	Determinístico e probabilístico
Rede dinâmica/estática	Dinâmica ou estática	Dinâmica ou estática	Dinâmica ou estática
Tipos de sensores	Mica Motes	JMotes(emulação) IEEE802.15.4(reais)	Mica Motes
Baseado em eventos	Sim	Sim	Sim
Heterogeneidade (aplicação)	Não	Sim	Sim
Tipo de estruturação	Por componentes	Por camadas	Por camadas
Fiabilidade da simulação	Alta	Média	Alta
Linguagem	NesC, Python e C++	Java	Java
Utilidade em análise de resultados	Boa	Fraca	Boa
Interface Gráfica	Sim (TinyViz)	Sim	Sim
Usabilidade	Complexa	Simples	Simples
Extensibilidade da ferramenta	Complexa	Simples	Simples

modelo de rádio e, por conseguinte, torna-se pouco adequado para verificação e análise de resultados. Já o TOSSIM é desvalorizado, especialmente devido à sua limitação de escalabilidade e por não permitir heterogeneidade nos nós, a sua complexidade também foi tida em conta. Assim sendo, o JProwler é o simulador de base escolhido.

2.3.5 Sumário

A análise e verificação numa RSSF pode ser uma tarefa complicada, como tal, existem diversas ferramentas com o objectivo de simular e emular essas redes. As principais ferramentas estudadas são o TOSSIM, Freemote e o JProwler.

O TOSSIM é um simulador e emulador baseado em eventos que suporta TinyOS. A ferramenta oferece uma boa fiabilidade de simulação, no entanto, a escalabilidade tem algumas limitações. O modelo de energia apenas está incluído na versão PowerTOSSIM, já os modelos de rádio são o modelo simples e o modelo de perdas. De forma geral, a ferramenta tem alguma complexidade associada.

Freemote é uma ferramenta de simulação e emulação baseada em Java, na qual a sua principal vantagem consiste em juntar nós reais e simulados numa só rede, suportando alta escalabilidade. Um modelo de rádio fraco e a ausência de um modelo de

energia são duas desvantagens significativas, contudo tem uma simples usabilidade.

JProwler é um simulador de eventos em Java para alta escalabilidade, em que os seus nós suportam heterogeneidade na camada aplicacional e têm um camada MAC baseada em *Mica Motes*. O modelo de rádio é probabilístico e determinístico, o de energia não está incluído mas a fácil extensibilidade da ferramenta poderá incluí-lo, e o simulador adequa-se bem a análises de desempenho dos protocolos. A ferramenta escolhida é o JProwler, por satisfazer os requisitos essenciais para os objectivos propostos nesta dissertação.

Este estudo dos simuladores visa endereçar essencialmente as contribuições que dizem respeito à concepção da plataforma genérica, a escolha do simulador é feita de modo a cumprir os requisitos pretendidos, como é o caso de simular dispositivos *Micaz* ou *TelosB* em ambiente IEEE 802.15.4 e ainda proporcionar ambientes de larga escala.

3

Modelação e Arquitectura

3.1 Visão inicial do ambiente de simulação

A plataforma de simulação foi concebida tendo por base o ambiente de simulação JProwler que constitui assim o seu núcleo. Este ambiente foi estendido por vários componentes que pretendem oferecer mais funcionalidades ao utilizador, de modo a que possa ter ao seu dispor um maior número de ferramentas que lhe permitam analisar e avaliar protocolos de comunicação em determinadas topologias de rede com condições por ele parametrizadas. Portanto, de início interessa fazer uma abordagem geral ao núcleo da plataforma e como este foi estendido.

3.1.1 Descrição Geral

O JProwler [JPr] é um ambiente de simulação desenvolvido no ISIS (*Institute for Software Integrated System*) que está implementado em linguagem Java e permite simulações por eventos discretos. As suas funcionalidades são, sobretudo, permitir analisar e avaliar protocolos de comunicações (encaminhamento, distribuição de chaves, *etc.*) baseados em dispositivos do tipo *Mica Motes* (em especial *Mica2*) e *TelosB*, que estão de acordo com a *framework* TinyOS, realizando comunicações segundo a norma IEEE 802.15.4 e incorporando modelos de rádio para colisões. Uma das características importantes deste ambiente de simulação é a capacidade em gerir redes de larga escala, ou seja, na ordem dos milhares de nós.

No entanto, este simulador encontra-se numa fase primária, pois as suas funcionalidades são limitadas e a separação de camadas e componentes do nó sensor tem algumas deficiências, é o caso de protocolos de encaminhamento serem endereçados a um nível aplicacional, o que pode ser resolvido por benefício da fácil extensibilidade da ferramenta.

3.1.2 Motor de Simulação

O motor de simulação do JProwler trata de realizar a respectiva execução dos vários eventos que vão sendo depositados na fila de eventos discretos. Após a criação dos nós, quando estes iniciam as comunicações entre si, os eventos vão surgindo e cada um representa uma determinada acção a ser processada. Os eventos têm obrigatoriamente uma estampilha temporal que indica o valor temporal em que o evento deverá ser executado, sendo que esse valor está de acordo com o referencial de tempo interno ao próprio simulador.

As simulações são efectuadas por uma das três seguintes possibilidades de execução dos eventos:

1. execução com visualização em tempo real - Os eventos são processados com uma aproximação ao tempo real, isto por intermédio de um factor de conversão do tempo de simulação para o tempo real. Este modo revela-se lento quando se avaliam redes de larga escala;

2. execução com visualização - Os eventos são processados e os seus efeitos visualizados sem ter qualquer noção de tempo real. A execução é mais rápida e mais apropriada a redes de maiores dimensões;

3. execução sem visualização - Este modo não se preocupa com qualquer tipo de visualização da rede, pois o processamento dos eventos é efectuado imediatamente, o que permite avaliar redes de larga escala.

3.1.3 Modelos de rádio

Relativamente aos modelos de rádio, estes são importantes uma vez que simulam a transmissão, propagação e recepção de dados, assumindo a possibilidade de ocorrerem colisões, tal como acontece nas RSSF reais. Este simulador suporta dois modelos: o *Gaussian* e o *Rayleigh*. No momento da criação da rede, ambos os modelos definem as ligações físicas possíveis entre os nós, ou seja, estabelecem as vizinhanças físicas.

Quanto ao modelo de rádio *Gaussian*, este assume que os nós muito raramente mudam de posição e baseia-se numa distribuição probabilística. O modelo utiliza um

ruído gaussiano que é composto por uma parte dinâmica, recalculada em cada transmissão, e por uma parte estática que é definida logo que o nó seja criado. O modelo *Gaussian* é o considerado para a avaliação dos protocolos, uma vez que se assume que as redes se encontram sem mobilidade.

O modelo de rádio *Rayleigh* é também ele baseado numa distribuição probabilística, no entanto adequa-se a redes onde a mobilidade está presente. A intensidade do sinal é modulada tendo por base um período, designado tempo de coerência, que utiliza uma distribuição exponencial. O ruído é constituído por uma parte dinâmica e uma estática, sendo que a dinâmica é recalculada por cada vez que ocorre um período igual ao tempo de coerência. Este modelo, embora disponível, não será utilizado para efeitos desta dissertação.

3.1.4 Versão do simulador utilizado

O desenvolvimento da plataforma tem por base o ambiente de simulação JProwler, sendo ele o verdadeiro núcleo da plataforma. Contudo, o facto de existir uma versão posterior ao ambiente JProwler, designada WiSeNet Simulator [dS11], torna-se muito útil e interessante. Com isto foram adicionados os componentes necessários nesta dissertação tendo por uso a WiSeNet, o que permite usufruir de ferramentas já desenvolvidas que se revelam bastante funcionais na fase de testes e avaliação.

A WiSeNet herda do JProwler as características atrás descritas, porém, contrariamente a WiSeNet está bastante desenvolvida a nível visual, oferecendo um conjunto de ferramentas que dão a possibilidade ao utilizador de facilmente especificar uma topologia pretendida, controlar os nós da rede, obter informações acerca de cada nó ou de toda a rede (por exemplo analisar as vizinhanças), bem como observar o comportamento dinâmico do protocolo em execução. Estão ainda presentes componentes para controlar o estado da simulação que indicam, por exemplo, o número corrente de eventos ou o tempo decorrido da respectiva simulação.

Nesta versão utilizada, a representação de cada nó da rede está de acordo com a estrutura de camadas da pilha, o que facilita a implementação de protocolos para a respectiva camada, em particular a implementação de protocolos de encaminhamento, uma vez que são desenvolvidos ao nível da camada de encaminhamento. Outra característica importante é a existência de um modelo energético de base que regista o consumo energético, o qual não está presente no JProwler e será útil para o módulo de energia a ser implementado. Na figura 3.1 observa-se o aspecto visual do simulador WiSeNet Simulator.

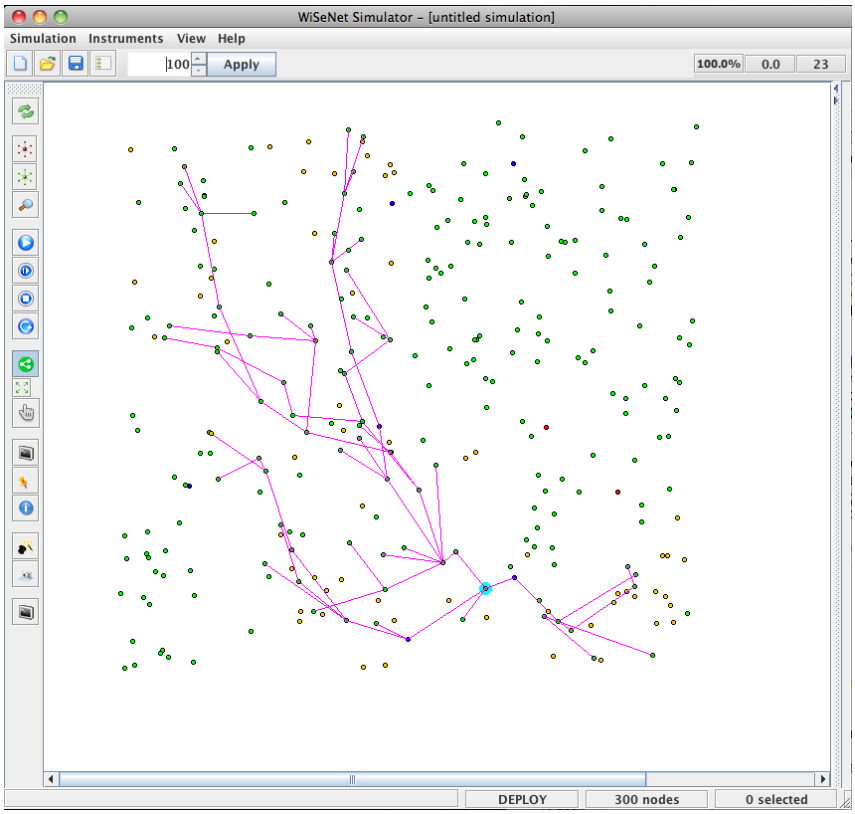


Figura 3.1: WiSeNet Simulator.

3.2 Arquitectura da plataforma

Tendo em vista o tipo de testes e avaliações a serem realizados, o WiSeNet Simulator será integrado com os módulos de cobertura, latência, fiabilidade e energia, para que seja possível efectuar medições aos protocolos de encaminhamento. A plataforma será ainda constituída por um módulo de injecção de ataques para que os ataques possam ser injectados na rede e posteriormente sejam efectuadas as medições pretendidas. Por fim, acrescenta-se também o módulo de visualização e uma camada de segurança baseada na normalização TinySec [KSW04] para comunicações seguras.

3.2.1 Módulo de Cobertura

O módulo de cobertura (MC) tem como objectivo obter um valor percentual da cobertura (taxa de cobertura) em função de uma topologia com um determinado número de nós, dimensão da área (em metros) onde são implantados os sensores, e por vezes também é necessária uma percentagem de emissores que enviem mensagens para certos destinos. Este módulo permite medir três tipos de cobertura, são eles a cobertura rádio ou física, cobertura parcial e cobertura total.

1. **Cobertura rádio**

   A cobertura rádio ou física consiste em apresentar uma taxa de cobertura ao nível das ligações físicas que foram estabelecidas entre nós. De facto, por cada nó da rede verifica-se quantos nós são seus vizinhos fisicamente, se tal valor ultrapassar o limiar mínimo considerado (valor um por predefinição), então o nó irá ser admitido como fisicamente coberto. Avaliando todos os nós obtém-se, assim, a cobertura de rádio da rede. Calcula-se ainda o número médio, mínimo e máximo de nós vizinhos em cada nó. Este tipo de cobertura não necessita que emissores enviem mensagens, uma vez que é obtida logo após a criação da rede, e também não se encontra efectivamente ao nível do encaminhamento, no entanto é relevante no sentido em que influencia as coberturas parcial e total. A taxa de cobertura rádio é calculada da seguinte forma:

$$TaxaCobertura_{radio} = \left(\frac{N^{o}NosCobertos_{radio}}{N^{o}TotalNos} \right) \times 100 \tag{3.1}$$

2. **Cobertura parcial**

   A cobertura parcial da rede é calculada verificando se os nós da rede conseguem encaminhar dados para, pelo menos, um nó da rede que seja seu vizinho, sendo diferente da cobertura anterior, pois aqui a verificação trata-se ao nível do encaminhamento. Esta cobertura tem interesse porque, por exemplo, o simples facto de ocorrerem intrusões na vizinhança do nó emissor poderia logo impedir que a mensagem fosse recebida por um dos seus vizinhos.

   Este tipo de cobertura requer que seja indicada a percentagem de nós da rede que desempenham o papel de emissores e que enviam um certo número de mensagens de forma a garantir a possibilidade de atingir algum nó vizinho. A taxa de cobertura parcial é calculada da seguinte forma:

$$TaxaCobertura_{parcial} = \left(\frac{N^{o}EmissoresCobertos_{parcial}}{N^{o}TotalEmissores} \right) \times 100 \tag{3.2}$$

3. **Cobertura total**

   A cobertura total consiste em verificar se as mensagens enviadas pelos emissores atingem um determinado nó de destino, designado por *sink node*, se tal verificar-se então esse emissor está totalmente coberto. Mais uma vez, é necessário definir uma percentagem de nós emissores de mensagens, e estas devem ser enviadas repetidamente de forma a garantir que conseguem ou não alcançar o *sink node*.

   Este tipo de cobertura é mais interessante que a anterior, pois num ambiente real

os dados são normalmente encaminhados em direcção ao *sink node* e, portanto, esta taxa de cobertura reflecte melhor a qualidade de cobertura do protocolo. A taxa de cobertura total é calculada da seguinte forma:

$$TaxaCobertura_{total} = \left(\frac{N^{o}EmissoresCobertos_{total}}{N^{o}TotalEmissores} \right) \times 100 \tag{3.3}$$

A arquitectura do módulo de cobertura, observada na figura 3.2, é composta por um controlo de cobertura, o qual tem a função de controlar a medição, bem como receber a parametrização indicada pelo utilizador, iniciar a simulação, notificar acerca do seu final, solicitando a apresentação dos resultados finais. Durante a medição o controlo comunica com o componente de cobertura de rádio, parcial ou total, pois estes funcionam como um intermediário entre o controlo e o receptor de resultados da cobertura. Este receptor de resultados é importante no sentido de que armazena temporariamente os resultados a serem apresentados no final da simulação.

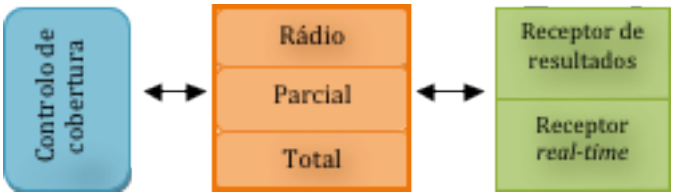


Figura 3.2: Arquitectura do módulo de cobertura.

Por vezes é utilizado o receptor de resultados em tempo real, este tem como objectivo analisar a evolução da correspondente taxa de cobertura através de um gráfico em tempo real que representa a actual taxa de cobertura em função do número de emissores testados até ao momento, inclui-se ainda a média e o desvio padrão. Este gráfico apenas se aplica à cobertura total e parcial, pois a taxa de cobertura rádio é obtida logo após a formação da rede.

3.2.2 Módulo de Latência

O módulo de latência (ML) é concebido para que os protocolos de encaminhamento possam ser medidos e avaliados em termos de latência, quando se está perante uma rede com determinadas características parametrizadas pelo utilizador.

De facto, pretende-se que uma certa percentagem de nós da rede desempenhe o papel de emissor e envie mensagens que têm por destino o *sink node*. Por conseguinte, determina-se a latência associada a cada mensagem desde que é enviada até ao momento em que é recebida no destino. Como resultado final é obtida a latência mínima,

média e máxima (em milissegundos), por contabilização de todas as mensagens da rede que foram enviadas e entregues no *sink node*. Repare-se que certos emissores, porventura, não estão totalmente cobertos e assim as suas mensagens nunca chegarão ao *sink node*, isto leva à necessidade de procurar emissores que estejam totalmente cobertos antes de ser efectuada a medição de latência.

Um factor relevante e influente nos resultados de latência é o nível de congestionamento em que a rede se encontra, ou seja, se a rede contém um elevado congestionamento de mensagens possivelmente ocorrem efeitos negativos nos resultados. Assim, permite-se parametrizar o nível de congestionamento fixando um número de mensagens enviadas por cada emissor, bem como o período decorrido entre o envio de cada mensagem, sendo que os emissores estão enviando as suas mensagens de forma concorrente, susceptível a possíveis colisões.

O cálculo da latência média final baseia-se na latência média obtida por cada emissor, caso contrário, uma vez que cada emissor envia um certo número de mensagens para o *sink node* e com o surgimento de possíveis colisões, alguns emissores têm mais mensagens entregues que outros, e assim a latência média final tenderia incorrectamente para os emissores que mais mensagens conseguiram entregar. Saliente-se ainda que a latência é medida utilizando o referencial de tempo interno ao simulador, pois é este que representa a verdadeira dimensão temporal da simulação.

Outro aspecto a ter em consideração é o facto de a latência estar relacionada com o número de saltos entre nós (número de *hops*), logo o número de saltos mínimo, médio e máximo deve ser apresentado nos resultados obtidos.

Figura 3.3: Arquitectura do módulo de latência.

A arquitectura do módulo de latência, presente na figura 3.3, é constituída pelo componente de controlo, que trata de controlar a medição, bem como receber a parametrização necessária. Contém o componente de latência que se responsabiliza por fazer a ligação entre o controlo e o receptor de resultados, gerindo alguma informação. O componente de recepção de resultados em tempo real é opcionalmente utilizado e, nesse caso, durante a medição os resultados vão sendo apresentados sob a forma de gráfico. Neste caso são apresentados dois gráficos, o primeiro diz respeito à pesquisa

de emissores cobertos na rede, mostrando a taxa de sucesso em função dos emissores verificados. O segundo é relativo à latência, observando-se como evolui a latência (mínima, média e máxima) em função do total de mensagens que já alcançaram o *sink node*.

3.2.3 Módulo de Fiabilidade

O módulo de fiabilidade (MF) permitirá analisar os protocolos de encaminhamento relativamente à sua capacidade de entrega de mensagens. Este módulo tem o funcionamento semelhante ao módulo de latência referido em 3.2.2, como é o caso da percentagem de nós emissores que enviam mensagens para o *sink node*, ou as parametrizações efectuadas que permitem também elas definir qual o nível de congestionamento da rede. Contudo, aqui interessa-nos obter a taxa de fiabilidade (mensagens entregues) por análise do número total de mensagens recebidas no *sink node* e o número total de mensagens enviadas pelos emissores.

$$TaxaFiabilidade = \left(\frac{N^{o}MensagensRecebidas}{N^{o}MensagensEnviadas} \right) \times 100 \tag{3.4}$$

Mais uma vez, ocorre a necessidade de encontrar emissores que sejam totalmente cobertos antes de iniciar a medição da fiabilidade, pois só fará sentido analisar o número de mensagens recebidas quando tais mensagens conseguem atingir o nó de destino. O número de saltos entre nós também se revela importante neste tipo de medição, sendo apresentado no resultado final o número de saltos mínimo, médio e máximo.

Figura 3.4: Arquitectura do módulo de fiabilidade.

A arquitectura do módulo de fiabilidade, observável na figura 3.4, integra o controlo de fiabilidade, responsável por controlar a medição e pelas parametrizações, à semelhança da arquitectura dos módulos anteriores. Contém o componente de fiabilidade que funciona como intermediário para o receptor de resultados. O receptor de resultados em tempo real permite a obtenção do gráfico de pesquisa de emissores totalmente cobertos e o gráfico que apresenta a evolução da taxa de fiabilidade em função do número de mensagens que são recebidas no *sink node*, inclui-se ainda a média e o desvio padrão dessa taxa.

3.2.4 Módulo de Energia

O módulo de energia (ME) é destinado a analisar o consumo energético em protocolos de encaminhamento sob duas vertentes: o consumo energético durante a configuração do protocolo e após a configuração do protocolo, ou seja, durante a fase do encaminhamento de dados.

1. **Durante a configuração do protocolo**

   A fase de configuração de um protocolo de encaminhamento é um processo que envolve algum consumo energético, por vezes, significativo quando se lida com redes de larga escala. Assim sendo, torna-se importante contabilizar a energia necessária por parte dos nós durante a configuração do protocolo, a qual inclui a fase de descoberta de rotas e a selecção de rotas. O resultado final apresenta o consumo total de energia necessário para criar as tabelas de encaminhamento.

2. **Após a configuração do protocolo (encaminhamento)**

   Seguidamente à configuração do protocolo, os nós passam então a encaminhar mensagens segundo as rotas que foram estabelecidas, o que implica consumo energético por parte de vários nós para encaminhar cada mensagem até um determinado nó de destino. Assim, o objectivo passa por escolher uma percentagem de nós da rede que desempenham o papel de emissores e que enviam mensagens para um *sink node*, de tal forma que por cada mensagem é analisado o consumo energético despendido pela rede para que esta chegue ao *sink node*. Após enviadas todas as mensagens, o resultado final apresenta o consumo mínimo, máximo e médio (em Joules). O cálculo de consumo médio final baseia-se no consumo médio por cada emissor, caso contrário, como cada emissor envia um certo número de mensagens para o *sink node* e com a possível ocorrência de colisões, alguns emissores têm mais mensagens entregues que outros e, por conseguinte, o consumo energético médio final tenderia incorrectamente para os emissores que mais mensagens conseguiram entregar.

   Mais uma vez, é essencial que inicialmente sejam encontrados emissores totalmente cobertos, à semelhança dos módulos em 3.2.2 e 3.2.3. A este nível o congestionamento da rede também é relevante, bem como o número (mínimo, médio e máximo) de saltos que a mensagem acarreta.

Relativamente à arquitectura do módulo de energia, ilustrada na figura 3.5, esta é composta pelo controlo de energia, pelo componente que trata o tipo de medição energética (durante a configuração - *setting* ou após a configuração - *routing*) e que interliga

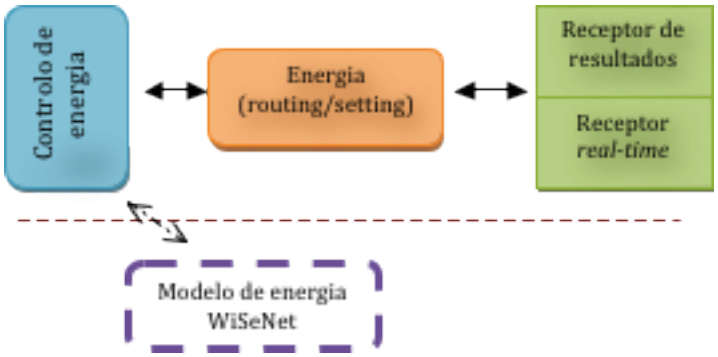


Figura 3.5: Arquitectura do módulo de energia

Tabela 3.1: Valores energéticos utilizados no modelo de energia.

Acção	Custo energético
Transmissão	59,2 μJ/byte
Recepção	28,6 μJ/byte
Cifra/Decifra	1,788 μJ/byte
Assinatura	5,9 μJ/byte
Verificação da assinatura	5,9 μJ/byte
Função de síntese	5,9 μJ/byte
Verificação da síntese	5,9 μJ/byte
Transição para CPU	0,001 μJ
Transição para transceptor	0,002 μJ
Estado *idle*	5,9 μJ/s
Estado adormecido	7,5 μJ/s
Estado activo (processamento)	13800 μJ/s

o controlo ao receptor de resultados. O receptor de resultados em tempo real apenas é utilizado quando se está perante uma medição do tipo "após a configuração do protocolo" e disponibiliza os dados para a apresentação de gráficos em tempo real, como os mencionados no módulo de latência em 3.2.2, embora neste caso a nível energético.

Note-se que este módulo tira partido de um modelo de energia previamente concebido e integrado na WiSeNet. Este modelo energético está preparado para simular a bateria de um sensor, permitindo registar o seu consumo energético que pode ocorrer por diversas acções: comunicações, processamento, operações criptográficas e transições nos estados internos do dispositivo. A tabela 3.1 apresenta os valores de referência [WGE+05][GSSK05] utilizados para as várias acções que provocam consumo energético. Repare-se que os valores reais para um sensor em particular podem ter uma ligeira diferença em relação a estes valores de referência, porém eles reflectem o consumo tipicamente gasto pelos sensores considerados.

3.2.5 Módulo de Injecção de ataques

O módulo de injecção de ataques (MIA) tem como objectivo induzir comportamentos incorrectos na camada de encaminhamento para determinados nós da rede. Os nós pretendidos para injecção de comportamentos incorrectos devem ser seleccionados antes do início da simulação, existindo várias formas de selecção: seleccionar graficamente pela interface visual, seleccionar todos automaticamente, seleccionar uma percentagem de nós de forma aleatória e especificar quais os nós pretendidos através dos seus identificadores. Além da selecção de nós, é necessário escolher que tipo de comportamento malicioso é pretendido, o qual substituirá o comportamento legítimo do nó de acordo com o respectivo protocolo.

Relativamente à apresentação de resultados, se trata-se de um (i) ataque à selecção de rotas exibem-se duas secções: a afluência de mensagens em nós legítimos e atacantes, e o encaminhamento redundante. Na primeira, uma vez que interessa avaliar a capacidade do atacante influenciar a escolha da rota, compara-se a afluência de mensagens em nós legítimos da rede em relação à dos nós atacantes, isto através do número médio, mínimo, máximo e total de mensagens em nós atacantes e em nós legítimos. A segunda, por sua vez, tem como objectivo analisar a capacidade que o protocolo tem em realizar encaminhamento redundante que contorne nós atacantes, indicando-se então o número de mensagens que circulam apenas em caminhos infectados, apenas em caminhos legítimos, simultaneamente por infectados e legítimos, *etc*.

Se o ataque corresponde a um ataque (ii) ao controlo de encaminhamento e que contempla descarte de mensagens (*blackhole* e *selective-forwarding*) apresentam-se unicamente os resultados do encaminhamento redundante, indicando-se o número de mensagens recebidas, descartadas, simultaneamente recebidas e descartadas, *etc*.

3.2.6 Módulo de visualização de resultados

O módulo de visualização de resultados (MVR) é simples no sentido em que apenas tem a finalidade de apresentar o gráfico que expressa o resultado obtido. Os gráficos são construídos inicialmente e actualizados aquando a recepção de valores fornecidos pelos módulos anteriormente descritos.

3.2.7 Camada de segurança

Para fornecer primitivas de comunicação segura entre os nós foi criada na plataforma uma camada de segurança que segue a normalização TinySec [KSW04]. O TinySec é uma arquitectura desenhada para oferecer segurança tendo em conta as limitações dos recursos.

Esta camada possui duas opções de segurança distintas: pacotes com cifra e autenticação ou pacotes com apenas autenticação, como se observa nas figuras 3.6 e 3.7. O algoritmo criptográfico predefinido é o *Skipjack* com o modo CBC, tratando-se de criptografia simétrica, e as suas chaves podem ser distribuídas por várias estratégias, assumindo-se neste caso que são pré-carregadas antes da criação da rede. Para garantir a integridade das mensagens recorre-se à utilização de operações MAC (*Message Authentication Code*).

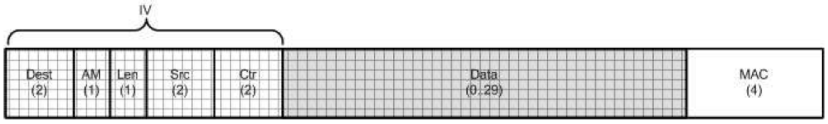


Figura 3.6: Pacote TinySec com cifra e autenticação.

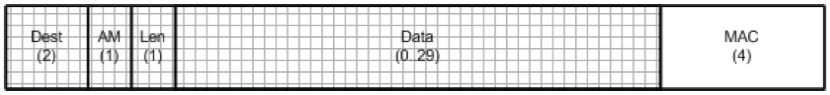


Figura 3.7: Pacote TinySec com apenas autenticação.

A abordagem utilizada consiste em sempre que seja pretendida uma mensagem segura o nível de encaminhamento entrega-a ao nível MAC e este, por sua vez, verificando que se trata de uma mensagem que requer segurança entrega-a à camada TinySec que cria um pacote TinySec (seguro), de acordo com a especificação documentada, que permite garantir as necessidades de segurança básicas.

Na corrente implementação utiliza-se a camada TinySec, no entanto, alguns protocolos requerem a utilização de outros métodos criptográficos não contemplados na sua especificação de base. Portanto, procedeu-se à implementação de uma extensão da camada TinySec que dispõe de mecanismos que não eram antes oferecidos e que se revelam necessários para os protocolos a serem implementados.

3.2.8 Plataforma integral

A arquitectura da plataforma pode ser visualizada na figura 3.8, sendo constituída pela versão WiSeNet Simulator que contém o núcleo de base JProwler, pelos módulos descritos anteriormente e por uma camada de segurança TinySec. Os sistemas de encaminhamento não estão incorporados na plataforma, contudo foram implementados para que possam actuar sobre esta e serem alvo de experimentação. Apenas a versão

WiSeNet e o seu núcleo de base foram reutilizados, todos os restantes componentes/-módulos da arquitectura foram desenvolvidos para efeitos desta dissertação.

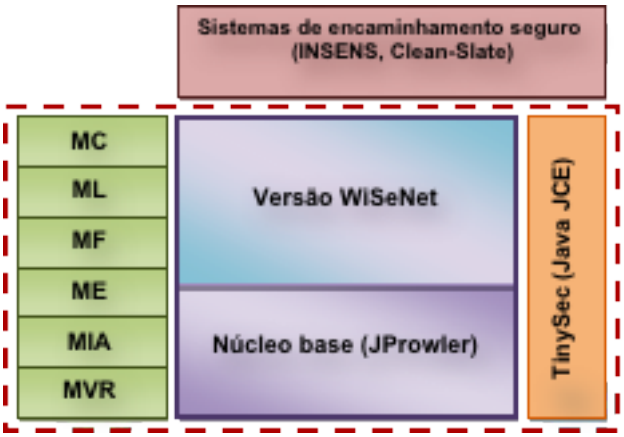


Figura 3.8: Arquitectura da plataforma.

Importa referir que todos os módulos de medição da plataforma estão ainda preparados para a apresentação de resultados gráficos após várias simulações realizadas. Tendo como exemplo o caso da cobertura, é possível gerar um gráfico que apresenta a evolução da taxa de cobertura em função do número de nós da rede para uma determinada percentagem de emissores, o que requer o armazenamento dos resultados de várias simulações. É também possível gerar o gráfico que apresenta a evolução da cobertura em função da percentagem de emissores para um certo número de nós.

Os resultados obtidos pelos módulos de medição, quando considerados interessantes pelo utilizador, são armazenados em disco para posteriormente serem alvo de análise. Os módulos de cobertura, latência, fiabilidade, energia e injecção de ataques oferecem interfaces gráficas que facilitam o utilizador a efectuar as operações pretendidas.

3.3 Modelação de protocolos de encaminhamento

Os protocolos de encaminhamento quando implementados devem seguir uma certa especificação exigida pela própria plataforma, sendo aqui abordada a metodologia geral necessária para implementar um protocolo de encaminhamento.

- O primeiro requisito passa por configurar a fábrica de nós *node factory*, que irá criar nós de acordo com a especificação desejada, tendo em conta o protocolo de encaminhamento pretendido. Isto pressupõe que sejam especificadas as seguintes características:

1. camada de aplicação;

2. camada de encaminhamento;
3. camada MAC;
4. atributos do próprio nó;
5. possíveis comportamentos incorrectos.

A especificação da camada de aplicação é tida em conta porque permitirá definir o comportamento aplicativo, o qual fará solicitações à camada de encaminhamento de modo a que esta opere. A camada de encaminhamento é a mais relevante na implementação de um protocolo de encaminhamento, enquanto a camada MAC normalmente não sofre alterações, excepto se o utilizador pretende simular o comportamento MAC de outro tipo de sensores.

• No passo seguinte da metodologia sugere-se a definição das mensagens específicas do protocolo. Essas mensagens deverão ser uma extensão da mensagem de base que o simulador utiliza, a qual contém informação pertinente. Na camada de encaminhamento existem as primitivas de transmissão e de recepção de mensagens, sendo que na recepção de mensagens verifica-se qual o tipo de mensagem recebida e, por conseguinte, é tratado o respectivo código associado ao tipo da mensagem. Qualquer mensagem que se pretende torná-la numa mensagem segura deve ter nos seus atributos a indicação de que se trata de uma mensagem segura, o que levará a camada TinySec a criar um pacote TinySec que lhe acrescenta as garantias de segurança básicas.

• Após definidas as mensagens é essencial que seja programado o código a executar logo após a recepção da respectiva mensagem. Por vezes, ocorre a necessidade de executar acções de carácter não imediato, neste caso recorre-se ao uso de eventos que permitem o agendamento de um evento que será executado mais tarde. De facto, se é pretendido que o protocolo seja alvo de medições, deverá ter-se o cuidado de utilizar as primitivas dos módulos de medição apropriadamente.

3.4 Visão geral

A plataforma concebida tem por base a versão de simulação WiSeNet Simulator, que utiliza como núcleo base o JProwler, o que permite assim oferecer ferramentas bastante úteis quando aliadas aos módulos adicionados. O modelo de rádio *Gaussian* é o utilizado na plataforma, sendo indicado para redes pouco móveis.

Os módulos concebidos para a plataforma são os seguintes: módulo de cobertura (MC), módulo de latência (ML), módulo de fiabilidade (MF), módulo de energia (ME), módulo de injecção de ataques (MIA) e módulo de visualização de resultados (MVR).

O módulo de cobertura contém três tipos de cobertura: (i) a cobertura rádio indica a taxa de cobertura ao nível físico entre os nós; (ii) a cobertura parcial é determinada verificando se o nó consegue enviar com sucesso mensagens para algum dos seus vizinhos; (iii) a cobertura total verifica se as mensagens emitidas conseguem atingir o *sink node*. O módulo de latência calcula a latência mínima, média e máxima, considerando as mensagens recebidas no *sink node*. Já o módulo de fiabilidade preocupa-se em analisar quantas mensagens são recebidas no *sink node*, em relação ao total de mensagens emitidas. O módulo de energia avalia a energia utilizada durante a fase de configuração do protocolo (*setting*) e durante o encaminhamento de dados (*routing*), neste último indicando o consumo mínimo, médio e máximo provocado pelas mensagens recebidas no *sink node*.

Tabela 3.2: Resumo das características dos módulos de medição.

Módulo	**Resultado**	**Emissores**	**Destino**	**Cobertura**	***Stress***	**Saltos**	***Real-time***
Cobertura Rádio	Taxa (%)	-	-	-	-	-	-
Cobertura Parcial	Taxa (%)	+	vizinho	-	-	-	+
Cobertura Total	Taxa (%)	+	*sink node*	-	-	-	+
Latência	Min,Média, Max (ms)	+	*sink node*	+	+	+	+
Fiabilidade	Taxa (%)	+	*sink node*	+	+	+	+
Energia (*Routing*)	Min,Média, Max (Joules)	+	*sink node*	+	+	+	+
Energia (*Setting*)	Total (Joules)	-	-	-	-	-	-

A tabela 3.2 apresenta uma visão compacta das características dos módulos de medição. Por cada módulo (tipo de medição) apresenta-se o tipo de resultado obtido, se são necessários emissores, se é utilizado um destino para as mensagens e qual é, se é necessário realizar uma pesquisa de emissores cobertos, se é parametrizável o congestionamento da rede, se é considerado o número de saltos das mensagens, e se é possível visualizar resultados em tempo real. A ordem dos campos descritos corresponde à ordem dos campos presentes na tabela.

O módulo de injecção de ataques permite induzir comportamentos incorrectos na camada de encaminhamento dos nós, existindo várias formas de selecção dos nós atacantes. O módulo de visualização de resultados apenas tem a finalidade de apresentar os resultados graficamente. Para dispor de primitivas de comunicação segura adicionou-se a camada de segurança TinySec, tendo sido ainda estendida devido a mecanismos requeridos pelos protocolos e que não estavam antes contemplados.

Relativamente à modelação de protocolos de encaminhamento, sugere-se que inicialmente se defina a fábrica de nós, de seguida as mensagens utilizadas pelo protocolo e por fim o respectivo código a executar na recepção da mensagem.

4

Implementação da plataforma de simulação

Neste capítulo pretende-se fazer uma abordagem aos diversos aspectos que dizem respeito à implementação da plataforma e que se consideram relevantes, tendo em conta a especificação já referida em 3.2.

A plataforma foi implementada segundo o modelo de objectos Java, tal como acontecia com o seu núcleo de base JProwler. Os módulos pretendidos e anteriormente especificados foram adaptados à plataforma WiSeNet Simulator [dS11] (ambiente já desenvolvido sobre o JProwler), sendo utilizada uma versão desta que apresentava as características e as funcionalidades necessárias para parametrizar e gerir a rede de acordo com os objectivos de cada módulo e permitindo ainda realizar simulações com as condições desejadas.

4.1 Interfaces gráficas

Uma grande parte dos módulos desenvolvidos requer a utilização de componentes gráficos que permitam uma fácil interacção entre o utilizador e a plataforma, de modo a que este proceda às parametrizações que pretende. Para tal, utilizou-se o modelo de objectos gráficos da biblioteca Java Swing, o qual já era utilizado no ambiente gráfico de base concebido na WiSeNet. Adicionaram-se componentes gráficos para parametrizar a injecção de ataques, realizar as medições, bem como apresentar os resultados.

Mais concretamente, na plataforma WiSeNet existe uma secção *instruments*, como se pode observar na figura 3.1 do capítulo 3, onde implementaram-se duas opções: *Measurement* e *Injection attacks*. A opção *Measurement* abre uma pequena interface gráfica que contém um menu que permite ao utilizador entrar no tipo de medição pretendida (cobertura, latência, fiabilidade e energia) e parametrizá-la, para de seguida iniciar uma simulação e a correspondente medição. A opção *Injection attacks* permite abrir uma outra interface que oferece ao utilizador a possibilidade de configurar o tipo de ataque. Na figura 4.1 podem observar-se as componentes gráficas de injecção de ataques e de medições, onde neste caso está seleccionada a medição de latência.

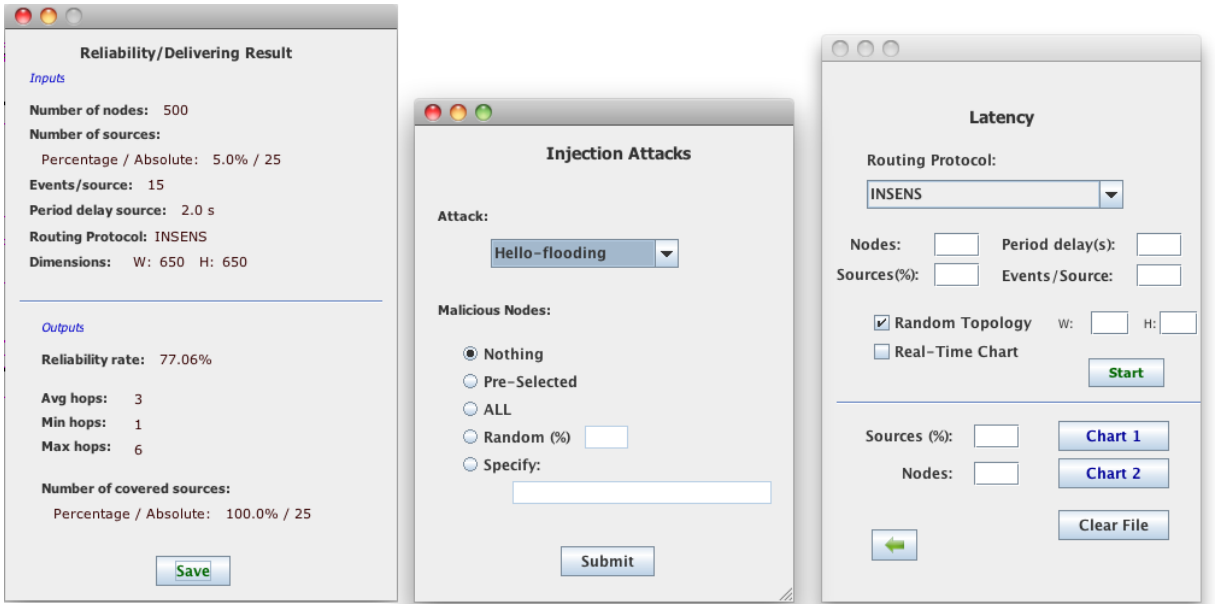


Figura 4.1: Componentes gráficos para ataques, medições e resultados.

Na componente de medição de latência podemos visualizar que existem botões para a apresentação de gráficos (*Chart 1* e *Chart 2*), estes gráficos são criados através do módulo de visualização de resultados que utiliza a biblioteca Java Free Chart. Além destes componentes gráficos, foram ainda implementados outros que no final de cada medição apresentam os resultados obtidos. Na apresentação de resultados indicam-se várias informações úteis ao utilizador acerca da respectiva medição, um exemplo do resultado de fiabilidade é apresentado na figura 4.1. Outros detalhes foram também implementados e adaptados à WiSeNet, nomeadamente a possibilidade de seleccionar graficamente os atacantes ou os emissores de mensagens.

4.2 Implementação de módulos e camadas

Módulos de medição

Cada módulo de medição inclui uma interface na sua implementação para disponibilizar as primitivas necessárias a serem utilizadas, principalmente no nível de encaminhamento. Exemplificando, na medição de cobertura total quando uma mensagem atinge a estação-base, o seu nível de encaminhamento utiliza a primitiva correspondente para informar o controlo de cobertura acerca da sua chegada.

Uma vez que a apresentação de resultados ocorre posteriormente ao final da medição, importa saber quando esta se dá por terminada. O mecanismo implementado que efectua esta verificação consiste em utilizar um evento que analisa periodicamente se existem outros eventos pendentes para serem executados. Esses eventos pendentes dizem respeito aos originados pelo protocolo, e não existindo mais nenhum pendente significa que a medição terminou. Este mecanismo utiliza-se não só para determinar quando a medição termina por completo mas também noutras situações, é o caso da medição de cobertura, onde inicialmente se espera que a execução do protocolo termine para de seguida iniciar o envio de mensagens que calculam a taxa de cobertura.

Na medição da cobertura total e parcial, após terminar a execução do protocolo (verificada com o mecanismo referido), as mensagens para cálculo da cobertura são emitidas na rede, uma de cada vez, para evitar colisões. Assim, existe também um evento periódico que verifica e lança a mensagem quando mais nenhuma se encontra na rede.

Relativamente à latência, após ser detectado que o protocolo terminou a execução inicia-se a pesquisa de emissores totalmente cobertos, onde mais uma vez se utiliza um evento responsável por lançar e garantir que não ocorrem colisões. Terminada essa fase, dá-se início ao lançamento de mensagens por parte dos emissores de modo a obter os valores de latência, como tal existe um evento por cada emissor que é responsável por enviar periodicamente as mensagens do seu emissor. Novamente, volta a ser utilizado o mecanismo que detecta quando a medição termina, sendo de seguida apresentado o resultado. Este procedimento aplica-se igualmente ao módulo de fiabilidade e de energia (na medição à fase de encaminhamento).

Existe um ficheiro de configuração dos módulos de medição que foi implementado no sentido de facilitar o utilizador a fazer algumas parametrizações, por exemplo os valores temporais associados a este tipo de eventos periódicos acabados de referir ou, relativamente à cobertura, alterar o valor predefinido para o número de mensagens enviadas por emissor que verificam se é um emissor coberto.

A contagem do número de saltos de cada mensagem é também realizada pelos módulos de medição (latência e energia), essa contagem é conseguida através de um simples algoritmo que utiliza a indicação do nó emissor da mensagem e do nó receptor. Assim, por cada recepção de uma mensagem a primitiva de actualização do número de saltos é chamada e esse número devidamente actualizado. Quando a mensagem é recebida pela estação-base/*sink node* obtém-se o verdadeiro número de saltos efectuados durante todo o caminho.

Módulo de injecção de ataques

A implementação do módulo de injecção de ataques, após seleccionados os nós atacantes, trata de modificar a camada de encaminhamento de cada nó atacante para uma nova camada que define o comportamento do ataque que é pretendido simular, diferenciando visualmente os nós atacantes dos restantes nós.

O módulo contém também uma interface que disponibiliza várias primitivas e que são utilizadas na camada de encaminhamento com o objectivo de oferecer informação relativa ao ataque, sendo apresentada no final da medição. Para que esses resultados sejam apresentados, os módulos de medição notificam este módulo quando terminada a medição. No caso de estarmos perante o ataque *wormhole*, o módulo gere ainda os nós que já têm um *wormhole* formado e aqueles que ainda estão disponíveis para formarem um novo.

Este módulo contém um ficheiro de configuração para que o utilizador defina quais os ataques que pretende ver disponíveis para seleccionar e permite ainda parametrizar alguns detalhes dos próprios ataques, por exemplo o alcance dos nós que realizam ataques *hello-flooding*.

Camada de Segurança TinySec

A camada de segurança TinySec é implementada através do Java Cryptography Extension (JCE). As mensagens têm um parâmetro que indica se é uma mensagem segura ou não.

Quando é chamada a primitiva para o envio de uma mensagem (ao nível MAC) o parâmetro é verificado, se trata-se de uma mensagem para incluir segurança então a camada de segurança cria um pacote TinySec que envolve a mensagem, caso contrário é utilizada a mensagem original, sendo enviada de seguida. Quando a mensagem é recebida pelo receptor na sua camada MAC examina-se se é um pacote TinySec, caso seja então fazem-se as respectivas verificações de segurança, e se o procedimento for

bem-sucedido então a mensagem é entregue na camada de encaminhamento (já sem o pacote TinySec). Deste modo, os pacotes TinySec apenas ocorrem ao nível MAC.

4.3 Detalhes de implementação da WiSeNet

Uma vez que foi utilizada a versão WiSeNet Simulator, importa assim referir algumas alterações de implementação mais notáveis em relação ao simulador JProwler.

Uma das características que mais se destaca nesta implementação da WiSeNet é a separação conceptual que é feita tanto ao nível das camadas de um nó como dos seus componentes. Isto é, cada nó contém separadamente as camadas MAC, encaminhamento e aplicação, e contém ainda separadamente os componentes CPU e o transceptor. Esta separação de conceitos facilita o programador a elaborar uma programação dos nós sensores mais específica a um determinado nível conceptual do sensor.

Para que os nós sejam facilmente criados está implementada uma fábrica de nós que os cria de acordo com as respectivas parametrizações, essas parametrizações definem a aplicação, a camada de encaminhamento, a camada MAC e a definição do próprio nó.

Outra característica presente é a utilização de uma mensagem de base que contém os campos primários de qualquer mensagem. Os principais campos desta mensagem são: o identificador do emissor, o identificador do destinatário, o identificador do último nó que emitiu a mensagem, o *payload*, a indicação de se é ou não uma mensagem segura. Outros campos são também incluídos, embora apenas para ter um maior controlo da mensagem, como é o caso da inclusão do seu número de identificação. Mensagens mais complexas, que requerem mais campos, devem estender esta mensagem de base.

Relativamente ao núcleo de gestão dos eventos do simulador, este mantém-se praticamente semelhante ao do JProwler, pois na verdade a versão WiSeNet utilizada incorpora essencialmente o núcleo JProwler, fazendo uma melhor separação de conceitos e aliando uma interface gráfica bastante mais desenvolvida.

Modelo de energia

Visto que o modelo de energia não era incluído no JProwler, interessa abordar alguns aspectos da sua implementação. Este modelo de energia preconiza, de certo modo, os principais estados de um sensor: *idle*, *sleep* e *processing*. O estado *idle* é simulado através de um consumo periódico (por cada segundo) de cada nó. Já o estado *processing* ocorre quando uma mensagem é recebida e consequentemente processada,

contudo isto ocorre apenas ao nível do encaminhamento. Relativamente ao *sleep*, este não é contemplado nesta implementação, no entanto se pensar-se em desenvolver uma camada MAC que utilize este estado, o modelo de energia encontra-se preparado para oferecer o respectivo valor. Todas as outras operações que dizem respeito à criptografia ou à transmissão/recepção de dados são devidamente contabilizadas para consumo energético, em que o consumo dependerá do tamanho da mensagem. Os valores de energia utilizados encontram-se já especificados na tabela 3.1 do capítulo 3.

4.4 Alterações à configuração-base

A versão WiSeNet utilizada, revelando-se muito útil para os objectivos pretendidos, necessita ainda assim de alguns ajustes na sua configuração inicial.

Um primeiro ajuste a mencionar está relacionado com o tempo de transmissão utilizado no envio das mensagens, o qual é configurado ao nível MAC. De facto, a WiSeNet mantém o mesmo valor de transmissão que está presente no JProwler, o valor 960 que corresponde a 24 milissegundos. No entanto, os 24 milissegundos são o tempo médio de transmissão para sensores que têm uma taxa de transmissão de 38,4 Kbps. Uma vez que os sensores considerados nesta dissertação (*Micaz* e *TelosB*) têm uma taxa de transmissão de 250 Kbps, recorrendo à documentação [PCG+05] verificou-se que em média o tempo de transmissão passa a ser de 6,5 milissegundos, o que corresponde a um valor de 260 no simulador. Repare-se que o tempo de transmissão de cada mensagem mantém-se constante, uma aproximação mais realista obrigaria a que este tempo tivesse variações, porém a implementação do JProwler utiliza sempre o tempo de transmissão médio para enviar um pacote TinyOS completo, sendo então mantido pela WiSeNet.

Uma outra parametrização necessária deve-se ao facto dos sensores *Micaz* e *TelosB* terem um alcance máximo de 100 metros no exterior, o que pressupõe que a intensidade máxima do sinal seja regulada de modo a representar este alcance. A WiSeNet mantém algumas opções da intensidade do sinal já predefinidas em função do alcance pretendido, porém nenhuma delas corresponde ao alcance de 100 metros, e portanto através de experimentações verificou-se que o valor 1300 no simulador é o que reflecte tal alcance.

Com a possível injecção de ataques é também importante indicar qual o comportamento correspondente a cada ataque, passando a ser incluído na fábrica de nós, onde agora se associam os vários tipos de ataques e os comportamentos correspondentes. A fábrica de nós especifica ainda alguma informação acerca da estação-base, uma vez

que esta difere dos restantes nós, por exemplo na sua camada de aplicação e de encaminhamento.

4.5 Complexidade da implementação

A tabela 4.1 apresenta uma visão geral da complexidade associada à implementação dos vários módulos e componentes que foram adicionados e constituem a plataforma desenvolvida, sendo essa complexidade expressa através do número de linhas de código utilizadas por cada *package*.

Tabela 4.1: Número de linhas de código utilizadas em cada *package*.

Package	**Nº linhas de código**
InjectionAttacks	1900
Coverage	1560
Energy	1960
Latency	1520
Reliability	1360
Visualization	70
SecurityLayer	655
GUI	6740
Utils	325
Total	**16090**

Os módulos de cobertura, latência, fiabilidade, energia e injecção de ataques foram os que implicaram maior complexidade e dispêndio de tempo de trabalho no desenvolvimento da plataforma. No *package* "gui" os componentes gráficos relativos às medições também exigiram algum tempo de trabalho, no sentido de definir apropriadamente a interface gráfica que permite o utilizador facilmente realizar as medições pretendidas. A camada de segurança, embora com menor número de linhas de código, necessitou de algum cuidado para que seguisse correctamente a especificação.

5

Implementação dos protocolos de encaminhamento

Neste capítulo são apresentados os protocolos de encaminhamento implementados nesta dissertação, através de uma visão algorítmica que aborda os detalhes das respectivas implementações. Tal como referido em 2.2.4, os protocolos considerados são o Clean-Slate e o INSENS por serem sistemas que abordam a problemática da defesa pró-activa, indo ao encontro dos ataques anteriormente especificados no modelo de adversário. Não obstante, foi ainda implementado o protocolo de encaminhamento Flooding (sem mecanismos de segurança integrados), o qual surge como exemplo de referência para o encaminhamento mais básico que poderá ocorrer, e permitindo assim realizar algumas comparações face aos outros protocolos. No final do capítulo é feita uma abordagem à implementação dos vários ataques nos protocolos INSENS e Clean-Slate.

5.1 Flooding

O protocolo de encaminhamento Flooding tem como objectivo realizar o encaminhamento para um determinado nó por inundação de mensagens na rede, o que logo à partida se pressupõe como bastante ineficiente devido ao elevado número de mensagens enviadas para entregar uma simples mensagem no nó de destino.

Os protocolos Clean-Slate e INSENS incluem a fase de configuração que cria as tabelas de encaminhamento, e posteriormente a fase de encaminhamento de mensagens pelos caminhos determinados. Assim, no protocolo Flooding implementou-se a fase de descoberta de nós vizinhos que incide nessa fase de configuração do protocolo, precedendo o encaminhamento de dados por inundação. Portanto, a principal ideia para o encaminhamento de dados neste protocolo consiste em cada nó, assim que receba uma mensagem, encaminhá-la em direcção aos nós vizinhos que ele conheceu na fase de descoberta, e assim sucessivamente até atingir o nó de destino.

Este protocolo não inclui mecanismos de segurança durante a sua fase de configuração, uma vez que se trata de um protocolo básico, no entanto considerou-se a utilização da camada TinySec durante a fase de encaminhamento de mensagens, criando pacotes TinySec que garantem as propriedades de segurança necessárias face a ataques externos.

O protocolo contém dois tipos de mensagem:

- *Hello* - Mensagem para a descoberta de vizinhos;
- *Data* - Mensagem de dados que contém os próximos vizinhos que deverão encaminhar a mensagem.

Listagem 5.1: Recepção da mensagem *Hello* num nó X.

```
1 protected void isHelloMessage(HelloMessage m){
2   if(!enviou){
3       envia nova HelloMessage
4       enviou = true;
5   }
6   considerar o emissor da mensagem como vizinho de X
7 }
```

Como se observa em 5.1, através de uma visão sintetizada do comportamento, quando uma mensagem *Hello* é recebida no nó X, este passa a considerar sempre o emissor da mensagem como seu vizinho, e verifica-se ainda se o nó X já enviou uma mensagem de *Hello*, enviando-a caso tal não tenha acontecido.

Na recepção de uma mensagem de dados em 5.2, inicialmente verifica-se se a mensagem é duplicada, caso seja então é descartada. Se trata-se de uma mensagem não duplicada, no caso do nó de destino ser o nó X, este entrega-a à aplicação, caso contrário verifica-se se o nó X é um dos nós vizinhos para encaminhar a mensagem, através da própria mensagem que indica quais os nós vizinhos do emissor que devem encaminhá-la.

Listagem 5.2: Recepção da mensagem *Data* num nó X.

```
protected void isDataMessage(DataMessage m){
  if(!mensagem duplicada){
      if(X é nó de destino)
          entrega mensagem na aplicação
      else if(X é um nó vizinho para encaminhar)
          envia mensagem
  }
}
```

5.2 INSENS

A implementação do protocolo INSENS assenta em quatro fases principais: fase de pedidos, respostas, distribuição de rotas e a fase de encaminhamento de dados.

No INSENS a estação-base tem um comportamento bastante particular e distinto de todos os outros nós vulgares da rede, o que pressupõe que esta seja implementada diferentemente. O seu comportamento também se enquadra nas quatro fases de implementação, embora procedendo de outra forma. Sabendo-se que ocorre a partilha de chaves simétricas entre a estação-base e cada um dos nós, bem como valores associados ao mecanismo de *one-way-sequences*, como especificados em 2.2.2, então previamente à execução do protocolo é essencial que os segredos criptográficos sejam pré-configurados nos nós pela própria estação-base.

O protocolo utiliza quatro tipos de mensagens, de acordo com as quatro fases da implementação:

- *Request* - Mensagem de pedido que permitirá os nós conhecerem os seus vizinhos;
- *Feedback* - Mensagem de resposta que permite a estação-base conhecer a vizinhança de cada nó;
- *Routing* - Mensagem para o envio das tabelas de encaminhamento;
- *Data* - Mensagem de dados para um determinado destino.

Fase de pedidos

Nesta fase o papel da estação-base é gerar uma mensagem de *request* que será recebida e propagada pelos restantes nós da rede. Os nós da rede quando recebem esta mensagem verificam através do mecanismo *one-way-sequence* se não se trata de um duplicado. Caso não seja um duplicado, então o nó pode registar quem foi o emissor da

mensagem (o que permite determinar um caminho desde a estação-base até ao nó corrente), gerar uma nova mensagem de *request* para os outros nós e agendar um evento para que mais tarde entre na fase de *feedback*. Por cada mensagem de *request* recebida no nó X, o emissor da mensagem é adicionado ao grupo de vizinhos desse nó X.

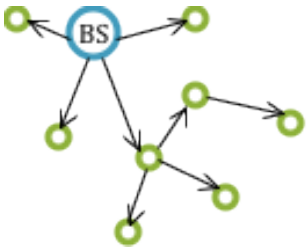


Figura 5.1: Árvore obtida após troca de mensagens *request*.

Após esta fase o resultado pode ser exemplificado na figura 5.1, a qual ilustra os caminhos que foram criados desde a estação-base até aos restantes nós. Contudo, estes caminhos são criados apenas num sentido (origem na estação-base), não sendo garantido que exista bidireccionalidade nas ligações de modo a permitir que as mensagens circulem no sentido inverso.

Uma vez que na fase seguinte (fase de *feedbacks*) é essencial que as mensagens circulem no sentido inverso (a estação-base é o destino), a não bidireccionalidade das ligações provocaria partições na árvore que teriam um impacto bastante negativo no desenrolar do protocolo. Portanto, para contrariar de certo modo este problema, na actual implementação assume-se que apenas mensagens de *request* que sejam emitidas com uma intensidade de sinal considerável podem ser aceites para formar caminho. Isto porque em função da intensidade do sinal que o emissor está a emitir é possível inferir a proximidade entre os nós, e consequentemente existindo uma maior proximidade entre nós a bidireccionalidade da ligação é mais provável de ocorrer.

Fase de respostas

Nesta fase cada nó envia a sua mensagem de resposta em direcção à estação-base para que esta receba a informação dos vizinhos desse nó. A estação-base centraliza a informação da topologia de rede por representação de um grafo unidireccional, então quando recebe uma mensagem de *feedback* aplica as operações criptográficas necessárias para confirmar a validade da mensagem e actualiza o grafo com a respectiva informação recebida, verificando a consistência dessa informação. Uma vez que é a estação-base que agenda o evento para iniciar o processo de cálculo de rotas, sempre que uma mensagem deste tipo é recebida o evento é reagendado para mais tarde, pois

ainda podem existir mensagens de *feedback* por receber e interessa que todas sejam recebidas.

Listagem 5.3: Recepção da mensagem *Feedback* num nó X.

```
protected void isFeedbackMessage(FeedbackMessage fm) {
  if(Se X é o nó pretendido e não é mensagem duplicada){
      if(pode enviar mensagem)  //controlo de envio de mensagens
          envia-a para o seu ``pai'' na árvore
      else
          agenda o envio da mensagem para mais tarde
  }
}
```

Do lado dos restantes nós, estes quando recebem uma mensagem de *feedback* actuam como apresentado na listagem 5.3. Os nós encaminham estas mensagens através dos caminhos pré-determinados na fase anterior mas tendo por destino a estação-base. Portanto, o nó X envia a mensagem de *feedback* para o seu "pai" se X é o nó pretendido para encaminhá-la e se a mensagem não é um duplicado. Porém, existe um mecanismo de controlo da taxa de emissão das mensagens que evita que as mensagens sejam enviadas a ritmos muito altos, como protecção a susceptíveis ataques desse tipo, logo envia-a se o mecanismo o permite, caso contrário é agendado um evento para a enviar mais tarde.

Fase de distribuição de rotas

Relativamente à terceira fase, que diz respeito à fase de distribuição de rotas, a estação-base tem um papel bastante importante, no sentido em que a determinada altura um evento é despoletado e origina o cálculo das tabelas de encaminhamento através do grafo que foi construído. A criação das tabelas é conseguida por intermédio do algoritmo de Dijkstra que é aplicado ao grafo, e permite assim determinar um ou dois caminhos entre cada nó e a estação-base: o caminho mais curto e o caminho redundante (sempre que possível).

Após a obtenção das tabelas, estas devem ser enviadas para os nós de forma iterativa, dando prioridade aos nós mais próximos da estação-base. Isto leva a que os nós colaborem e encaminhem as tabelas até aos nós mais distantes, por utilização das tabelas já distribuídas até ao momento. Assim, o tratamento da recepção de uma mensagem de *routing* por parte de um nó consiste em encaminhá-la para outro nó, por utilização da sua tabela de encaminhamento já determinada, ou simplesmente tratar-se de ser o nó que deve receber a tabela.

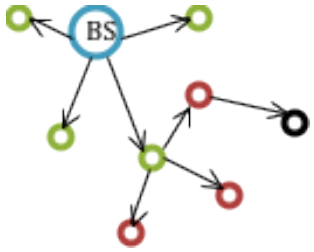


Figura 5.2: Propagação das tabelas de encaminhamento.

A figura 5.2 exemplifica a propagação das tabelas de encaminhamento, tal que os nós com a cor verde são os primeiros a receberem as tabelas, de seguida a estação-base envia as tabelas dos nós de cor vermelha, onde o nó de cor verde terá o papel de as encaminhar, e assim sucessivamente para o nó de cor preta.

Fase de encaminhamento de dados

Por fim, após a fase de configuração do protocolo ocorre a fase de encaminhamento de dados, tal que quando um nó recebe uma mensagem de dados ele simplesmente verifica se é o nó de destino da mensagem, entregando-a à aplicação, ou se é um nó encaminhador para o destino, encaminhando-a.

Este protocolo utiliza a camada TinySec estendida, pois durante o tratamento de mensagens, por vezes, é necessário realizar operações que não estão cobertas pela camada TinySec de base, por exemplo operações que envolvem *one-way-sequences*. Todas as parametrizações ao nível da segurança que não estejam mencionadas na documentação do protocolo [JD02] seguem as especificações da camada TinySec. Nesta fase de encaminhamento de mensagens não são utilizados pacotes TinySec, pois a documentação específica a forma como as mensagens podem ser enviadas de forma segura através dos segredos criptográficos assumidos pelo protocolo, como é o caso das chaves partilhadas entre a estação-base e cada nó.

O protocolo requer a configuração dos seus parâmetros, quando um nó recebe a primeira mensagem de *request* determina-se aleatoriamente, num intervalo de tempo (0 a 2 segundos), o período de espera até enviar a sua mensagem de *request*, isto para que se reduza o número de colisões. Segue-se o tempo de espera para enviar a mensagem de *feedback* (agendada após ser recebida a primeira mensagem *request*), sendo também aleatório (70 a 90 segundos). As mensagens de *routing* são também encaminhadas com um período de espera aleatório (0 a 5 segundos). Quanto à estação-base, esta utiliza um parâmetro que indica o período de espera (10 segundos) após a última mensagem de *feedback* recebida, para dar início ao cálculo das tabelas de encaminhamento, e contém ainda um parâmetro que determina a velocidade de envio de cada tabela.

5.3 Clean-Slate

No Clean-Slate todos os nós são implementados da mesma forma, excepto o *sink node*/estação-base que consiste numa extensão ao comportamento dos restantes nós. Tal como explicado em 2.2.1, o protocolo assume a existência de uma autoridade de rede, sendo portanto determinante que ocorra uma **pré-configuração** dos segredos criptográficos, o que em termos de implementação fica ao abrigo da própria estação-base que distribui pelos nós os segredos que a autoridade de rede lhe fornece.

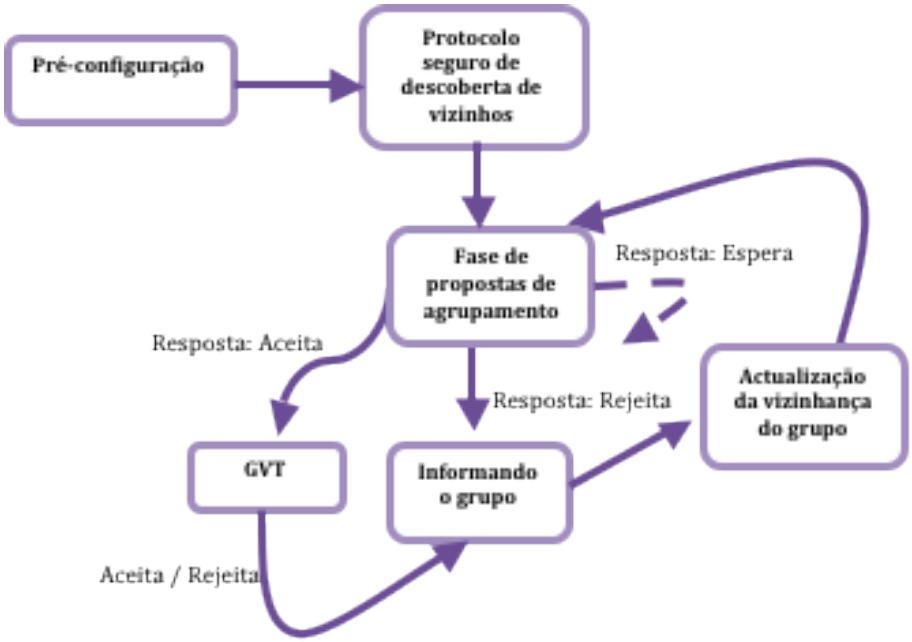


Figura 5.3: Visão geral do protocolo Clean-Slate.

A visão geral do protocolo é ilustrada na figura 5.3, a qual representa as principais fases que constituem o protocolo.

Protocolo seguro de descoberta de vizinhos

Após a pré-configuração e dada a ordem ao *sink node* para iniciar a execução do protocolo, este envia uma mensagem *Hello* que dá início a um protocolo seguro de descoberta de vizinhos. Uma das assumpções do Clean-Slate é a bidireccionalidade das ligações entre nós, ora isso terá de ser tratado logo durante a fase de descoberta de vizinhos, e uma vez que a documentação [PLGP06] não especifica concretamente uma solução para o problema, foi necessário criá-la, sendo apresentada de seguida e ilustrada na figura 5.4.

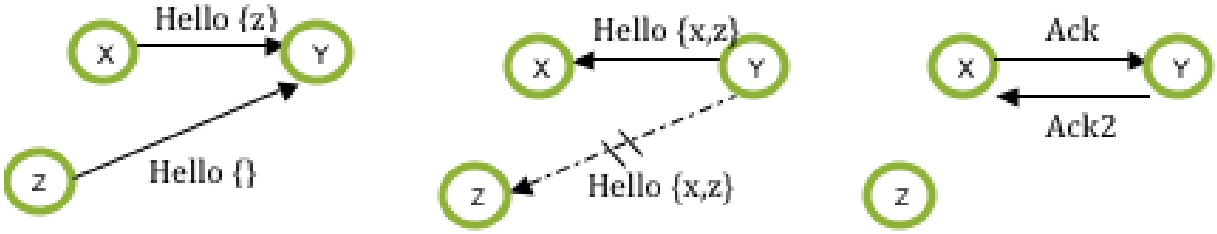


Figura 5.4: Protocolo de descoberta de vizinhos.

Na primeira imagem, os nós enviam mensagens de *Hello* que contêm convites com os identificadores dos nós que já conheceram antes, neste caso o nó *X* envia a mensagem *Hello* com o Z no convite, enquanto o nó *Z* envia a mensagem sem nenhum convite. Os nós quando recebem uma mensagem *Hello* registam o seu emissor, o que acontece com o nó *Y* que regista os nós *X* e *Z*. Agora quando o nó *Y* enviar a sua mensagem de *Hello* entregará os convites com os identificadores dos nós que conheceu anteriormente, ou seja, os nós *X* e *Z*, como se observa na segunda imagem. Em seguida, quando o nó *X* recebe uma mensagem *Hello* (já depois da sua ter sido enviada) verifica se o seu identificador está incluído nos convites {x,z}, uma vez que isso acontece ele tem garantias de que existe bidireccionalidade. O nó *Z* não conseguiu receber uma mensagem de *Y*, logo não admite bidireccionalidade. Portanto, por agora o nó *X* tem conhecimento da bidireccionalidade, mas o nó *Y* ainda não, então *X* envia uma mensagem *Ack* para *Y* que lhe dá essa informação e é retransmitida até que receba um *Ack2* de volta com a confirmação da entrega. No final, ambos os nós têm conhecimento da bidireccionalidade e consideram-se vizinhos.

Uma vez que se trata de um protocolo seguro de descoberta de vizinhos, são incluídos mecanismos de segurança na troca de mensagens. De referir que os *acknowledges* começam a ser enviados após um determinado período de tempo configurável, o mesmo acontece para que o nó entre na fase de propostas de agrupamento.

Fase de propostas de agrupamento

Esta fase diz respeito ao agrupamento recursivo que é conseguido através do determinismo, logo importa a fiabilidade da entrega de mensagens. A fiabilidade da entrega de mensagens não é uma questão abordada na documentação do protocolo, portanto mais uma vez foi necessário criar uma solução nesse sentido. O envio de mensagens fiáveis é implementado através de *acknowledges*, sendo que a mensagem é retransmitida até ser recebido o respectivo *acknowledge*, e para que não sejam consideradas mensagens duplicadas foi necessário implementar a detecção de duplicados.

Listagem 5.4: Recepção em X de uma proposta de agrupamento vinda de Y.

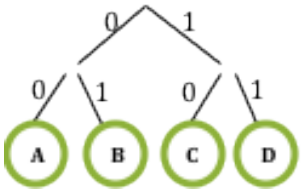
```
protected void isMergePropMessage(MergePropMessage m){
  {...}
  if(nó Y é vizinho de X){
    if(grupo de destino == grupo de X){
        if(grupo bloqueado && agrupou){
            envia resposta waiting; return;
        }
        else if(grupo bloqueado && !agrupou && aceita proposta){
            envia resposta de aceitação; return;
        }
        else if(grupo bloqueado && !agrupou && !aceita proposta)
            insere o grupo emissor no conjunto de grupos vizinhos;

        if(!grupo bloqueado && !tem grupos vizinhos){
            envia resposta de waiting;
            insere o grupo emissor no conjunto de grupos vizinhos;
        }
        else if(aceita proposta)
            proposta aceite;
        else {
            actualiza os grupos vizinhos;
            envia mensagem de waiting;
        }
    }
    else
      {... Rejeita proposta ...}
  }
}
```

Inicialmente cada nó constitui um único grupo, e cada nó é responsável por enviar propostas fiáveis de agrupamento a outros grupos, a listagem 5.4 apresenta em pseudocódigo o tratamento de uma proposta de agrupamento. Se é recebida uma proposta de um nó Y que é vizinho bidireccional de X, e se o grupo de destino da proposta é o grupo corrente do nó X, então várias hipóteses podem ocorrer. A variável "grupo bloqueado" indica que o grupo está em processamento e não pode sofrer alterações, enquanto "agrupou" indica se este nó já realizou o agrupamento em questão.

A recepção de uma proposta de agrupamento provoca três tipos de resposta: (i) resposta de espera que sugere ao grupo emissor que espere e tente novamente; (ii) resposta de rejeição, em que a proposta é rejeitada e o grupo emissor deverá tentar agrupar-se a outro grupo; (iii) resposta de aceitação que desencadeia o possível agrupamento. A resposta de aceitação pode ser considerada como uma proposta no sentido inverso, tal como a documentação sugere, contudo determina no imediato que irá ocorrer um possível agrupamento entre ambos os grupos. Qualquer proposta de agrupamento pode ser retransmitida no caso de não ser recebida uma resposta, isto deve-se ao mecanismo de fiabilidade implementado.

Informando o grupo - *Broadcast* fiável

Por vezes, o nó que emite uma proposta para agrupar recebe uma resposta de rejeição que leva à necessidade de dar a conhecer essa resposta a todos os nós pertencentes ao seu grupo, o mesmo se aplica quando o agrupamento é aceite. Uma vez que esta informação é pertinente para todo o grupo, no sentido de manter a sua consistência, requer-se que esse nó realize um *broadcast* para todo o seu grupo de forma fiável. A implementação de *broadcast* fiável segue a sugestão da documentação de referência do protocolo [PLGP06].



Prefixo	Next-hop
1.*	C (1.0)
0.1	B (0.1)

Figura 5.5: Árvore de endereços do grupo e tabela de encaminhamento do nó *A*.

Portanto, a ideia será manter o mesmo mecanismo de fiabilidade de entrega de mensagens já mencionado, através das retransmissões e *acknowledges*, mas agora trata-se de enviar mensagens para todo o grupo, o que será possível por utilização das tabelas de encaminhamento já construídas até ao momento. Por cada agrupamento as tabelas de encaminhamento são actualizadas, oferecendo assim uma forma de envio de mensagens para qualquer destino dentro do próprio grupo.

Mais concretamente, as tabelas de encaminhamento mapeiam cada prefixo de endereço para um determinado *next-hop* para que este possa encaminhar a mensagem em direcção aos nós com o respectivo prefixo. O domínio de endereços dos nós do grupo pode ser visualizado na figura 5.5 como uma árvore, assim o objectivo passa por percorrer a árvore de modo a que todos os nós do grupo sejam alcançados, utilizando os prefixos do mais geral para o mais específico e por intermédio de mensagens fiáveis.

Actualização da vizinhança do grupo

Durante as fases anteriormente descritas, alguns nós escutam mensagens destinadas a outros nós, como tal cada nó aproveita esse facto para verificar qual o grupo do nó emissor da mensagem e se possível actualiza a sua estrutura de novos grupos vizinhos pendentes. Não obstante, após um agrupamento ou uma rejeição no grupo, este deve entrar em fase de actualização da informação da sua vizinhança, sendo nesse momento que importam os novos grupos que foram conhecidos por cada nó, de modo

a manter a informação da vizinhança consistente e actualizada.

Para realizar esta actualização do grupo na sua íntegra, todos os nós que conheçam pelo menos um grupo vizinho devem, de forma espaçada no tempo, lançar um *broadcast* fiável a todo o seu grupo, indicando quais os grupos vizinhos que são do seu conhecimento. Assim, todos os nós do grupo recebem garantidamente a informação de toda a vizinhança de forma coerente e actualizada, e poderão avançar para uma nova fase de envio de propostas de agrupamento.

Árvore de verificação de grupos (GVT)

Quando um grupo aceita uma proposta de junção, o acontecimento é comunicado ao grupo emissor da proposta, de modo a que este inicie o processo de verificação da árvore de grupos ou GVT (*Group Verification Tree*). Por conseguinte, apenas após o tratamento da GVT o agrupamento poderá ocorrer, dependendo ainda do sucesso dessa verificação. A implementação deste mecanismo evita inconsistências durante o agrupamento recursivo, pois a junção de grupos apenas ocorre quando ambos os grupos conseguem responder correctamente aos desafios de segurança colocados entre eles.

Este mecanismo segue um determinado conjunto de passos e está implementado através de *hash trees* que representam de uma forma conceptual a árvore de agrupamentos já efectuados. A árvore é ilustrada na figura 5.6, onde cada nó da árvore contém o identificador do grupo, o qual é calculado em função dos seus nós descendentes por aplicação de uma função de *hash*, já a raiz da árvore representa o identificador do grupo corrente, enquanto as folhas da árvore dizem respeito ao identificador inicial de cada nó, antes de qualquer agrupamento. Tem-se também como pressuposto que qualquer nó sempre que se junta a outro grupo armazena numa estrutura o identificador desse grupo a que se juntou. Portanto, o nó A regista B e V_2 como grupos aos quais se juntou, destacados em 5.6.

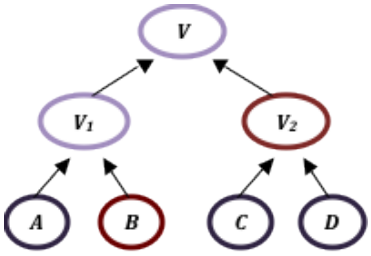


Figura 5.6: Construção de uma *hash tree*.

Relativamente às etapas inerentes ao mecanismo de GVT para os grupos denominados por G e G', a ideia consiste em G aplicar uma operação de verificação da *hash tree* de G' que determina a consistência do seu agrupamento anteriormente realizado. Exemplificando, esta operação reside em verificar a expressão $V = \mathrm{h}\,(\,\mathrm{h}\,(\,A \parallel B\,) \parallel V_2\,)$, onde o respondedor sujeito ao desafio é o nó com o identificador A.

Portanto, inicialmente selecciona-se um desafiador entre os nós do grupo G de modo a que este produza um desafio C_k que deverá ser enviado para o grupo G', sendo que o desafio determina qual o nó do grupo G' que será o respondedor do desafio. O respondedor envia o seu certificado de identidade e envia a sua tabela de agrupamentos, que irá permitir aplicar a operação de verificação da *hash tree* por parte do grupo G. O desafiador corresponde ao nó que tenha um endereço de rede prefixo de $\mathrm{F}(\mathrm{ID}_G)$, sendo F uma função *hash* e o ID_G o identificador do grupo G; já o respondedor é determinado da mesma forma, mas usando o desafio recebido na seguinte expressão: $\mathrm{F}(\mathrm{ID}_G\,, |G|\,, \mathrm{ID}_{G'}\,, |G'|\,, C_k)$. O mecanismo pode ser sintetizado em cerca de dez passos, sendo que estes devem ocorrer simultaneamente em ambos os grupos para que a verificação ocorra nos dois sentidos e no final o agrupamento seja aprovado ou rejeitado. Este conjunto de etapas exigiu diversos tipos de mensagens a fim de se cumprirem todos os passos exigíveis.

Resumo da implementação da GVT

(Grupo G agrupa-se com G' e nó X em G recebe uma proposta de um nó X' em G')
1: Escolhe um nó desafiador C no grupo G;
2: Desafiador selecciona um desafio C_k e envia-o por *broadcast* para o seu grupo G;
3: Nós em G verificam se o desafio C_k é válido;
4: Quando o desafio atinge o nó X, este envia-o para o nó X' em G';
5: Baseado no desafio C_k determina-se o respondedor no grupo G';
6: Respondedor envia o seu certificado de identidade e a tabela de agrupamentos para o nó X';
7: X' envia a resposta para o nó X;
8: O nó X verifica a GVT através da resposta obtida;
9: X envia resposta final para X' (sucesso / não sucesso);
10: Grupo G' efectua o mesmo processo concorrentemente, no final agrupam-se ou ocorre rejeição.

Outros detalhes de implementação

A questão da recuperação da rede foi tida em conta, por implementação da técnica *Honeybee* que remove nós maliciosos da rede. Quando um nó detecta um comportamento malicioso é feito um anúncio por *broadcast* e os restantes nós removem da sua

tabela de encaminhamento o anunciador e o nó malicioso. Contudo, esta técnica pode ter poucos efeitos práticos nesta dissertação, uma vez que por exemplo a detecção de nós maliciosos em certas circunstâncias torna-se uma tarefa complexa, como é o caso da detecção de replicações que vai além da implementação base do protocolo.

No que diz respeito à fase de encaminhamento de dados, esta é conseguida por intermédio das tabelas anteriormente calculadas, contudo existem diferentes formas de os encaminhar, dependendo da implementação abordada. Durante a junção de dois grupos pode ocorrer a existência de mais que um *next-hop* para alcançar o outro grupo, como tal mantêm-se três possíveis *next-hops* (L,M,R) por cada entrada da tabela de encaminhamento. Isto permite que o encaminhamento seja feito, por exemplo, com base numa direcção determinada pelo emissor. Porém, nesta implementação foi utilizada outra opção que passa por utilizar uma direcção totalmente aleatória. Além da direcção, o encaminhamento poderia ser feito com base nas distâncias para os possíveis *next-hops*, o que não foi o caso.

Um detalhe relevante é o facto de a autoridade de rede utilizar criptografia assimétrica, o que é indesejável neste tipo de redes. Porém, as operações são estabelecidas através do algoritmo criptográfico Rabin que se revela bastante eficiente, e assim considerou-se que o custo energético de tais operações é relativamente semelhante ao da criptografia simétrica. A camada TinySec será responsável por criar mensagens seguras na fase de envio de dados, pois a documentação não sugere nenhum mecanismo intrínseco ao protocolo que ofereça segurança nesta fase, logo considera-se que os dados são protegidos de ataques externos através de pacotes TinySec.

5.4 Injecção de ataques aos protocolos

Nesta secção descrevem-se, em termos de implementação, os vários ataques especificamente contemplados para serem submetidos a testes e avaliação nos protocolos de encaminhamento. A implementação destes ataques é feita por uma redefinição parcial do comportamento dos nós na camada de encaminhamento, de acordo com o respectivo protocolo e visando as características do ataque considerado.

Hello-flooding

A implementação do ataque *hello-flooding* considera que o atacante gera tráfego legítimo, no entanto tem condições de transmissão bastante superiores aos restantes nós,

induzindo em erro a noção de vizinhança e daí ser visto como um atacante do tipo *laptop*. Este ataque apenas incide na fase em que os nós enviam pacotes *hello*, pois nas restantes fases do protocolo o ataque não tem qualquer tipo de comportamento. Existe um parâmetro configurável que indica o raio de transmissão que o nó atacante consegue atingir, tal que todos os nós que se encontrem dentro desse raio de distância são alcançáveis. O raio de transmissão predefinido é 250 metros, superior ao alcance habitual. Considerou-se ainda nesta implementação que nós atacantes deste tipo não trocam mensagens *hello* entre si, pois o objectivo é induzir em erro os nós legítimos, e que é desprezável o gasto energético da transmissão destes nós, devido às suas capacidades de transmissão.

O ataque foi implementado para os três protocolos anteriores, no caso do protocolo Flooding ocorre durante o envio das mensagens *hello*, já no protocolo INSENS ocorre durante o envio das mensagens *request*, e por fim no Clean-Slate redefine-se o tratamento das mensagens *hello* que ocorrem no protocolo de descoberta de vizinhos.

Wormhole

O ataque *wormhole* requer uma afectação entre nós, pois o ataque é conseguido através da cooperação destes para que formem túneis por onde serão enviadas as mensagens. Porém, visto que o ataque *wormhole* pode ser realizado com comportamentos distintos, implementaram-se três tipos:

1. **wormhole-simples:** A cada dois nós atacantes forma-se um túnel, sendo que estes desempenham o protocolo igualmente aos nós legítimos, a diferença é que o túnel permite realizarem-se posteriormente ataques arbitrários e possui condições de transmissão bastante superiores que poderão ter impacto nos vários critérios de medição. De modo a analisar a vulnerabilidade do protocolo devem ser também contabilizadas as mensagens que passam nos túneis.

2. **wormhole-mitm (man-in-the-middle):** Consiste em provocar uma situação semelhante a um ataque *man-in-the-middle*, de tal forma que irá proporcionar que dois nós acreditem erradamente que são vizinhos, pois quando a mensagem é recebida por uma extremidade do canal ela é imediatamente enviada para a outra extremidade e propagada. Esta situação decorre durante toda a fase de configuração do protocolo, iludindo os nós acerca das suas vizinhanças e produzindo implicações graves na fase de encaminhamento de dados porque nesta implementação os túneis deixam de estar activos durante o encaminhamento de dados, com o objectivo deste ataque criar partições na rede. Refira-se que os nós

atacantes não participam no protocolo de forma igual aos nós legítimos, pois eles apenas colaboram no túnel.

3. **wormhole-overlay:** Consiste em criar uma *overlay* através de vários *wormholes*, designando este tipo por *wormhole-overlay*. Significa assim que ao invés de existir a colaboração entre dois nós para formar um túnel, passa a existir a colaboração por parte de vários nós que formam uma rede de *wormholes*. Os nós atacantes participam no protocolo igualmente aos nós legítimos.

Considera-se para os três tipos de *wormhole* que as mensagens transmitidas neste tipo de ligações nunca se perdem, o tempo de transmissão é nulo, e o consumo energético é reduzido. Os três tipos foram implementados para todos os protocolos referidos.

Sybil

Importa desde já referir que nesta implementação do ataque *sybil* considera-se que o atacante apenas utiliza os segredos criptográficos do próprio nó capturado, ou seja, não faz uso de segredos criptográficos que estejam noutros dispositivos. É utilizado um parâmetro configurável que indica o número de identidades diferentes pelas quais se faz passar, estando predefinidas três identidades. Portanto, o nó além da sua identidade correcta dá a conhecer outras identidades que não são correctas. No caso do INSENS isso ocorre no envio da mensagem *request* que irá conter uma origem falsa, já no Clean-Slate o ataque é implementado ao nível das mensagens *Hello* que contêm também o identificador da origem incorrecto. Repare-se que este ataque não é aplicável ao protocolo Flooding, pois o facto de o encaminhamento ser realizado por inundação da rede leva a que não faça sentido um atacante tentar desviar maior parte do tráfego para si.

Sinkhole

Um atacante *sinkhole* tem como objectivo influenciar os outros nós a escolherem-no para encaminhar mensagens, de modo a desviar para si o tráfego. Logo, o ataque não é aplicável no protocolo Flooding porque as mensagens são enviadas por inundação da rede, não fazendo sentido que o nó atacante se preocupe em receber a maior parte do tráfego. No protocolo Clean-Slate o agrupamento é feito de forma determinística e na possibilidade de ocorrerem vários *next-hops* para o mesmo prefixo de endereço é escolhido um deles de forma aleatória, não se baseando em distâncias, logo inviabiliza-se a possibilidade deste ataque ocorrer. Se a implementação fosse baseada em distâncias o ataque poderia ocorrer e seria então implementado.

Relativamente ao INSENS, o ataque é aplicável no sentido de um nó atacante conseguir indicar à estação-base que é vizinho de um grande número de nós, aumentando a probabilidade de ser escolhido para fazer parte dos caminhos determinados pela estação-base, ainda assim sem garantias que isso aconteça. Nesses pressupostos, o nó atacante só envia a sua mensagem *feedback* após esperar pelo período máximo permitido, de modo a conhecer o maior número de vizinhos (dos quais recebe mensagens *request*). Outro aspecto envolvido é o envio de várias mensagens *request* (três mensagens por pré-configuração), em vez de apenas uma como estipulado na especificação, isto para que o nó seja conhecido por todos os seus vizinhos.

Um ataque *sinkhole* só por si poderá não ter um impacto significativo, contudo permite realizar posteriores ataques arbitrários, e como tal deve ser medido em termos do número de mensagens que são encaminhadas para o nó atacante.

Blackhole

A implementação do ataque *blackhole* faz a redefinição do comportamento da recepção de mensagens de dados, de modo a que qualquer mensagem de dados recebida seja sempre descartada. Quando é solicitado ao nó atacante que envie uma mensagem de dados ele descarta-a também, tal como faz quando a recebe. Este ataque aplica-se facilmente a qualquer um dos protocolos abordados.

Selective-Forwarding

O ataque *Selective-Forwarding* assemelha-se ao anterior, no entanto, o descarte não é obrigatório, pois apenas algumas mensagens de dados são descartadas. Vários critérios de descarte poderiam ser implementados, porém considerou-se o descarte por aleatoriedade tendo por base um parâmetro probabilístico que indica a probabilidade da mensagem não ser descartada. Exemplificando, tendo o parâmetro definido a 70% ocorre uma probabilidade de 70% da mensagem não ser descartada e ser, então, transmitida. O valor predefinido para o parâmetro de não descarte é 30%.

5.5 Complexidade da implementação

A tabela 5.1 apresenta uma visão geral da complexidade associada à implementação dos vários protocolos, sendo essencialmente expressa através do número de linhas de código utilizadas por cada *package*.

Tabela 5.1: Número de linhas de código utilizadas em cada *package*.

Package	**Nº linhas de código**
CleanSlate	5570
CleanSlate-attacks	3090
Insens	3430
Insens-attacks	1680
Flooding	770
Flooding-attacks	880
Total	**15420**

A implementação do protocolo Clean-Slate, pressupondo a sua concepção de base apresentada em 2.2.1, denotou-se como algo complexa quando se trata de redes com um elevado número de nós, sendo o caso desta dissertação. Alguns detalhes relevantes no protocolo são apenas assumidos teoricamente, o que colocou dificuldades em termos da sua implementação, onde por vezes foi necessário encontrar soluções que visam resolver essas assumpções. O facto de se tratar de um protocolo baseado essencialmente no determinismo, como é o caso do agrupamento recursivo, criou dificuldades acrescidas no sentido em que é indispensável uma abordagem que garanta a entrega de mensagens. Considera-se ainda uma grande variedade de mensagens que são necessárias para executar todas as etapas inerentes ao protocolo. A complexidade do Clean-Slate reflecte-se no elevado número de linhas de código necessárias na sua implementação.

Quanto ao INSENS, este também envolveu alguma complexidade, sobretudo no cálculo de rotas por intermédio de grafos representativos da rede, embora tenha sido menor que o protocolo anterior. Por último, a implementação do Flooding foi relativamente simples.

6

Testes e Avaliação

Neste capítulo de testes e avaliação apresentam-se os resultados e a respectivas análises para os vários testes realizados aos protocolos Clean-Slate, INSENS e Flooding, dando maior ênfase aos dois primeiros.

Inicialmente são apresentadas as parametrizações gerais que envolvem os testes realizados, sendo posteriormente exibidos os resultados dos testes de cobertura, fiabilidade, latência e energia em nós sensores com comportamentos correctos. De seguida, serão também apresentados os testes para os mesmos critérios mas quando estão perante vários tipos de ataques e, por fim, é ainda feita uma avaliação a outras características relativas ao encaminhamento, nomeadamente a eficácia dos ataques à selecção de rotas e o encaminhamento redundante. Em cada teste é feita uma descrição que explica em que consiste o teste e apresentam-se as parametrizações necessárias, bem como as devidas conclusões dos resultados obtidos.

6.1 Parametrizações gerais

Para os testes realizados vários aspectos e detalhes gerais de parametrização foram considerados, os quais são indicados de seguida:

i. Em relação ao número de nós utiliza-se um limite mínimo de 500 nós e um limite máximo de 8000 nós, no entanto os números utilizados dependem do respectivo teste no sentido em que as simulações podem tornar-se muito demoradas em determinadas circunstâncias;

ii. Tendo em vista as características dos sensores que se pretende simular (*Micaz* e *TelosB*), define-se a intensidade máxima do sinal no simulador com o valor 1300 de modo a reflectir o alcance real que estes sensores atingem, sendo cerca de 100 metros de alcance máximo em ambientes exteriores;

iii. Em todos os testes a fase de implantação ou *deployment* dos nós segue uma distribuição totalmente aleatória, de tal forma que os nós ficam relativamente dispersados pela área definida, e não se considera que estes nós tenham mobilidade nem que entrem e saiam durante o funcionamento da rede;

iv. A maioria dos testes requer uma percentagem de nós emissores a emitirem mensagens, sendo que esses nós emissores são sempre escolhidos aleatoriamente;

v. Relativamente aos testes que incluem ataques, utilizam-se várias percentagens de nós atacantes, sendo esses nós atacantes escolhidos também aleatoriamente. Apenas se consideram percentagens que não incorrem acima dos 50%, pois acima deste valor é improvável que tal aconteça na realidade, e quando isso se verifica a rede encontra-se quase por completo na posse do atacante, sendo pouco interesse analisar;

vi. A estação-base está sempre posicionada no centro da rede, pelas razões mencionadas mais à frente em 6.2;

vii. A partir do teste de cobertura que utiliza as dimensões das áreas ajustadas a cada protocolo, tratado em 6.2, todos os restantes testes mantêm essas dimensões, pois foram determinadas experimentalmente e revelaram-se as mais adequadas.

6.2 Cobertura

Percentagem de emissores

Para a análise da taxa de cobertura nos vários testes realizados requer-se uma determinada percentagem de emissores, escolhidos aleatoriamente, que consiga representar uma boa estimativa da cobertura real da rede. Uma percentagem de emissores bastante elevada torna-se problemática no sentido em que torna as simulações demoradas, tanto quanto maior o número de nós da rede. Já por contrário, uma percentagem muito inferior pode ser divergente em relação ao verdadeiro resultado. Assim, efectuou-se uma análise que permitiu determinar a percentagem de emissores mais adequada para avaliar a cobertura dos protocolos, sendo que 60% de emissores foi a percentagem determinada.

Influência da posição da estação-base / *sink node*

A posição da estação-base é um aspecto que, porventura, pode ter um impacto significativo, sobretudo no que diz respeito à cobertura da rede. Nos protocolos abordados é a estação-base que inicia a fase de configuração, então uma posição adequada talvez beneficie as condições de operabilidade da rede. Como tal, efectuou-se uma comparação entre o Clean-Slate e o INSENS quando a estação-base se encontra em duas posições distintas (centro e periferia da rede) e mantendo a mesma dimensão da área independentemente da posição. Isto permitirá avaliar o impacto que a posição da estação-base provoca na taxa de cobertura total em função do número de nós (500 a 2000).

Tabela 6.1: Influência da posição da estação-base.

Posição / Nº nós	CT (%) Clean-Slate	CT (%) INSENS	Dim. Clean-Slate	Dim. INSENS
Centro / 500	98,99	65,21	1000 x 1000 (m^2)	650 x 650 (m^2)
Centro / 1000	96,99	53,75	1500 x 1500 (m^2)	950 x 950 (m^2)
Centro / 2000	95,99	41,7	1900 x 1900 (m^2)	1350 x 1350 (m^2)
Periferia / 500	98,99	45,15	1000 x 1000 (m^2)	650 x 650 (m^2)
Periferia / 1000	94,32	30,55	1500 x 1500 (m^2)	950 x 950 (m^2)
Periferia / 2000	89,49	12,01	1900 x 1900 (m^2)	1350 x 1350 (m^2)

Como se pode analisar na tabela 6.1, no caso do Clean-Slate a taxa de cobertura total (CT) não sofre alterações significativas quando se muda a posição da estação-base e se aumenta o número de nós. Já no caso do INSENS o mesmo não se verifica, pois quando a estação-base se encontra no centro da rede os valores de cobertura são superiores em relação aos obtidos quando esta se encontra na periferia.

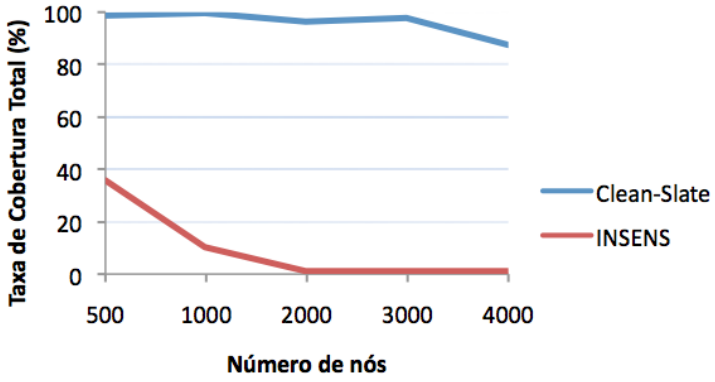
Este resultado explica-se pelo facto do Clean-Slate ser baseado em retransmissões, e assim facilmente proporciona uma boa taxa de cobertura independentemente da posição da estação-base. Por contrário, o INSENS não possui retransmissões e quando a estação-base se encontra no centro da rede há uma maior probabilidade de se formar uma árvore que é expandida em todo o seu redor, o que não aconteceria caso se encontrasse na periferia. Portanto, para todos os restantes testes efectuados a estação-base encontra-se no centro da rede.

Análise de cobertura sem ajuste da dimensão de área por protocolo

A taxa de cobertura além das características inerentes ao protocolo depende ainda de uma relação entre o número de nós e a dimensão de área utilizada. Alguns protocolos, porventura, lidam melhor em redes com vizinhanças de nós mais concentradas,

enquanto outros preferem vizinhanças mais dispersas. Um exemplo primário é o protocolo Flooding, em que logo de imediato nos apercebemos que se adequa a redes com vizinhanças bastante dispersas devido à sua capacidade de inundação de mensagens.

Portanto, este teste pretende avaliar a evolução da taxa de cobertura total dos protocolos INSENS e Clean-Slate quando as suas dimensões de área são exactamente iguais por cada número de nós utilizados, esse número varia entre 500 a 4000 e 60% deles são emissores. As áreas definidas criam uma baixa concentração de nós, ainda assim obtêm uma cobertura rádio muito próxima dos 100%, significando que praticamente todos os nós da rede têm pelo menos um nó vizinho.



Nº nós	Largura (m)	Altura (m)
500	750	750
1000	1200	1200
2000	1900	1900
3000	2200	2200
4000	2800	2800

Figura 6.1: Cobertura total sem ajuste da dimensão de área.

O gráfico em 6.1 reflecte bem a diferença entre ambos os protocolos quando é utilizada a mesma dimensão. Veja-se que o Clean-Slate tem uma cobertura muito positiva, próxima dos 100% até aos 3000 nós, enquanto o INSENS aos 1000 nós já apresenta valores muito reduzidos, evidenciando que não se adapta à mesma dimensão e requer ajustamentos que lhe permitam melhorar os seus índices de cobertura. De facto, o INSENS apropria-se a redes com vizinhanças de nós muito concentradas, já o Clean-Slate lida melhor na situação oposta. Por conseguinte, nos próximos testes realizados passa a ser utilizada uma dimensão de área ajustada experimentalmente a cada protocolo, de modo a obter os valores de cobertura apropriados mediante o número de nós da rede.

Análise de cobertura com ajuste da dimensão de área por protocolo

Contrariamente ao teste anterior, importa agora analisar a evolução da taxa de cobertura dos protocolos em função do número de nós da rede, tendo os parâmetros de dimensão da respectiva área ajustados a cada protocolo. Assim, permite-se avaliar a capacidade do protocolo em manter índices de cobertura aceitáveis quando o número

de nós da rede aumenta. A tabela 6.2 apresenta os valores experimentalmente determinados e ajustados de modo a obter a melhor cobertura e mantendo uma boa operabilidade do protocolo, sendo que no caso do INSENS determinou-se que a melhor cobertura é conseguida com uma média de cerca de 17 nós vizinhos por cada nó (com cobertura rádio de 100%), enquanto no Clean-Slate apenas cerca de 7 (com cobertura rádio próxima dos 100%).

Tabela 6.2: Dimensões ajustadas ao protocolo.

Nº nós	**Clean-Slate (m^2)**	**INSENS (m^2)**	**Flooding (m^2)**
500	1000 x 1000	650 x 650	1000 x 1000
1000	1500 x 1500	950 x 950	1500 x 1500
2000	1900 x 1900	1350 x 1350	1900 x 1900
3000	2250 x 2250	1700 x 1700	2250 x 2250
4000	2600 x 2600	1950 x 1950	2600 x 2600
5000	3000 x 3000	2200 x 2200	3000 x 3000
6000	3300 x 3300	2400 x 2400	3300 x 3300
7000	3400 x 3400	2600 x 2600	3400 x 3400
8000	3800 x 3800	2800 x 2800	3800 x 3800

Estas dimensões são fundamentalmente teóricas, no sentido em que estão a considerar que os nós têm um alcance máximo de 100 metros, quando na verdade esse alcance pode ser muito inferior por vários factores externos que afectam as comunicações. Embora o modelo de rádio do simulador tenha em consideração o enfraquecimento do sinal, que acaba por baixar o alcance, não é garantido que se traduzam exactamente as condições impostas no caso de se estar perante um ambiente real. Portanto, pressupondo que se pretendia implementar os protocolos num ambiente real, estes seriam apenas valores de referência a serem utilizados.

Este teste foi realizado para redes com 500 a 8000 nós, em que 60% deles desempenham o papel de emissores, e avalia-se a cobertura total e parcial dos protocolos Clean-Slate, INSENS e Flooding (apenas a cobertura total). As dimensões utilizadas são as referidas, e a estação-base encontra-se sempre no centro da rede.

Por observação do gráfico em 6.2, no que diz respeito à cobertura total, como se esperaria o protocolo Flooding permite uma cobertura total próxima de 100% devido à inundação de mensagens na rede. Quanto aos protocolos de maior interesse, Clean-Slate e INSENS, estes revelam diferenças significativas nas taxas de cobertura total e parcial. No Clean-Slate a taxa de cobertura total para 500 nós é bastante boa, próxima dos 100%. Com o aumento do número de nós decresce de forma relativamente moderada, nunca descendo abaixo dos 50% até aos 8000 nós. O INSENS, por sua vez, aos 500 nós tem uma taxa com cerca de 65% e decresce de modo mais acentuado até aos

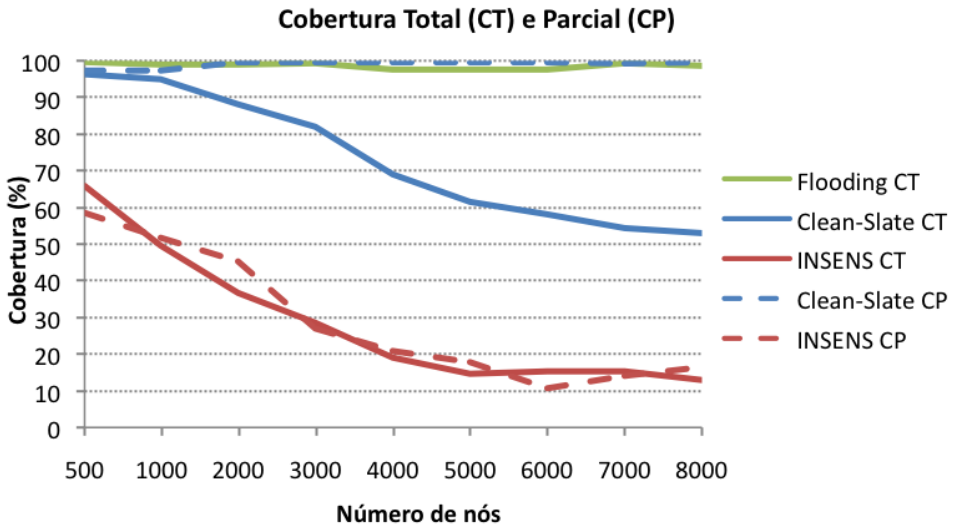


Figura 6.2: Cobertura total com ajuste da dimensão de área.

4000 nós, sendo que a partir daí a taxa já se encontra abaixo dos 20%. Estes valores explicam-se pelo facto do Clean-Slate ser um protocolo que utiliza um mecanismo de descoberta de vizinhos com garantias de bidireccionalidade e ainda um mecanismo de retransmissões que permite uma boa comunicação entre todas as vizinhanças. O INSENS não contém esses mecanismos, a sua concepção leva a que quanto maior for a distância à estação-base maior a probabilidade de o nó não estar totalmente coberto devido à perda de mensagens, agravando assim a cobertura.

Relativamente à cobertura parcial, repara-se que o Clean-Slate aproxima-se dos 100% porque normalmente os nós acabam por pertencer a algum grupo mesmo que esse grupo esteja isolado na rede. No INSENS aproxima-se da cobertura total porque se o nó é parcialmente coberto então faz parte de um caminho para a estação-base, logo também é totalmente coberto. Importa referir que os resultados apresentados constituem valores médios obtidos, pois durante as medições verificaram-se algumas oscilações dos valores, sobretudo no Clean-Slate.

6.3 Fiabilidade

A análise de fiabilidade tem como objectivo oferecer uma visão da capacidade do protocolo em garantir a entrega de mensagens emitidas por nós em direcção à estação-base, o que depende da forma como o protocolo constrói os seus caminhos e dos mecanismos utilizados para efectuar o respectivo encaminhamento. Os resultados da fiabilidade dependem ainda de alguns factores, como é o caso da cobertura, no sentido em que se requer que existam nós totalmente cobertos para que possam desempenhar o

papel de emissores. O nível de *stress* ou congestionamento da rede constitui outro factor de enorme relevância, e como tal torna-se importante avaliar a evolução da taxa de fiabilidade quando a rede se encontra perante um cenário de baixo, moderado ou alto congestionamento de mensagens. A posição da estação-base também tem influência nos resultados obtidos, mantendo-se mais uma vez no centro da rede.

A comparação entre os resultados de fiabilidade deve ser cuidada no sentido em que cada protocolo é testado nas suas condições mais apropriadas, pois utilizam-se as dimensões de área que lhe permitem obter a melhor cobertura para um determinado número de nós. Os emissores de mensagens terão de ser totalmente cobertos, então, no INSENS sabendo-se que tem uma cobertura inferior ao Clean-Slate, existe uma maior probabilidade de serem seleccionados emissores que se encontrem mais próximos da estação-base, enquanto no Clean-Slate esses emissores eventualmente podem encontrar-se mais distantes. Portanto, as comparações efectuadas pressupõem as próprias condições adequadas a cada protocolo e a sua própria cobertura.

A análise de fiabilidade interessa realizar tanto na presença como na ausência de ataques, sendo por agora apresentada sem qualquer tipo de ataque. O seguinte teste tem como objectivo analisar a fiabilidade dos protocolos Clean-Slate, INSENS e Flooding, quando estes se encontram em três possíveis cenários de congestionamento de mensagens na rede: baixo, moderado e alto. Esta análise permitirá avaliar a variação da taxa de fiabilidade por cada protocolo (nas suas condições de cobertura adequadas), para redes de várias dimensões e nos cenários de congestionamento referidos.

Tabela 6.3: Configuração dos níveis de *stress* da rede.

Tipo	**Emissores (%)**	**Período (s)**	**Nº mensagens**
Baixo	1	7	15
Moderado	5	2	15
Alto	10	0,5	15

Os três tipos de congestionamento foram configurados através de várias experimentações que permitiram obter valores que reflectem o cenário pretendido. A percentagem de nós emissores influencia bastante o nível de congestionamento, pois muitos emissores a enviarem mensagens em simultâneo aumenta a probabilidade de colisões. O período decorrido entre o envio de cada mensagem por emissor é também um factor determinante, quanto menor o período maior o número de mensagens que se encontram em simultâneo na rede. O número de mensagens enviadas por emissor mostrou ser pouco influente no nível de *stress*, contudo notou-se a importância de que várias mensagens sejam enviadas pelo mesmo emissor, de modo a que existam mensagens suficientes para oferecer uma boa estimativa de fiabilidade.

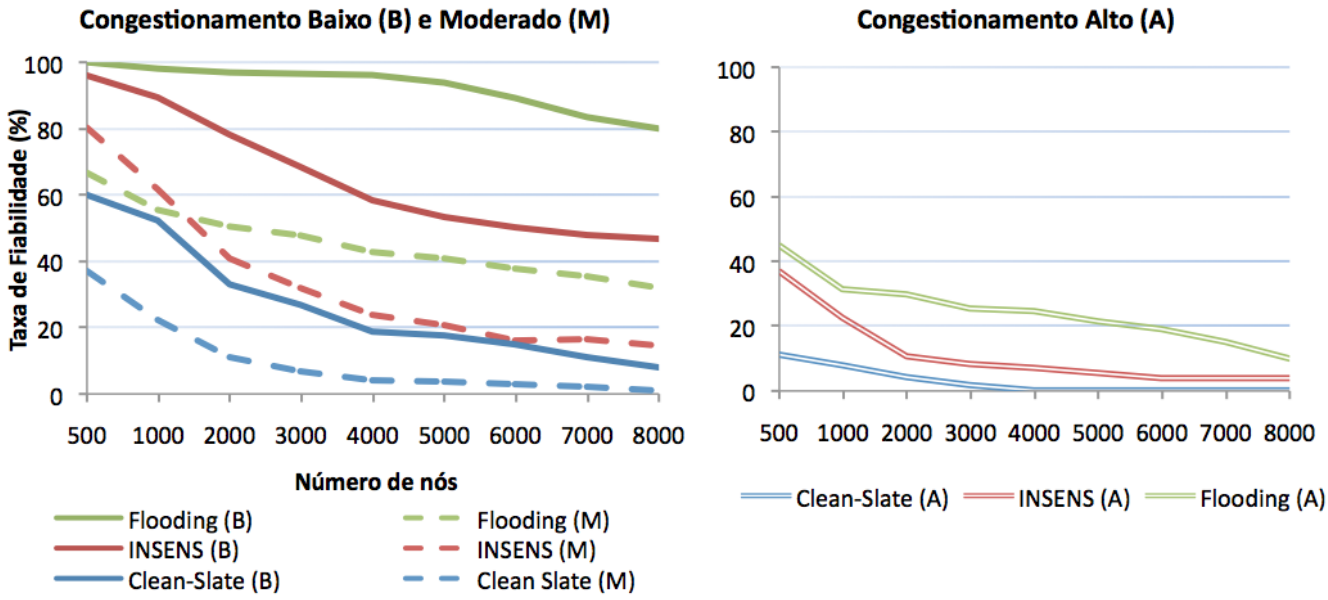


Figura 6.3: Taxa de fiabilidade dos protocolos sem ataques.

Por observação dos resultados expressos na figura 6.3 verifica-se que o protocolo Flooding é aquele que consegue garantir as melhores taxas de fiabilidade nos vários cenários de congestionamento. Com congestionamento baixo a fiabilidade do Flooding mantém-se próxima dos 100%, decrescendo ligeiramente a partir dos 4000 e atingindo os 80% aos 8000 nós. Sendo o congestionamento moderado, aos 500 nós a taxa encontra-se em cerca de 65% e decresce até perto dos 30%, enquanto que em alto observam-se valores mais reduzidos que acabam por atingir os 10% aos 8000 nós.

No caso dos protocolos INSENS e Clean-Slate (nas condições apropriadas), o primeiro revela que para um baixo nível de congestionamento a sua taxa de fiabilidade é relativamente boa, decrescendo com o aumento do número de nós mas sempre muito superior à do Clean-Slate, aos 500 nós regista valores próximos dos 100% e reduz até 45% aos 8000 nós. Quando o nível de *stress* é moderado o INSENS tende a decrescer até valores abaixo dos 20%, o que se regista a partir dos 5000 nós, no entanto mantendo ainda a superioridade em relação ao Clean-Slate. Perante um nível alto, embora o INSENS ainda seja superior ao Clean-Slate, ambos tendem a obter valores muito reduzidos, sendo que o Clean-Slate mais facilmente se aproxima do valor nulo. O Clean-Slate mesmo quando o nível é baixo apresenta uma taxa de fiabilidade de 60% para 500 nós, evidenciando bem a sua inferioridade em relação aos outros protocolos.

Os bons valores de fiabilidade obtidos para o Flooding devem-se ao facto das mensagens serem enviadas por inundação da rede, o que permite que estas cheguem ao destino por algum caminho. O INSENS cria as suas rotas tendo por base o caminho mais curto e ainda encaminha por múltiplos caminhos, ao contrário do Clean-Slate

que não utiliza qualquer noção de caminho mais curto (caminhos criados pelo agrupamento recursivo), nem envia a mensagem por caminhos distintos simultaneamente, explicando assim a diferença de resultados.

Importa referir que os valores de fiabilidade apresentados constituem valores médios esperados, uma vez que ocorrem pequenas variações quando efectuadas várias amostras sucessivas. Os restantes testes que passam a incluir ataques utilizam o nível de congestionamento moderado, isto porque um nível baixo não oferece uma visão definitivamente clara da aptidão do protocolo em lidar com este tipo de circunstâncias. Já um nível muito alto torna-se também pouco perceptivo e mais improvável de ocorrer na realidade, obtendo-se valores que poderão ser muito diminutos e pouco esclarecedores.

6.4 Latência

A análise de latência está dependente da cobertura da rede, bem como da fiabilidade, pois é necessário que existam emissores totalmente cobertos e mensagens suficientes a alcançarem a estação-base de modo a fazer uma avaliação correcta. A latência associada ao encaminhamento de uma mensagem enviada desde o emissor até à estação-base está dependente do protocolo em questão, sobretudo porque os caminhos não são criados igualmente e caminhos mais apropriados melhoram as condições de latência, como por exemplo, a redução do número de saltos que a mensagem precisa de efectuar. O nível de congestionamento constitui outro factor a ter em conta, pois a ocorrência de muitas mensagens a serem emitidas simultaneamente pode provocar colisões, bem como atrasos de emissão. Assim, para teste serão utilizados os níveis de congestionamento já anteriormente definidos na fiabilidade e descritos na tabela 6.3.

Nestes pressupostos, pretende-se testar para os protocolos Flooding, Clean-Slate e INSENS a latência média nos vários níveis de congestionamento e sem a presença de ataques, o que permitirá avaliar e determinar o protocolo que consegue entregar mensagens com a menor latência média em função do nível de *stress* e do número de nós da rede, o qual varia entre 500 a 8000.

À semelhança do anteriormente referido para a fiabilidade, os resultados obtidos para cada protocolo estão de acordo com as condições de cobertura apropriadas a cada protocolo. Os valores de latência são estimativas que se baseiam nos tempos internos do simulador, tentando sempre dar uma visão próxima da real. Visto que se pretende fazer uma comparação entre protocolos, todos eles são testados nas mesmas condições temporais do simulador, dando assim sentido a esta comparação.

Refira-se também que os resultados apresentados na figura 6.4 constituem valores médios determinados através de várias amostragens, uma vez que ocorreram algumas oscilações nas amostras. De modo a que os resultados possam ser explicados importa ainda observar a tabela 6.4 que apresenta o número de saltos médio que as mensagens necessitam de desempenhar no respectivo protocolo.

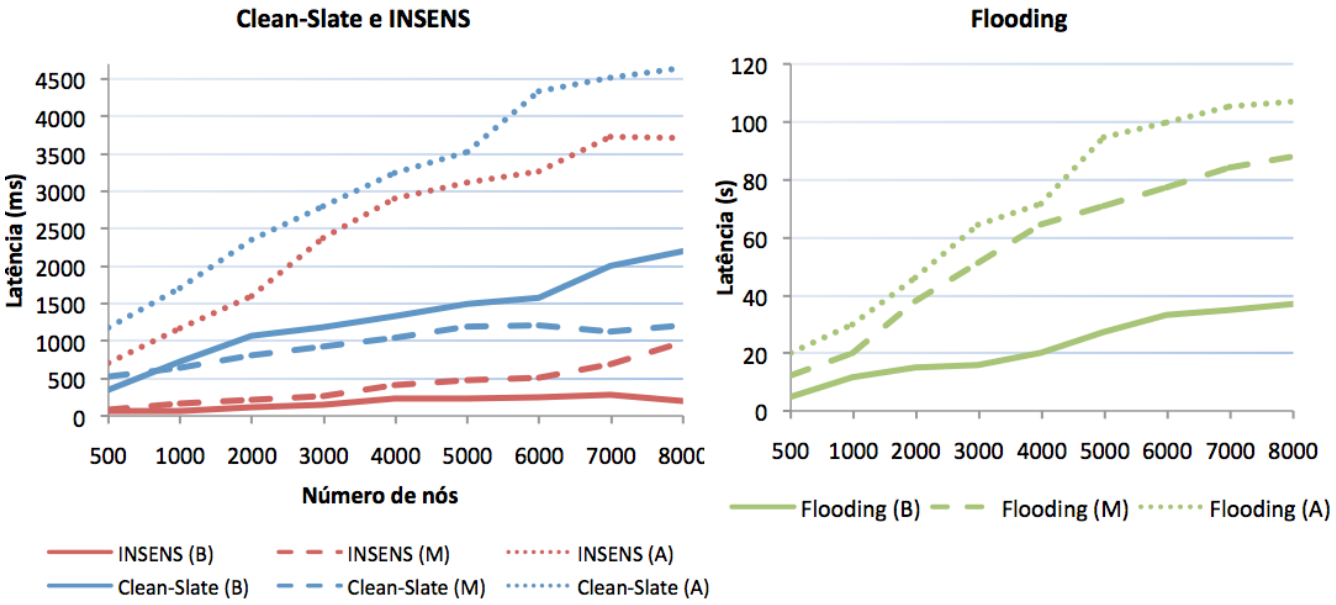


Figura 6.4: Latência dos protocolos sem ataques.

Nos gráficos da figura 6.4 observa-se que o protocolo Flooding é aquele que obteve a maior latência (medida em segundos), chegando a atingir os 107 segundos aos 8000 nós e com alto congestionamento. Relativamente ao Clean-Slate e INSENS, estes têm valores bastante inferiores (medidos na ordem dos milissegundos) quando comparados com o Flooding. O INSENS tem uma latência média inferior à do Clean-Slate, em qualquer tipo de congestionamento, repare-se que no caso de ser baixo e moderado os valores de latência para o INSENS não ultrapassam os 1000 *ms*, enquanto o Clean-Slate aos 8000 com congestionamento baixo ultrapassa os 2000 *ms*. Sendo o congestionamento alto, nota-se uma subida algo mais acentuada da latência, tendendo a crescer significativamente em função do aumento do número de nós da rede. Um detalhe interessante que se observa é o facto da latência do Clean-Slate não aumentar quando passa de baixo congestionamento para moderado, o que se deve à relação que existe com a taxa de fiabilidade, pois esta é afectada.

O Flooding embora não seja o protocolo que provoca maior número de saltos é o que apresenta maior latência porque cada *next-hop* antes de encaminhar a mensagem espera um período aleatório entre zero a dois segundos. Uma vez que se trata de um

Tabela 6.4: Número médio de saltos da mensagem.

Nº nós	INSENS	Clean-Slate	Flooding
500	4	34	9
1000	6	50	13
2000	7	79	18
3000	10	101	20
4000	12	144	23
5000	14	171	29
6000	16	186	30
7000	15	205	32
8000	15	215	35

protocolo de inundação, este período foi necessário de modo a diminuir o número de colisões, e assim em detrimento da latência permite uma boa taxa de fiabilidade. Por experiência verificou-se que no INSENS e Clean-Slate este período não melhora significativamente a taxa de fiabilidade, portanto cada *next-hop* encaminha imediatamente a mensagem, o que torna a latência média muito reduzida. No que diz respeito à diferença entre o INSENS e o Clean-Slate, esta explica-se pelo número médio de saltos no Clean-Slate ser bastante superior ao do INSENS, levando a que a mensagem demore mais tempo até atingir a estação-base. O aumento do nível de congestionamento geralmente tende a agravar a latência porque havendo maior tráfego os nós são obrigados a esperar mais vezes por um certo período de tempo antes de iniciar a transmissão até encontrarem o canal livre, isto ao nível MAC.

6.5 Energia

A energia é um factor de extrema importância nestas redes e como cada protocolo foi concebido com mecanismos próprios que exigem maior ou menor consumo energético, torna-se interessante realizar uma análise entre os vários protocolos que permita assim compará-los tanto na sua fase de configuração como na fase de encaminhamento.

Fase de configuração

O consumo energético na fase de configuração está apenas dependente da própria concepção do protocolo, ou seja, depende do consumo provocado pelos mecanismos inerentes ao protocolo e que são utilizados, consistindo no gasto necessário para a construção dos caminhos. Nesse sentido, este teste oferece uma visão que permite aferir qual dos protocolos é mais adequado para certas condições de energia, de modo a

Tabela 6.5: Consumo energético na fase de configuração.

Nº nós	Clean-Slate (Kjoules)	INSENS (Kjoules)	Flooding (Kjoules)
500	6235	359	0,31
1000	11068	1126	0,57
2000	57703	2521	1,47
3000	144996	5068	2,35
4000	187592	9263	3,10
5000	272156	10923	3,66
6000	341529	13003	4,36
7000	414585	21434	5,311

desempenhar por completo a sua fase de configuração. O teste é realizado aos protocolos Flooding, INSENS e Clean-Slate em redes de 500 a 7000 nós sem ataques.

A tabela 6.5 apresenta os resultados, evidenciando uma forte disparidade entre os três protocolos. O protocolo Flooding exige um baixo consumo energético, enquanto o INSENS requer um consumo superior, contudo é o Clean-Slate que provoca um gasto energético muito notável, altamente superior aos anteriores. Os valores do Clean-Slate devem-se ao grande número de mensagens que são trocadas, sobretudo causado pelo mecanismo de retransmissões. O Flooding nesta fase apenas procede ao conhecimento dos seus vizinhos, necessitando de um pequeno gasto energético. Por fim, o INSENS exige que se realizem certos procedimentos, no entanto o forte contributo oferecido pela estação-base reflecte-se no seu moderado consumo energético.

Fase de encaminhamento

Na fase do encaminhamento interessa avaliar e comparar a capacidade dos protocolos em construir caminhos que permitam encaminhar o tráfego com o menor gasto energético, considerando os vários cenários de congestionamento. A este nível estão relacionados os factores de cobertura e de fiabilidade.

O teste consiste em simular os protocolos Flooding, INSENS e Clean-Slate quando se encontram em redes que têm entre 500 a 8000 nós para os vários cenários de congestionamento especificados na tabela 6.3 e sem a inclusão de ataques.

Os resultados apresentados na figura 6.5 constituem valores médios de energia determinados por várias amostragens. Observando os gráficos facilmente se pode constatar que o Flooding é o que provoca maior consumo energético, na ordem dos milhares de joules, por exemplo num cenário de congestionamento moderado aos 500 nós obteve-se cerca de 1000 joules, enquanto aos 8000 já se atingem os 12000 joules. Com o aumento do número de nós o consumo energético no Flooding é muito significativo. O INSENS e o Clean-Slate revelam um consumo muito inferior ao Flooding, pois os seus

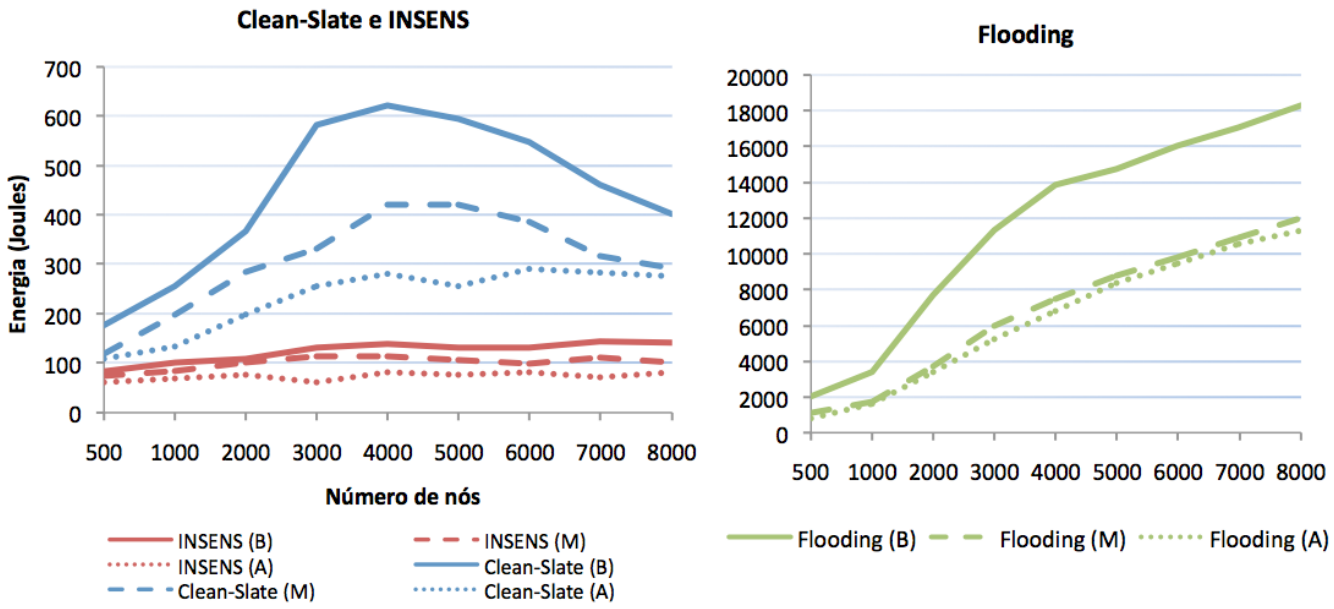


Figura 6.5: Consumo energético na fase de encaminhamento.

valores estão na ordem das centenas de joules, sendo que o protocolo INSENS é aquele que apresenta valores mais reduzidos para efectuar o encaminhamento de mensagens. Por exemplo, com congestionamento moderado o INSENS obtém um consumo entre cerca de 70 a 115 joules, ou seja, com uma baixa variação, enquanto o Clean-Slate varia entre 100 a 430 joules, o que demonstra bem a diferença entre ambos.

No que diz respeito à análise perante os vários tipos de congestionamento, verifica-se para os três protocolos um decréscimo (menos notável no caso do INSENS) do consumo energético em função do aumento do nível de *stress* da rede. Poderia pensar-se que o aumento de congestionamento provocasse maior consumo energético, contudo tal não ocorre porque esta medição está directamente relacionada com a fiabilidade, no sentido em que apenas se consideram as mensagens que alcançam a estação-base. Ora, se o congestionamento aumenta então a fiabilidade decresce, pois ocorrem mais perdas de mensagens e as que chegam à estação-base provavelmente são emitidas por emissores mais próximos desta, o que requer um menor dispêndio de energia durante o seu encaminhamento. Repare-se também que no Clean-Slate presencia-se um notável decréscimo da energia despendida a partir dos 4000 nós, o que se deve ao facto da taxa de fiabilidade ser muito reduzida a partir desse número e, assim, pelos mesmos motivos acabados de referir o consumo energético decresce.

De facto, o consumo energético está muito relacionado com o número de saltos que a mensagem efectua pela rede, isto porque por cada salto são realizados vários procedimentos que consomem energia, tais como a transmissão, recepção e operações criptográficas, e principalmente o processamento das mensagens que evidenciou um

consumo notável. Nestes termos, os resultados facilmente se justificam, pois o INSENS é um protocolo que exige um pequeno número de saltos, enquanto o Clean-Slate exige um número muito maior, já no Flooding a mensagem propaga-se por quase toda a rede (mesmo que com poucos nós alcance a estação-base). Outro aspecto que tem influência no consumo energético é o tamanho das mensagens e por análise observou-se que os tamanhos das mensagens de dados por cada protocolo são: Clean-Slate com 43 *bytes* (usa pacote TinySec), INSENS com 38 *bytes*, e no Flooding entre 36 a 56 *bytes* (usa pacote TinySec).

6.6 Injecção de ataques

6.6.1 *Hello-Flooding*

Para os vários testes que se seguem e que contemplam este ataque *hello-flooding*, os nós atacantes têm um alcance máximo de 250 metros (valor predefinido para o ataque), enquanto os restantes nós mantêm um alcance de 100 metros. Portanto, o atacante possui uma boa capacidade de transmissão, contudo a recepção de mensagens está dependente da capacidade de transmissão dos restantes nós.

Cobertura

A análise experimental ao impacto que um ataque *hello-flooding* provoca na cobertura dos protocolos INSENS e Clean-Slate é importante no sentido de avaliar a resiliência face a este ataque. Teoricamente o ataque é protegido pelo Clean-Slate e não pelo INSENS, então será verificado experimentalmente se os resultados obtidos o confirmam. Assim, o teste passa por avaliar a evolução da taxa de cobertura total em função do número de nós da rede, considerando várias percentagens de nós atacantes (5%, 10% e 20%) e com 60% de nós emissores.

Como se pode observar na figura 6.6, o gráfico que diz respeito ao Clean-Slate demonstra que a sua cobertura não é afectada pelo ataque, a evolução da taxa de cobertura, ainda que com pequenas irregularidades mantém-se próxima da obtida sem ataques. Quanto ao INSENS, verifica-se um decréscimo bastante elevado relativamente à cobertura total sem ataques, passando a obter-se valores abaixo dos 6% a partir dos 500 nós, e que tendem facilmente a aproximar-se do valor nulo, esta situação agrava-se quanto maior for a percentagem de nós atacantes.

Explicando os resultados obtidos, o Clean-Slate não é afectado devido ao protocolo de descoberta de vizinhos que verifica a bidireccionalidade, enquanto no INSENS tal

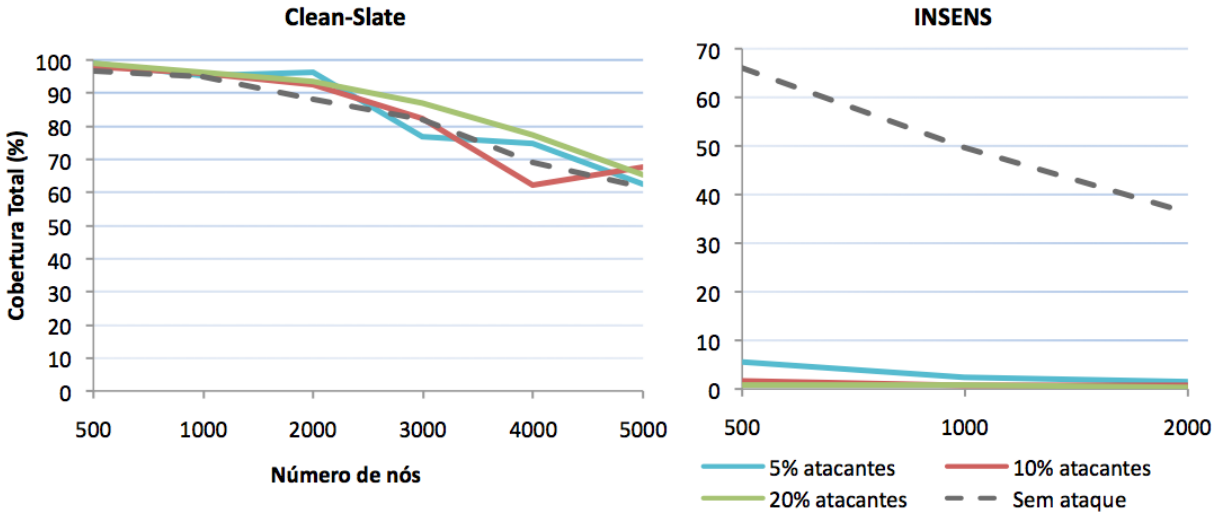


Figura 6.6: Impacto do *hello-flooding* na cobertura.

não acontece, e como a capacidade de transmissão dos nós legítimos é inferior à dos atacantes provocam-se imediatamente partições na rede que prejudicam gravemente a fase de mensagens *feedback*. Portanto, o ataque *hello-flooding* tem um impacto muito significativo quando aplicado ao INSENS, contrariamente ao Clean-Slate.

Fiabilidade

O ataque *hello-flooding*, sendo um ataque à selecção de rotas, talvez possa causar impacto ao nível da fiabilidade, na medida em que os caminhos seleccionados possivelmente são mais problemáticos para garantir a entrega de mensagens na estação-base. A análise de fiabilidade pode ser efectuada sobre o Clean-Slate, uma vez que a sua cobertura não sofre alterações com este ataque. Por contrário, no INSENS constatou-se uma forte redução da taxa de cobertura, o que coloca a rede praticamente inoperacional e retira assim o sentido de se realizar um teste à fiabilidade.

O teste à taxa de fiabilidade, aplicado ao Clean-Slate, foi efectuado para percentagens de nós atacantes de 5%, 10%, 20% e 30% entre 500 e 5000 nós e com congestionamento moderado. O resultado obtido assemelha-se ao respectivo gráfico da taxa de fiabilidade sem ataques, observável na figura 6.3. Portanto, concluiu-se que a taxa de fiabilidade do Clean-Slate não é afectada pelo ataque *hello-flooding*. O mecanismo seguro de descoberta de vizinhos é responsável por proteger o ataque e consequentemente por manter os mesmos valores de fiabilidade.

Latência e Energia

Na latência, o facto de o *hello-flooding* ser um ataque à selecção de rotas poderá implicar algum tipo de impacto a este nível, como tal efectuou-se um teste ao Clean-Slate, em condições idênticas ao teste de fiabilidade anterior, mas agora verificando se a latência se mantém semelhante à obtida sem ataques. Os resultados obtidos evidenciaram que a latência mantém-se semelhante, concluindo-se que não é afectada pelo ataque.

Quanto à energia, sabe-se que o número de saltos da mensagem tem influência nos resultados energéticos, e como este ataque pretende ter efeitos nos caminhos seleccionados, então interessa testar qual o impacto energético provocado. O teste foi realizado ao Clean-Slate nas mesmas condições do teste anterior. Por análise de resultados verificou-se que os valores são novamente semelhantes aos obtidos sem ataques, concluindo-se que independentemente do número de nós da rede e de atacantes o *hello-flooding* não afecta o consumo energético do encaminhamento.

6.6.2 *Sinkhole*

Nos próximos testes relativos a este ataque considera-se que os nós atacantes no protocolo INSENS enviam três mensagens *request* (o valor predefinido para o ataque), ao contrário dos nós legítimos que apenas enviam uma. Convém relembrar que este ataque não se aplica à actual implementação do Clean-Slate, como referido em 5.4.

Cobertura

Uma vez que um ataque de *sinkhole* leva os nós atacantes a persuadirem outros nós de modo a que os caminhos sejam formados convenientemente, interessa assim analisar até que ponto a cobertura pode sofrer ou não consequências. Portanto, este teste visa analisar o impacto que *sinkhole* provoca na cobertura total do INSENS em redes de 500 a 6000 nós, tendo percentagens de nós atacantes de 5%, 10%, 20%, 30% e 50%, e uma percentagem de emissores de 60%.

Como é ilustrado no gráfico da figura 6.7, na presença deste ataque a taxa de cobertura total não é substancialmente afectada, por comparação com a cobertura sem ataques. Os desvios verificados são relativamente baixos, no entanto, entre os 500 e os 3000 nós é onde ocorrem desvios mais notáveis, sobretudo quanto maior o número de atacantes. Isto explica-se pelo facto de quanto maior a percentagem de nós atacantes maior o número de mensagens *request* extra, as quais tendem assim a provocar mais

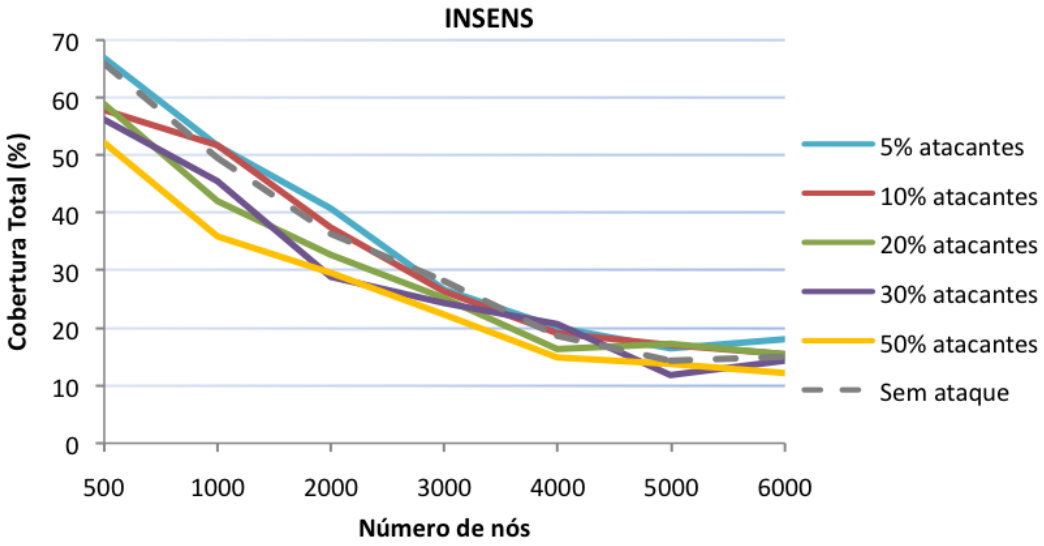


Figura 6.7: Impacto do *sinkhole* na cobertura.

colisões e a prejudicar a descoberta de vizinhos. Repare-se que com o natural decréscimo da cobertura esses desvios tendem a ser menos notáveis. Assim, conclui-se que o ataque *sinkhole* não revela um impacto considerável na cobertura do INSENS, sobretudo quanto maior o número de nós da rede.

Fiabilidade

Por agora interessa analisar qual o impacto do *sinkhole* relativamente à fiabilidade, de modo a avaliar se o facto de o ataque tentar influenciar a selecção dos caminhos prejudica de alguma forma a entrega de mensagens na estação-base. Para tal, o teste consistiu em analisar a taxa de fiabilidade do INSENS em redes de 500 a 6000 nós, tendo uma percentagem de nós atacantes de 5%, 10%, 20%, 30% e 50%, e com congestionamento moderado. As amostragens realizadas permitiram verificar que a taxa de fiabilidade esperada para qualquer percentagem de nós atacantes evolui de modo idêntico à taxa de fiabilidade quando não existem ataques (apresentada no respectivo gráfico da figura 6.3). Este resultado deve-se muito ao forte contributo oferecido pela estação-base, sendo ela que calcula os caminhos por intermédio da informação obtida, evitando que estes sejam completamente escolhidos por influência dos atacantes.

Latência

No que diz respeito à latência, com o objectivo de verificar se o *sinkhole*, ao tentar influenciar os caminhos seleccionados para encaminhar dados, provoca alterações a

este nível, foi realizado um teste ao INSENS em condições idênticas às do teste anterior de fiabilidade.

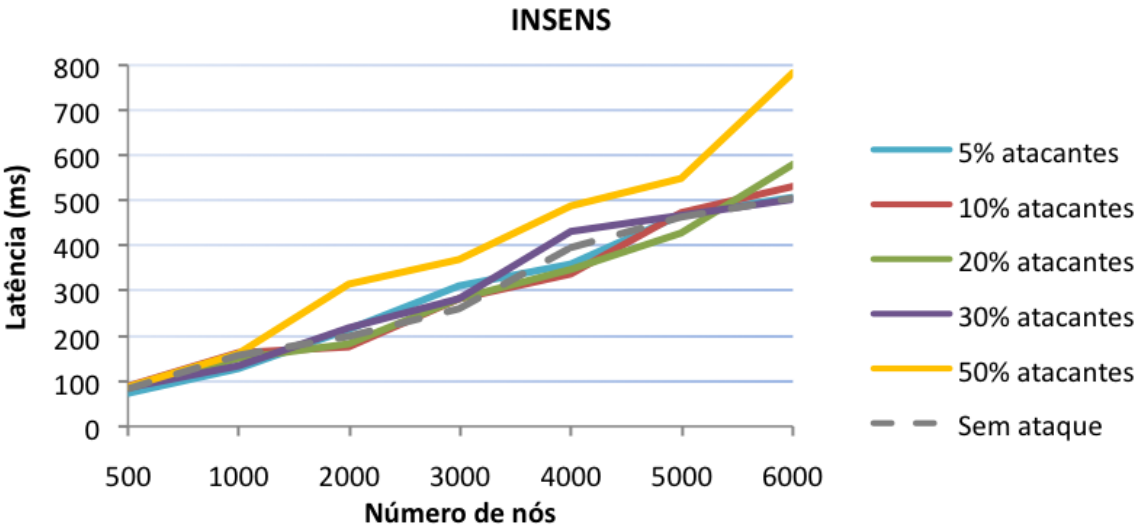


Figura 6.8: Impacto do *sinkhole* na latência.

Observa-se em 6.8 que para percentagens de atacantes de 5%, 10%, 20% e 30% os valores de latência são semelhantes aos obtidos sem ataque, contudo para 50% denotou-se um aumento significativo, chegando a aumentar em cerca de 280 ms numa rede de 6000 nós. Este resultado indica que a latência apenas é afectada significativamente quando estamos perante uma percentagem de atacantes bastante alta (a partir de 50%).

Uma elevada percentagem de atacantes a executarem este ataque aumenta consideravelmente o número de mensagens *request* na rede, propiciando muitas colisões que dificultam o processo de criação de vizinhanças. Assim, as vizinhanças são mais restritas, inibindo de certo modo a estação-base de construir melhores caminhos que beneficiariam a latência. Para percentagens de atacantes inferiores a centralização do cálculo dos caminhos na estação-base é suficiente para contornar o ataque. Naturalmente uma percentagem de 50% de atacantes é improvável de ocorrer.

Energia

O teste de energia ao *sinkhole* ocorre igualmente aos anteriores, no entanto agora com o objectivo de verificar se a existência de vários atacantes a realizarem este ataque tem algum impacto no consumo energético. Portanto, o teste é novamente efectuado ao INSENS numa rede com congestionamento moderado, com entre 500 a 6000 nós e com as seguintes percentagens de atacantes: 5%, 10%, 20%, 30%, 40% e 50%.

Por análise dos valores médios obtidos, através das amostras efectuadas, verificou-se que esses valores médios são semelhantes aos obtidos sem qualquer ataque, o que leva a concluir que o *sinkhole* não provoca alterações ao consumo energético durante a fase de encaminhamento.

6.6.3 *Sybil*

O ataque *sybil* é por natureza um ataque muito problemático, no entanto foi considerada uma visão mais restrita deste ataque, onde os atacantes actuam por intrusão apenas com os segredos capturados no próprio nó, ou seja, não fazem uso de segredos criptográficos presentes noutros dispositivos capturados. Em concreto, os atacantes enviam três identidades falsas (valor predefinido) além da sua identidade verdadeira, ao contrário dos nós legítimos que enviam apenas a sua identidade verdadeira.

Cobertura

Nos pressupostos anteriormente referidos, prevê-se que neste ataque *sybil* a cobertura não sofra alterações pelas razões a seguir apresentadas, no entanto para que tal seja comprovado foi realizado um teste à cobertura dos protocolos Clean-Slate e INSENS que envolveu uma rede com 500 a 6000 nós, 50% de nós atacantes e 60% de nós emissores. Ao utilizar uma percentagem de atacantes elevada foi imediato verificar se ocorre ou não algum impacto.

No protocolo Clean-Slate, assumindo-se que a autoridade de rede é uma entidade completamente confiável, esta distribui a identidade e a correspondente assinatura por cada nó. Logo, se o atacante envia a sua identidade correcta e outras identidades falsas, todas as falsas são rejeitadas porque não correspondem à assinatura que essa autoridade forneceu. O mecanismo seguro de descoberta de vizinhos permite verificar imediatamente as identidades falsas, rejeitando-as. No caso do INSENS, a estação-base mantém uma correspondência entre a chave de cada nó e a sua identidade, portanto quando a estação-base recebe mensagens de *feedback* procede logo à respectiva verificação, rejeitando identidades falsas.

Deste modo, assumindo que o nó atacante envia sempre a identidade correcta, além das falsificadas, espera-se que este comportamento incorrecto por parte dos nós atacantes não provoque qualquer tipo de efeito. De facto, os resultados obtidos no teste efectuado comprovaram o previsto, mantendo-se os valores de cobertura semelhantes aos obtidos sem ataque.

Fiabilidade, Latência e Energia

Relativamente à fiabilidade, latência e energia é novamente previsto que o ataque não produza qualquer efeito, pelas mesmas razões explicadas no teste de cobertura anterior. Assim, realizaram-se testes nas mesmas condições do teste anterior com o objectivo de comprovar que este ataque *sybil* também não provoca efeitos em relação a estes critérios de medição, o que se confirmou, pois obtiveram-se experimentalmente para ambos os protocolos valores da mesma ordem que os obtidos sem ataques.

6.6.4 *Wormhole-simples*

Cobertura

Um ataque *wormhole-simples* cria ligações entre nós que eventualmente podem estar muito distantes, o que não seria possível através do modelo de rádio utilizado pelos nós legítimos da rede. Assim, essas ligações possivelmente terão efeitos na cobertura dos protocolos Clean-Slate e INSENS, sendo bastante interessante avaliá-los. O teste realizado permite analisar a evolução da cobertura total em função do número de nós e de várias percentagens de atacantes (5%, 10%, 20%, 30% e 50%), existindo 60% de emissores. No INSENS o teste é realizado para um intervalo de 500 a 4000 nós, pois a injecção deste ataque torna a execução muito demorada a partir dos 4000 nós, e pelas mesmas razões o teste aplicou-se ao Clean-Slate no intervalo de 500 a 3000 nós.

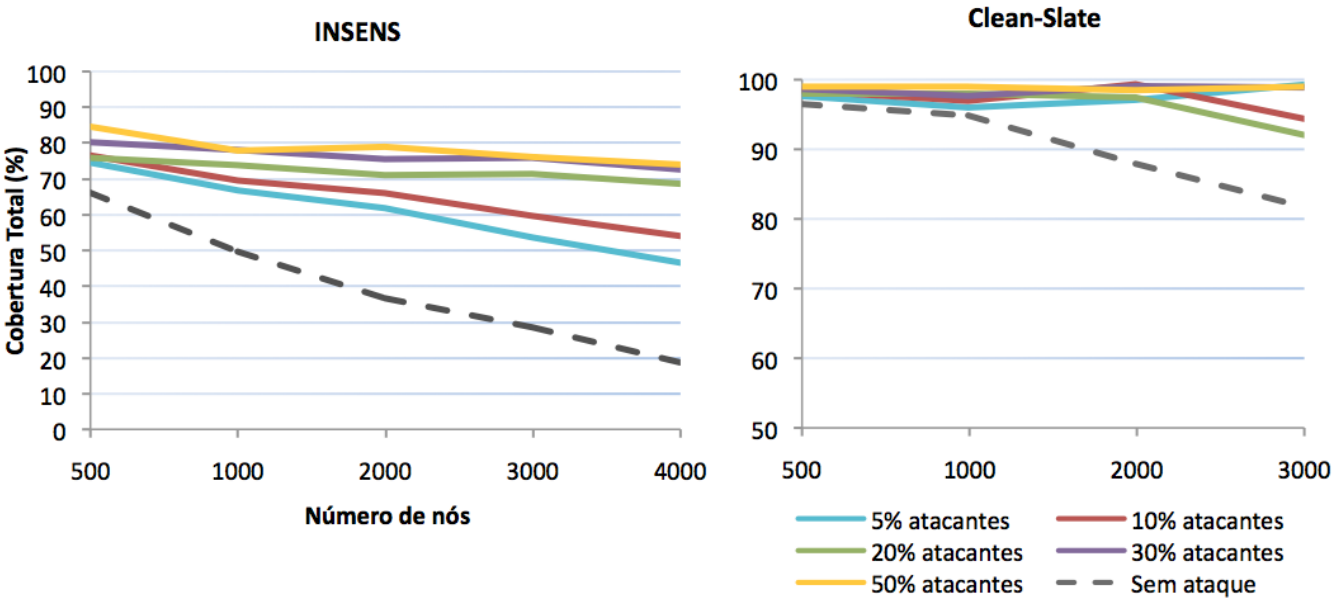


Figura 6.9: Impacto do *wormhole-simples* na cobertura.

Os gráficos da figura 6.9 revelam que ataques *wormhole-simples* aumentam substancialmente a cobertura total de ambos os protocolos. No caso do INSENS o aumento é mais notável e verifica-se claramente que quanto maior a percentagem de atacantes a formarem *wormholes* maior é a taxa de cobertura obtida, no entanto esta tem tendência a diminuir ligeiramente com o aumento do número de nós da rede e nunca atinge valores superiores a 85%. Já no Clean-Slate observa-se que qualquer uma das percentagens de atacantes testadas produz também um aumento considerável, agora o protocolo obtém uma cobertura total entre os 90% e os 100% até aos 3000 nós. Com este resultado constata-se que *wormholes* deste tipo melhoram significativamente a cobertura de ambos os protocolos, o que se deve à criação de ligações de longa distância onde não ocorrem perdas e que contribuem para que qualquer zona da rede possa estar coberta.

Fiabilidade

No teste anterior verificaram-se melhoramentos ao nível da cobertura, que possivelmente reflectir-se-ão também ao nível da fiabilidade. Para verificar essa possibilidade pretende-se agora testar se os ataques *wormhole-simples* provocam algum tipo de alteração na fiabilidade comparativamente à obtida sem qualquer tipo de ataque. Este teste consiste em avaliar a taxa de fiabilidade nos protocolos Clean-Slate e INSENS em redes de 500 a 3000 ou 4000 nós, com congestionamento moderado e para várias percentagens de nós atacantes: 5%, 10%, 20%, 30% e 50%.

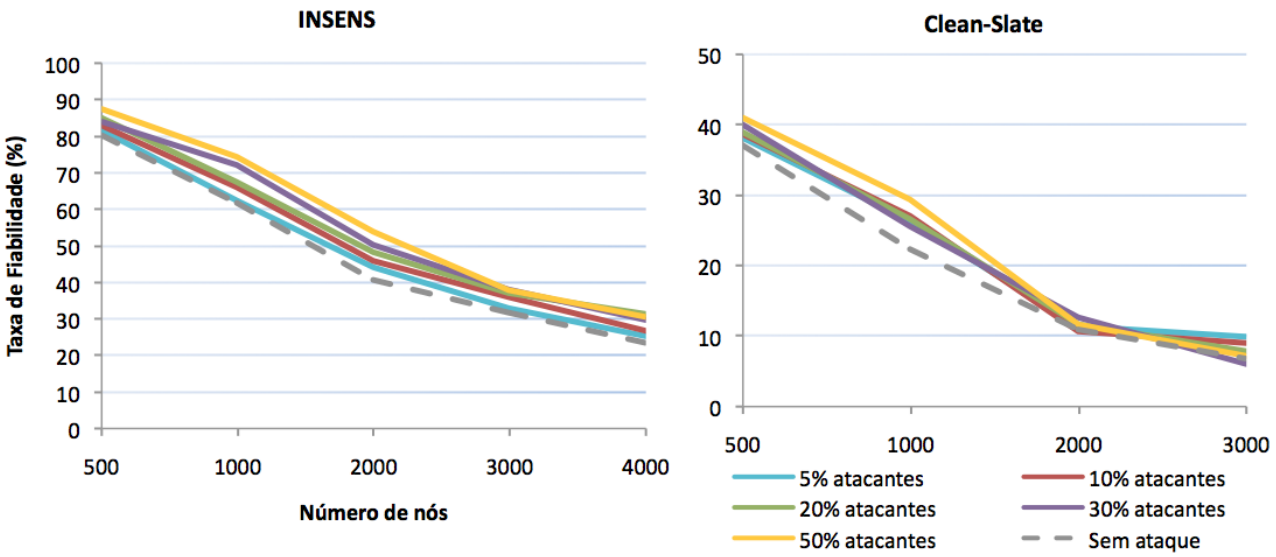


Figura 6.10: Impacto do *wormhole-simples* na fiabilidade.

Os resultados obtidos, ilustrados em 6.10, indicam que no caso do INSENS a taxa de fiabilidade é melhorada, sobretudo quanto maior a percentagem de atacantes. Por

exemplo, no caso de existir uma percentagem de atacantes de 50%, a fiabilidade melhora cerca de 5 a 15%, enquanto que com 5% de atacantes é bastante próxima da obtida sem ataques. O melhoramento verificado acontece devido a diversos factores: mensagens transmitidas em *wormholes* não se perdem (é uma ligação com melhores recursos); sabendo-se que a estação-base se baseia nos caminhos mais curtos e que o *wormhole* propicia ligações entre nós distantes, então o número de saltos da mensagem pode diminuir; e como a cobertura melhora, facilita-se a dispersão do tráfego desde os emissores até à estação-base reduzindo o número de colisões.

Relativamente ao Clean-Slate, verifica-se que a taxa de fiabilidade tem alguns melhoramentos, contudo não segue um padrão regular porque, por vezes, obtêm-se valores semelhantes aos obtidos sem ataque. Repare-se, também, que não existem diferenças conclusivas com o aumento da percentagem de atacantes. O melhoramento observado deve-se ao facto dos *wormholes* não permitirem perdas de mensagens e criarem ligações de longa distância. No entanto, as irregularidades são resultado do agrupamento recursivo que não tem qualquer noção de menor caminho, levando possivelmente a mensagem a efectuar na mesma muitos saltos e a manter a probabilidade de ocorrerem colisões.

Assim, conclui-se que este ataque melhora, de facto, a fiabilidade do INSENS, enquanto no Clean-Slate é um pouco melhorada mas evidencia algumas irregularidades.

Latência

Tendo conhecimento dos melhoramentos verificados nos testes anteriores, interessa-nos agora analisar experimentalmente se o *wormhole-simples* também permite melhorar a rede em termos de latência. Para tal realizou-se um teste ao Clean-Slate e ao INSENS nas mesmas condições do teste anterior de fiabilidade, no entanto agora destinado aos valores de latência.

Durante a experimentação denotaram-se oscilações nos valores de latência, por conseguinte os resultados assentam assim na média de valores. Os gráficos em 6.11 ilustram que para o Clean-Slate a latência é melhorada, tendendo a melhorar mais quando a percentagem de atacantes aumenta, repare-se que com 50% de atacantes para 2000 nós melhorou em cerca de 250 ms. Quanto ao INSENS, este vê a sua latência melhorada até cerca de 2000 nós, a partir daí curiosamente obtêm-se valores superiores aos obtidos sem ataque.

Os melhoramentos observados no Clean-Slate devem-se ao facto dos *wormholes* serem ligações que incutem uma latência muito reduzida (considerada nula em termos de implementação). Relativamente aos resultados do INSENS, o melhoramento até aos

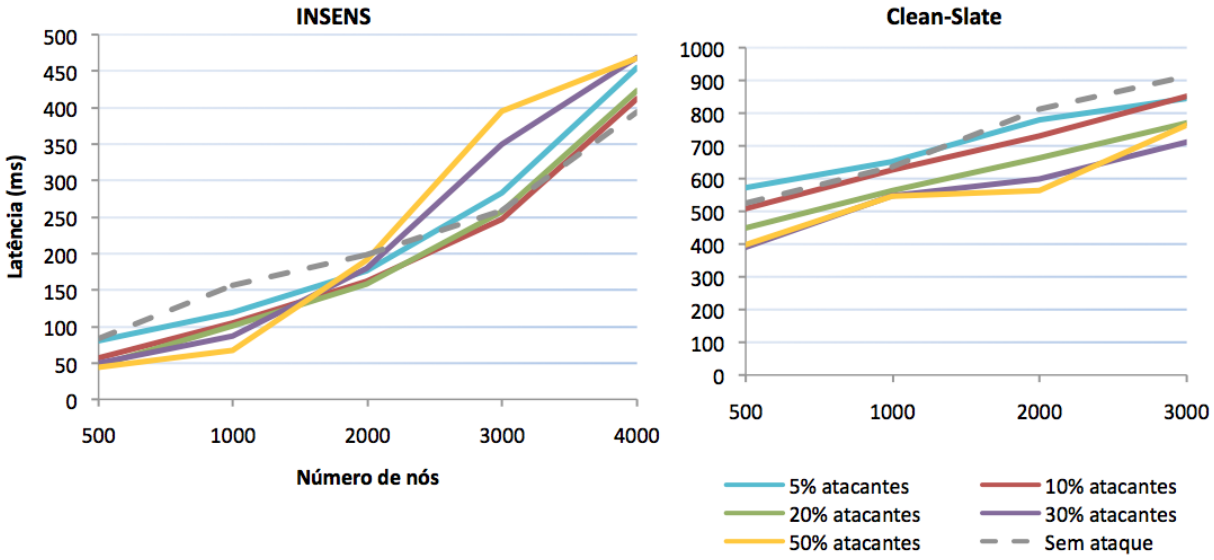


Figura 6.11: Impacto do *wormhole-simples* na latência.

2000 nós explica-se pela mesma razão e ainda pelo facto de a estação-base conseguir agora criar caminhos mais curtos que os criados sem o ataque. Porém, o aumento da latência a partir dos 2000 nós ocorre por um detalhe bastante particular, existindo uma rede com muitos *wormholes*, no INSENS o tráfego de qualquer ponto da rede tende a circular muito rapidamente até próximo da estação-base, sendo aí que se gera uma forte concentração de mensagens que se destinam à estação-base, obrigando os nós a aguardarem algum tempo até encontrarem o meio disponível para iniciarem a emissão.

Energia

Relativamente à energia, o facto do ataque *wormhole-simples* ter influência na cobertura e fiabilidade, como verificado anteriormente, pode levar a prever possíveis implicações no consumo energético durante o encaminhamento. Nesse sentido este teste pretende dar uma resposta, oferecendo uma visão comparativa do consumo energético dos protocolos quando estamos na presença do ataque. O teste é realizado aos protocolos Clean-Slate e INSENS em condições idênticas às dos testes anteriores.

Analisando os resultados da figura 6.12, observa-se que o Clean-Slate não sofre qualquer tipo de impacto energético, pois os valores mantêm-se algo semelhantes aos obtidos sem ataque. No que diz respeito aos resultados do INSENS, observa-se que o consumo energético diminui, tanto quanto maior for a percentagem de atacantes. Repare-se que no caso de uma percentagem de 5% de atacantes os valores são próximos dos valores obtidos sem ataque, no entanto com 50% de atacantes o consumo reduz

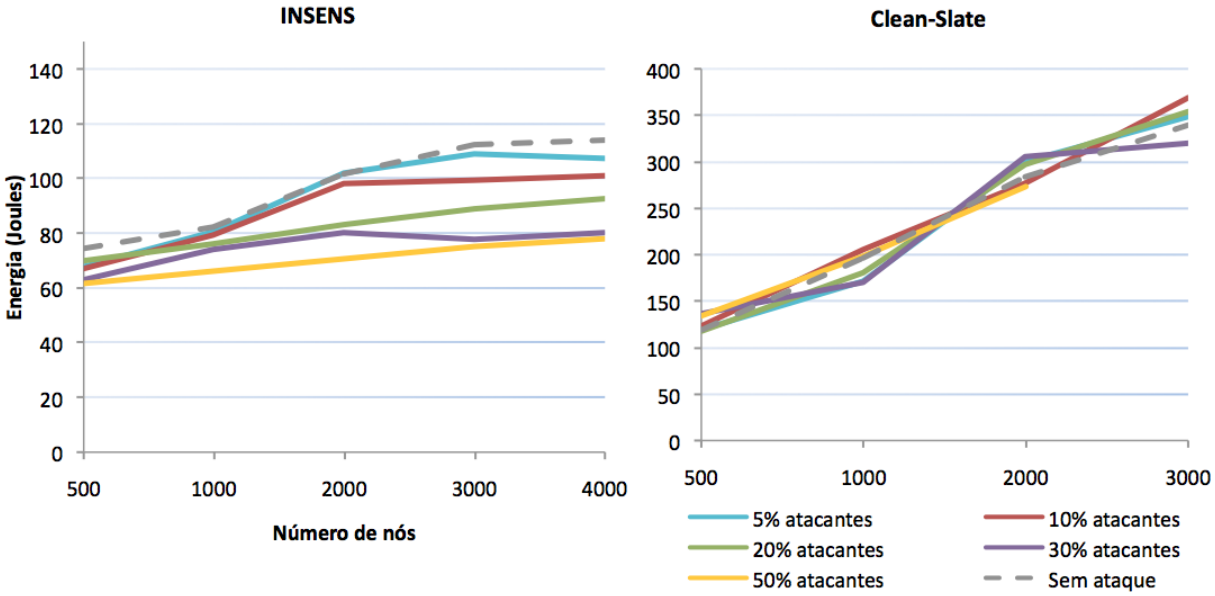


Figura 6.12: Impacto do *wormhole-simples* na energia.

entre cerca de 10 a 40 joules.

A diferença de resultados é explicada, essencialmente, pelo facto de no INSENS estes *wormholes* reduzirem efectivamente o número de saltos necessário para encaminhar a mensagem até à estação-base, diminuindo então o consumo. Ao invés, no Clean-Slate o número de saltos não sofre alterações consideráveis que permitam um decréscimo do consumo, e o facto destas ligações exigirem menor consumo energético também mostrou não ser suficiente para que ocorram alterações.

6.6.5 *Wormhole-overlay*

Cobertura

O ataque *wormhole-overlay*, tendo por base o *wormhole-simples* e ao permitir um maior número de ligações entre os nós atacantes que formam assim uma *overlay*, leva a prever que em termos de cobertura os resultados nunca sejam inferiores aos obtidos na figura 6.9, inclusive espera-se que sejam superiores. De facto, por obtenção de algumas amostras em condições idênticas às do teste de cobertura do ataque *wormhole-simples* confirmou-se que os valores agora obtidos são superiores aos do *wormhole-simples*. Isto verificou-se principalmente no INSENS, onde de 500 a 4000 nós se obtiveram valores que se encontram sempre entre os 75% e os 95% com as mesmas percentagens de atacantes do gráfico em 6.9. Quanto ao Clean-Slate os valores para o *wormhole-simples* já eram bastante positivos, por conseguinte os melhoramentos do *wormhole-overlay* não são tão notáveis.

Fiabilidade

Tal como foi analisado anteriormente, a fiabilidade com o *wormhole-simples* é em geral melhorada, deste modo pretende-se agora averiguar se é alterada quando se passa do *wormhole-simples* para um *wormhole-overlay*, pois o facto de passar a existir uma *overlay* de atacantes torna pertinente efectuar esta análise. O teste realizou-se nas mesmas condições do teste de fiabilidade para o *wormhole-simples*.

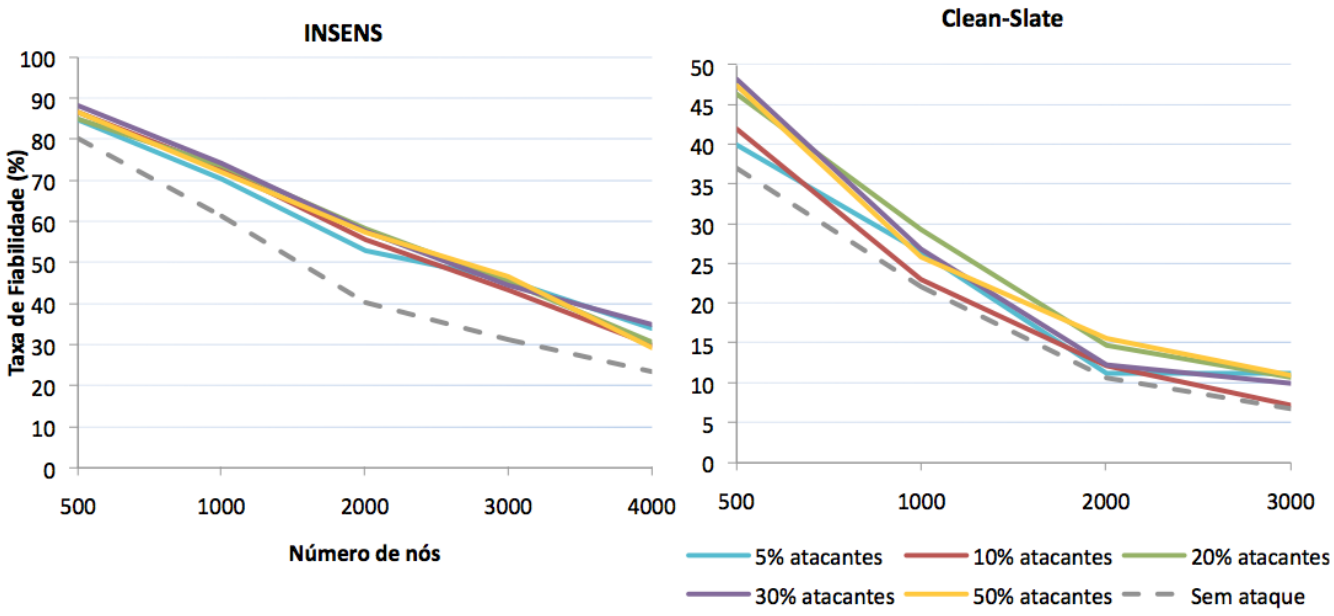


Figura 6.13: Impacto do *wormhole-overlay* na fiabilidade.

Por análise dos resultados, na figura 6.13 observa-se que no caso do INSENS a fiabilidade é, em geral, ligeiramente superior à obtida em 6.10. Contudo, um aspecto que difere do *wormhole-simples* é a taxa de fiabilidade não aumentar em função da percentagem de atacantes para o mesmo número de nós da rede. Como se pode observar, para 5% de atacantes a fiabilidade é próxima da obtida com 50% de atacantes. Isto acontece porque uma *overlay* formada por 5% de nós atacantes já é suficiente para construir caminhos de tamanho tão reduzido quanto o obtido por uma *overlay* constituída por 50% de atacantes. Já a pequena subida de valores ocorre essencialmente porque a *overlay* permite criar caminhos ainda mais curtos que os obtidos com o ataque *wormhole-simples*, reduzindo o número de saltos da mensagem, e ainda por existir um maior número de *wormholes*, onde não ocorrem perdas de mensagens.

Relativamente ao Clean-Slate, este apresenta taxas de fiabilidade superiores às obtidas sem ataque, tal como em certo modo acontecia no *wormhole-simples*, conseguindo-se melhorar até quase cerca de 15%. Em comparação com o *wormhole-simples* os dados

denotam algumas irregularidades que não permitem concluir que ocorra de facto um melhoramento em relação a este. Observa-se também que o aumento da percentagem de atacantes não expressa claramente o crescimento da taxa de fiabilidade. Estes resultados devem-se principalmente às razões já mencionadas para o *wormhole-simples*, a existência da *overlay* acrescenta um maior número de ligações fiáveis, no entanto é o agrupamento recursivo que cria os caminhos, sendo que este não dá prioridade a essas ligações nem se baseia em caminhos mais curtos.

Latência

No que diz respeito à latência, importa analisar como esta evolui nos protocolos Clean-Slate e INSENS quando se está perante uma *overlay* formada por vários *wormholes*. Nesse sentido realizou-se um teste que mantém as mesmas condições do teste de latência ao *wormhole-simples*, mudando apenas o tipo de ataque.

Durante a experimentação as várias amostras efectuadas evidenciaram uma acentuada irregularidade de valores, o que não possibilitou a construção dos respectivos gráficos rigorosamente, mas no entanto os intervalos de valores obtidos revelaram-se conclusivos. As amostras revelaram que para o Clean-Slate até aos 1000 nós e em qualquer percentagem de atacantes a latência é melhorada, mas a partir desse número tende a piorar um pouco de forma irregular. O INSENS, por sua vez, obteve valores muito piores que os obtidos sem ataques, a latência aumenta bastante quanto maior o número de nós da rede, por exemplo, aos 4000 nós foram obtidos valores que se aproximam dos três segundos.

O Clean-Slate melhora até aos 1000 nós porque as mensagens quando passam nos *wormholes* da *overlay*, embora sejam instantaneamente enviadas, constituem tráfego que ainda circula pela rede em condições adequadas. Contudo, quando a *overlay* tem dimensões maiores o tráfego de mensagens rapidamente atinge a zona onde a estação-base se encontra, constituindo um ponto de concentração que provoca períodos de espera que aumentam a latência, à semelhança do que acontecia com o INSENS no *wormhole-simples*. Relativamente ao INSENS a mesma justificação é aplicada, pois se tal situação já ocorria no *wormhole-simples* agora ainda é mais agravada.

Energia

Uma vez que estamos perante uma *overlay* de *wormholes* e tendo em conta os resultados já obtidos para o *wormhole-simples* em 6.12 será interessante perceber se esta *overlay* vem acrescentar algo mais que provoque consideráveis alterações ao consumo

energético. Assim, este teste foi realizado novamente nas mesmas condições do teste de energia que diz respeito ao ataque *wormhole-simples*.

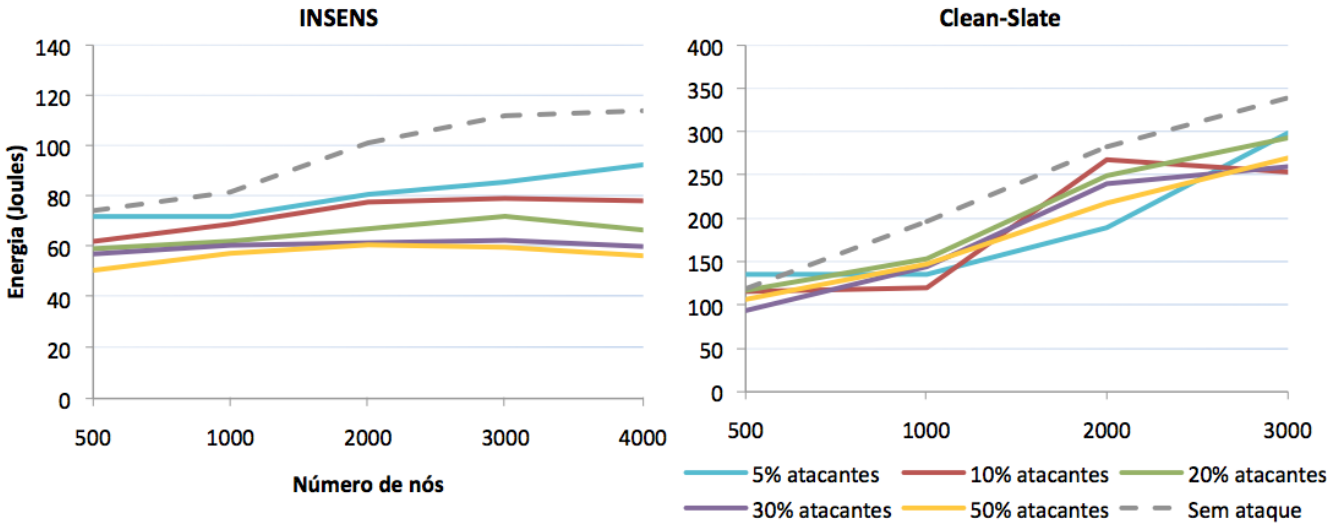


Figura 6.14: Impacto do *wormhole-overlay* na energia.

Por observação da figura 6.14 facilmente se visualiza que em ambos os protocolos o consumo energético diminuiu. Em relação ao INSENS, a energia despendida baixa tanto mais quanto maior a percentagem de atacantes, e um aspecto notável é que o consumo é ainda menor que o obtido para o *wormhole-simples*. Podemos observar que agora mesmo para 5% de atacantes o consumo passa a diminuir consideravelmente, e com uma percentagem de 50% de atacantes o consumo chega a reduzir a valores inferiores aos 60 joules. Quanto ao Clean-Slate, enquanto no *wormhole-simples* não se observava nenhum impacto, agora passa a evidenciar-se um razoável decréscimo, sem que exista uma relação entre a percentagem de atacantes e o decréscimo.

O melhoramento observado no INSENS explica-se pelo facto de a *overlay* permitir reduzir ainda mais que no caso do *wormhole-simples* o número de saltos que a mensagem efectua. O Clean-Slate passa agora a evidenciar melhoramentos principalmente devido a uma ligeira redução do número de saltos, mas também porque os *wormholes* são ligações que requerem um baixo consumo energético.

6.6.6 *Wormhole-mitm (man-in-the-middle)*

Cobertura

Uma vez que este tipo de *wormhole* cria falsas vizinhanças entre nós e os próprios nós atacantes não participam no protocolo, leva-nos a prever que a cobertura seja logo

à partida afectada. Assim, este teste permite avaliar qual o protocolo que sofre um maior impacto na cobertura total, sendo que no caso do INSENS é efectuado para o intervalo de 500 a 4000 nós, enquanto no Clean-Slate entre os 500 e os 3000 nós, isto porque com a injecção deste ataque a obtenção de amostras com um maior número de nós requer execuções bastante demoradas. Utilizam-se ainda 5%, 10%, 20%, 30% e 50% de nós atacantes, e 60% dos nós desempenham o papel de emissores.

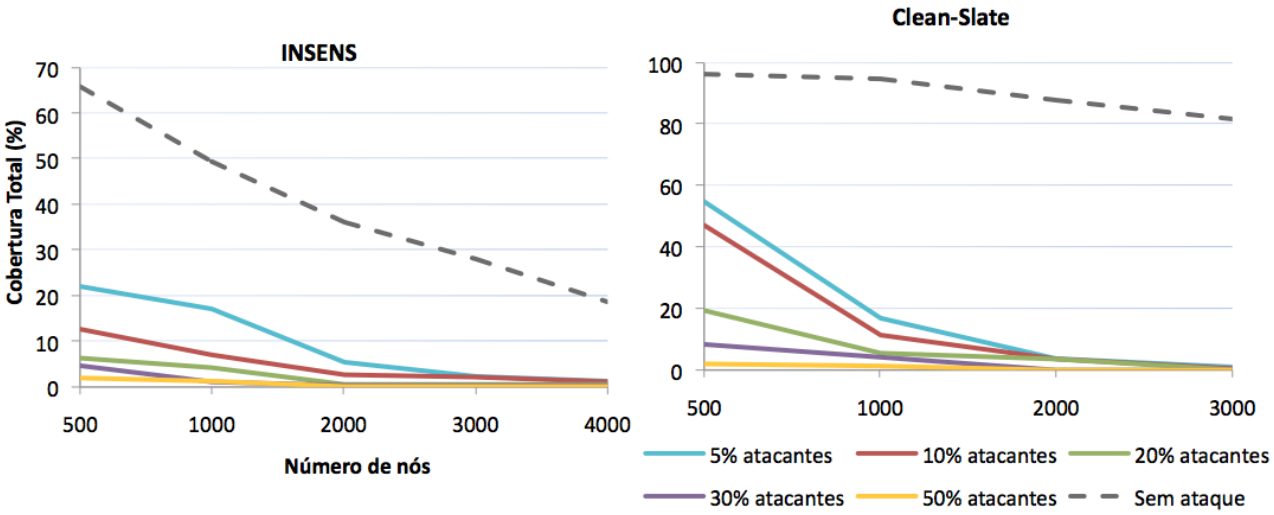


Figura 6.15: Impacto do *wormhole-mitm* na cobertura.

Através dos resultados obtidos, ilustrados na figura 6.15, verifica-se que tanto o INSENS como o Clean-Slate são bastante afectados pelo ataque, sobretudo quanto maior a percentagem de atacantes e o número de nós da rede, atingindo-se facilmente o valor nulo. Contudo, o INSENS demonstra ser o mais afectado mesmo para um baixo número de nós, pois como se pode observar obtêm-se valores de cobertura muito reduzidos para 500 nós e com uma baixa percentagem de atacantes, enquanto que nas mesmas condições o Clean-Slate permite obter valores relativamente superiores.

Conclui-se então que as falsas vizinhanças criadas por este tipo de *wormhole* prejudicam fortemente a cobertura dos protocolos, e refira-se que um aspecto a ter em conta nos resultados é o facto de a percentagem de atacantes estar directamente relacionada com o decréscimo da cobertura, isto porque os atacantes não participando na fase de configuração do protocolo implica logo à partida que nunca possam ser emissores totalmente cobertos.

Fiabilidade, Latência e Energia

Os danos provocados pelo ataque ao nível da cobertura reduzem muito a percentagem de emissores disponíveis, impedindo assim de se efectuar uma avaliação correcta acerca da fiabilidade, latência e consumo energético durante o encaminhamento.

6.6.7 *Blackhole*

Cobertura

Um ataque de *blackhole* é bastante interessante de analisar em termos de cobertura, pois o descarte de mensagens por parte dos nós atacantes pressupõe implicações graves caso o protocolo não possua um mecanismo adequado que solucione o problema. Nesse sentido este teste tem como objectivo avaliar a evolução da taxa de cobertura total nos protocolos Clean-Slate e INSENS quando submetidos ao ataque para determinadas percentagens de atacantes (5%, 10%, 20%, 30% e 50%), numa rede entre 500 a 6000 nós e 60% de nós emissores.

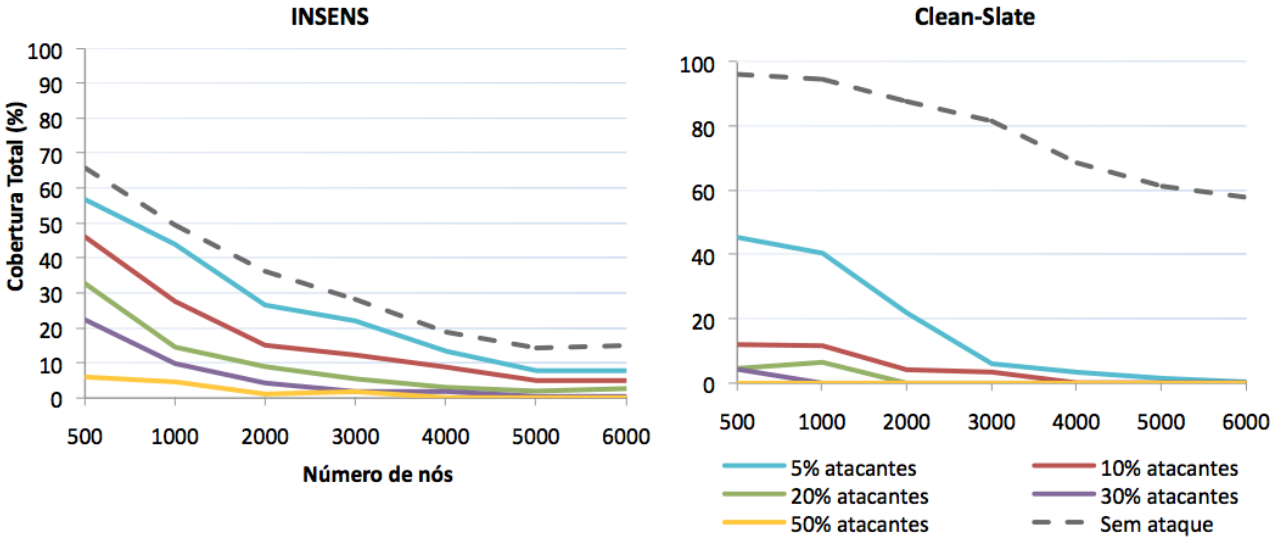


Figura 6.16: Impacto do *blackhole* na cobertura.

Como se pode observar na figura 6.16, o INSENS com 5% de nós atacantes apresenta valores próximos dos obtidos sem ataque e, tal como se esperaria, o aumento dessa percentagem reduz a taxa de cobertura. Porém, os resultados demonstram que o protocolo contorna relativamente bem o ataque quando a percentagem de atacantes não é muito elevada. Já no que diz respeito ao Clean-Slate, o mesmo não acontece, pois para 5% de nós atacantes o decréscimo da taxa de cobertura total é muito acentuado,

tal que aos 3000 nós já está próxima do valor nulo.

Os resultados explicam-se pelo facto do INSENS enviar cada mensagem por múltiplos caminhos (dois caminhos em simultâneo), o que permite o protocolo adaptar-se bem a ataques *blackhole*. Contrariamente, o Clean-Slate embora também contenha múltiplos caminhos não encaminha a mensagem por eles simultaneamente, determina apenas um caminho de forma aleatória, o que se revela insuficiente para proteger o ataque.

Fiabilidade, Latência e Energia

O *blackhole* sendo um ataque de descarte de mensagens que provoca alterações na cobertura, como se concluiu no teste anterior, torna-se problemático relativamente às medições de fiabilidade, latência e consumo energético durante a fase de encaminhamento. A taxa de cobertura diminui com o aumento da percentagem dos nós atacantes e estes critérios de medição estão directamente dependentes da cobertura, no sentido em que necessitam de nós que sejam totalmente cobertos para desempenharem o papel de emissores. Então, a cobertura ao ser afectada acaba por fazer com que não existam nós cobertos suficientes ou que a maioria dos nós com condições de cobertura total se encontre próxima da estação-base, o que levará a que os resultados finais sejam desinteressantes e inconclusivos. No entanto, os *blackholes* são ainda testados relativamente à eficiência do encaminhamento redundante, apresentado mais à frente.

6.6.8 *Selective-Forwarding*

O ataque *selective-forwarding* é uma particularidade do ataque anterior (*blackhole*), no sentido em que apenas uma certa parte das mensagens é descartada. Como tal, os resultados de cobertura assemelham-se à cobertura do *blackhole* quando o parâmetro de probabilidade de não descarte de mensagens é 0%, em oposição o parâmetro a 100% tende a que os valores de cobertura sejam semelhantes aos obtidos sem qualquer ataque. Realizaram-se amostragens utilizando 30% de probabilidade das mensagens não serem descartadas, e os resultados evidenciaram um melhoramento da cobertura em relação à obtida para o *blackhole* da figura 6.16.

6.7 Outras características

Os protocolos quando sujeitos a ataques envolvem ainda outras características que são inerentes ao próprio encaminhamento, as quais foram ainda alvo de estudo, nomeadamente no que diz respeito à capacidade do protocolo realizar multi-encaminhamento,

e a eficácia de um ataque à selecção de rotas em desviar o tráfego para o respectivo nó atacante.

Multi-encaminhamento

Relativamente ao multi-encaminhamento, o seguinte teste tem como objectivo analisar se o protocolo é capaz de realizar de forma eficaz o multi-encaminhamento de mensagens face aos ataques *wormhole-overlay* e *blackhole*, de modo a garantir a sua entrega por algum caminho legítimo. O teste é efectuado ao Clean-Slate e INSENS, tendo as seguintes condições: rede com 500 nós, 10% de atacantes, *delay* de 2 segundos, 5% dos nós são emissores, sendo que cada um envia 10 mensagens, e as dimensões são 1000m x 1000m (Clean-Slate) e 650m x 650m (INSENS) .

Os resultados confirmaram o esperado para o Clean-Slate, pois o encaminhamento não é feito de forma redundante por cada mensagem enviada, logo para o caso do *wormhole-overlay* a mensagem segue unicamente por um caminho legítimo ou por um caminho infectado, no caso do *blackhole* a mensagem ou é recebida na estação-base ou é descartada em algum nó *blackhole*. Relativamente ao INSENS, o encaminhamento redundante (são utilizados dois caminhos) é cumprido e assim para o *wormhole-overlay* os resultados indicaram que 62,4% das mensagens recebidas foram encaminhadas unicamente por caminhos infectados, 19,4% foram unicamente por caminhos legítimos e 18,2% foram por caminhos infectados e legítimos simultaneamente. Quanto ao *blackhole*, 29% das mensagens são recebidas na estação-base e nunca foram descartadas, 33% foram recebidas e simultaneamente descartadas num dos caminhos, e 37% foram descartadas e nunca chegaram a ser recebidas.

Estes resultados demonstram que o INSENS contorna relativamente bem o *blackhole* e razoavelmente o *wormhole-overlay*, quando consideradas percentagens de atacantes pouco elevadas, permitindo através do multi-encaminhamento entregar uma parte das mensagens mesmo que estas em alguma zona da rede sejam recebidas por atacantes.

Eficácia dos ataques à selecção de rotas

Em seguida, pretende-se testar a eficácia de ataques à selecção de rotas, o teste consiste em analisar o número médio de mensagens que passam por cada nó legítimo e o número médio que passa por cada nó atacante. Por comparação destes valores será possível obter uma visão geral da eficácia do ataque sem que os resultados se exprimam em termos de cobertura, latência, fiabilidade ou energia.

Este teste realizou-se nos protocolos Clean-Slate e INSENS, nas mesmas condições

do teste anterior de multi-encaminhamento, para os ataques *hello-flooding* (não aplicado ao INSENS devido a implicações na cobertura), *sinkhole* (não ocorre no Clean-Slate), *sybil* (segundo a implementação considerada), *wormhole-simples* e *wormhole-overlay*.

Tabela 6.6: Eficácia dos ataques à selecção de rotas.

Ataque	**Clean-Slate**	**INSENS**
Hello-Flooding	11,74 / 10,85 (56,39%)	-
Sinkhole	-	4,64 / 4,85 (78,8%)
Sybil	12,90 / 13,45 (45,2%)	5,60 / 3,81 (80,8%)
Wormhole-simples	12,92 / 10,48 (42,4%)	10,44 / 2,64 (94,8%)
Wormhole-overlay	391,0 / 10,87 (62,4%)	315,0 / 2,05 (85,6%)

O resultado obtido está expresso na tabela 6.6, sendo apresentados os valores na forma x / y, onde x representa o número médio de mensagens por nó atacante (poderá ser por *wormhole* ou *overlay*) e y o número médio de mensagens por nó legítimo, seguido da taxa de fiabilidade obtida.

Por análise dos resultados verifica-se que o número médio de mensagens em nós legítimos e atacantes para os ataques *hello-flooding*, *sinkhole* e *sybil* é relativamente parecido, significando que estes ataques não conseguem desviar o tráfego face aos mecanismos de resiliência dos protocolos, pois obtêm tantas mensagens quanto os restantes nós da rede. O ataque *sybil* passando a ter uma abordagem que considerasse a utilização de segredos criptográficos de outros nós capturados, porventura causaria algum impacto.

Quanto ao ataque *wormhole-simples*, observou-se uma diferença significativa para o protocolo INSENS, concluindo-se que este ataque influencia bastante os caminhos por onde as mensagens são encaminhadas, o que se deve essencialmente ao facto de constituírem ligações de longa distância que permitem conectar várias zonas da rede, de tal forma que a estação-base ao basear-se nos caminhos mais curtos tira imediatamente proveito destas ligações. No Clean-Slate verificou-se experimentalmente que o número médio de mensagens por *wormhole* é sempre superior ao dos nós legítimos, porém o desvio de tráfego é conseguido mas com uma eficácia muito baixa, pois os caminhos são criados sem qualquer critério que dê preferência a estas ligações, sendo que o baixo desvio está relacionado com o facto de estas ligações permitirem que grupos de nós distantes possam comunicar entre si.

Repare-se que este resultado do *wormhole-simples* é condizente com o facto de o ataque ter tido impacto em todos os critérios de medição anteriormente testados (cobertura, latência, fiabilidade e energia), demonstrando que produz efeito em ambos os protocolos, principalmente no INSENS. Embora, em geral, os resultados dos principais critérios de medição tenham melhorado, isto não é de todo positivo porque uma

grande parte das mensagens passa a estar sob a posse do atacante, que pode agora realizar ataques arbitrários.

Por fim, avaliando o ataque *wormhole-overlay* constata-se que para ambos os protocolos o ataque provoca um impacto muito significativo, tal que a *overlay* obtém uma grande parte do tráfego, o que se deve às várias ligações de longa distância que a constituem e permitem assim recolher mensagens em qualquer zona da rede.

6.8 Síntese de resultados e Avaliação geral

Com esta síntese de resultados pretende-se clarificar e oferecer uma visão comparativa dos resultados obtidos para os protocolos abordados, elaborando ainda uma avaliação geral quem tem por referência os resultados teoricamente previstos.

Inicialmente, concluiu-se que a posição da estação-base não tem influência nos resultados do Clean-Slate, enquanto no INSENS esta deve encontrar-se no centro da rede para melhorar a cobertura. Quanto à relação dimensão/número de nós, concluiu-se que tanto o Clean-Slate como o Flooding são protocolos que se adaptam a áreas com baixa concentração de nós, enquanto o INSENS requer uma alta concentração de nós.

As tabelas seguintes apresentam, em geral, os resultados para os vários critérios de medição, sendo apresentado o resultado sem ataque, bem como o resultado com ataque (indica se ocorre ou não um melhoramento em relação ao resultado sem ataque, utiliza-se a notação "==" para representar o caso em que o resultado se mantém igual).

Tabela 6.7: Resumo dos resultados de cobertura total.

Ataque	**Clean-Slate**	**INSENS**	**Flooding**
Sem ataque	Alta	Razoável	Muito alta
Hello-Flooding	==	Piora muito	-
Sinkhole	-	==	-
Sybil	==	==	-
Wormhole-simples	Melhora	Melhora	-
Wormhole-overlay	Melhora	Melhora muito	-
Wormhole-mitm	Piora muito	Piora muito	-
Blackhole	Piora muito	Piora pouco	-
Selective Forw.	Piora*	Piora*	-

Em 6.7 observa-se que os ataques que demonstraram causar algum tipo de impacto na cobertura total de algum dos protocolos são: *hello-flooding*, *wormhole-simples*, *wormhole-overlay*, *wormhole-mitm*, *blackhole* e *selective-forwarding*. Considera-se que o *selective-forwarding* piora a cobertura, no sentido em que se admite que existe a probabilidade de descarte das mensagens, mas nunca sendo pior que o *blackhole*.

Tabela 6.8: Resumo dos resultados de fiabilidade.

Ataque	**Clean-Slate**	**INSENS**	**Flooding**
Sem ataque	Baixa	Razoável	Alta
Hello-Flooding	==	Sem cobertura	-
Sinkhole	-	==	-
Sybil	==	==	-
Wormhole-simples	Melhora (irregularmente)	Melhora	-
Wormhole-overlay	Melhora	Melhora	-
Wormhole-mitm	Sem cobertura	Sem cobertura	-

Na tabela 6.8 observa-se que os ataques que provocam impacto na fiabilidade de pelo menos um dos protocolos são os *wormhole-simples* e *wormhole-overlay*. Os ataques *blackhole* e *selective-forwarding* não foram submetidos ao teste de fiabilidade pelas razões já referidas em 6.6.7, o mesmo se aplica quando não existe cobertura ("Sem cobertura").

Tabela 6.9: Resumo dos resultados da latência.

Ataque	**Clean-Slate**	**INSENS**	**Flooding**
Sem ataque	Baixa	Muito baixa	Muito Alta
Hello-Flooding	==	Sem cobertura	-
Sinkhole	-	Afectada se > 50% atacantes	-
Sybil	==	==	-
Wormhole-simples	Melhora	Melhora e piora	-
Wormhole-overlay	Piora pouco	Piora muito	-
Wormhole-mitm	Sem cobertura	Sem cobertura	-

Na tabela 6.9 observa-se que os ataques que provocam impacto na latência de pelo menos um dos protocolos são: *sinkhole* (a partir de 50% de atacantes), *wormhole-simples* e *wormhole-overlay*.

Tabela 6.10: Resumo dos resultados de consumo energético no encaminhamento.

Ataque	**Clean-Slate**	**INSENS**	**Flooding**
Sem ataque	Baixo	Muito baixo	Muito alto
Hello-Flooding	==	Sem cobertura	-
Sinkhole	-	==	-
Sybil	==	==	-
Wormhole-simples	==	Melhora	-
Wormhole-overlay	Melhora	Melhora	-
Wormhole-mitm	Sem cobertura	Sem cobertura	-

Quanto ao consumo durante a fase de configuração do protocolo, o Flooding é aquele que menos energia utiliza, enquanto o INSENS tem um consumo relativamente moderado, por fim o Clean-Slate requer um acentuado gasto energético. Em relação à energia despendida durante a fase de encaminhamento, como se pode observar na tabela 6.10, os ataques *wormhole-simples* e *wormhole-overlay* são os que provocam impacto.

De seguida, em 6.11 apresenta-se uma visão dos resultados obtidos para o teste que avalia a eficácia dos ataques à selecção de rotas e o teste do multi-encaminhamento. No primeiro teste é indicado na tabela à esquerda se o ataque afecta muito, pouco, ou nada o protocolo, no sentido do ataque conseguir desviar as mensagens para nós atacantes. Na tabela à direita apresenta-se a qualidade do multi-encaminhamento, representando a capacidade de contornar os atacantes, ou se este não é efectuado.

Ataque	**Clean-Slate**	**INSENS**
Hello-Flooding	Não	-
Sinkhole	-	Não
Sybil	Não	Não
Wormhole-simples	Pouco	Muito
Wormhole-overlay	Muito	Muito

Ataque	**Clean-Slate**	**INSENS**
Wormhole-overlay	-	Razoável
Blackhole	-	Bom

Tabela 6.11: Ataques à selecção (Esquerda) e Multi-encaminhamento (Direita).

Fazendo uma avaliação geral aos ataques, tendo por base os resultados previstos e apresentados na tabela 2.1 em 2.2.2, concluiu-se que o *hello-flooding*, *sinkhole* (excepto na medição de latência para muitos atacantes) e *sybil* obtiveram resultados que correspondem aos teoricamente previstos, se bem que o *sybil* pode ser desencadeado segundo outras aproximações mais eficazes que, porventura, colocam em causa a efectiva resiliência dos mecanismos utilizados.

O resultado do *wormhole-simples* no Clean-Slate difere do teoricamente previsto, uma vez que se esperava que o encaminhamento resiliente impedisse este ataque, quando na verdade demonstrou ser insuficiente e o ataque acaba assim por provocar algum impacto. No INSENS teve ainda um impacto mais notável, levando a concluir que a estação-base, sendo responsável pelo cálculo de rotas, não garante resiliência, e mesmo o contributo dado pelos caminhos redundantes também não o resolvem por completo. Os tipos *wormhole-mitm* e *wormhole-overlay* são casos particulares que não foram contemplados na abordagem teórica, no entanto experimentalmente revelaram um impacto considerável em ambos os protocolos.

Por fim, nos ataques *blackhole* e *selective-forwarding* o INSENS correspondeu ao teoricamente previsto, ao contrário do Clean-Slate que mostrou na prática ser substancialmente afectado, colocando em causa o seu mecanismo de encaminhamento resiliente, pois enquanto no INSENS as mensagens são enviadas por dois caminhos em simultâneo, no Clean-Slate apenas percorrem um caminho usando uma direcção aleatória, o que se revelou insuficiente.

Numa perspectiva geral, sem contemplar ataques, é notável que o Flooding evidenciou ser muito benéfico ao nível da cobertura e fiabilidade, porém a latência e energia apresentaram valores muito elevados, o que se deve à inexistência de caminhos adequados e por não considerar as restrições energéticas, sendo então desapropriado para este tipo de redes que são tão limitadas energeticamente.

O Clean-Slate é o protocolo que obtém melhor cobertura quando comparado com o INSENS, e requer também um menor número de sensores para cobrir uma determinada área, tornando-se numa solução mais económica. No entanto, ao nível da fiabilidade, latência e consumo energético é o INSENS que apresenta os melhores valores, evidenciando uma boa gestão energética que se deve essencialmente ao contributo oferecido pela estação-base. Ao contrário do INSENS, o Clean-Slate durante a fase de configuração exige um alto dispêndio de energia, necessitando assim de condições energéticas que permitam o protocolo ser correctamente executado.

De um modo geral e tendo em conta os ataques, os resultados obtidos permitem inferir que o Clean-Slate lida relativamente melhor que o INSENS nos ataques à selecção de rotas, principalmente por não se basear em nenhum critério de construção de caminhos (por exemplo o número de *hops* do caminho), por conseguinte a latência, fiabilidade e consumo energético são mais afectados que no INSENS, visto que este, por sua vez, cria os caminhos mais curtos. Por outro lado, o INSENS revelou lidar melhor com os ataques que incidem na fase de controlo de encaminhamento (*blackhole* e *selective-forwarding*), muito por força do seu encaminhamento redundante.

Na verdade, embora alguns ataques não causem qualquer impacto nos critérios de medição, como é o caso de alguns ataques à selecção de rotas que não conseguem desviar o tráfego, os nós atacantes não são efectivamente detectados e excluídos da rede, o que permite que esses atacantes realizem ataques arbitrários com as mensagens que lhes são encaminhadas. O INSENS, não detectando os nós atacantes, acaba ainda assim por beneficiar, em certa forma, do seu encaminhamento redundante. Para que estes protocolos mantenham uma resiliência completa precisariam de mecanismos de detecção que possam detectar nós atacantes e excluí-los por completo da rede.

7

Conclusões

7.1 Conclusões

Tendo em conta os objectivos e as contribuições inicialmente propostas, a presente dissertação pretendeu oferecer uma visão da importância da segurança nas redes de sensores sem fios (RSSF), essencialmente no que diz respeito ao encaminhamento, enfatizando-se as características destas redes que são assim determinantes para a concepção dos sistemas de encaminhamento seguro.

De facto, as RSSF são redes constituídas por dispositivos (sensores) de pequenas dimensões, dotados de capacidades computacionais limitadas, com recursos de comunicação reduzidos e com restrições energéticas notáveis. Normalmente estas redes encontram-se desprovidas de vigilância, o que requer então que os protocolos de encaminhamento sejam tão seguros quanto possível e que mantenham o bom funcionamento da rede relativamente ao encaminhamento de dados.

Tal como proposto nos objectivos da dissertação, foi desenvolvida uma plataforma de simulação que permite então conceber e implementar protocolos de encaminhamento, e posteriormente submetê-los a medições que permitem realizar uma rigorosa análise e avaliação destes. Esta plataforma foi implementada tendo por base um simulador escolhido previamente, neste caso o simulador escolhido foi o JProwler, no entanto foi utilizada uma versão já existente, designada por WiSeNet Simulator [dS11], que tem como núcleo o próprio JProwler.

Com o desenvolvimento da plataforma de simulação, na qual foram implementados módulos de medição, foi assim possível analisar o comportamento de determinados protocolos de encaminhamento seguro e ainda proceder às análises e avaliações destes, permitindo retirar as devidas conclusões. Vários protocolos de encaminhamento seguro têm sido propostos, porém nesta dissertação focaram-se dois protocolos seguros de larga escala que abordam a defesa pró-activa contra intrusões, mais concretamente os protocolos INSENS e Clean-Slate. Além destes, foi ainda implementado o protocolo Flooding, surgindo apenas como um protocolo de referência, e que não contém qualquer tipo de mecanismo de segurança. Uma vez que estamos a lidar com protocolos de encaminhamento seguro, interessou elaborar uma abordagem aos ataques que incidem no nível de encaminhamento e criar um modelo de adversário a considerar. O modelo de adversário especificado contempla os principais ataques ao encaminhamento: *hello-flooding*, *sinkhole*, *wormhole*, *sybil*, *blackhole* e *selective-forwarding*. Estes ataques incidem numa das três fases do encaminhamento: (i) descoberta de rotas, (ii) selecção de rotas e (iii) controlo do encaminhamento. Portanto, os ataques foram implementados para que possam ser injectados nos protocolos através da plataforma de simulação.

Os principais critérios de medição utilizados pela plataforma para a avaliação dos protocolos são a cobertura, latência, fiabilidade e energia, sendo que estes critérios revelaram-se muito úteis por conseguirem reflectir bem a qualidade de encaminhamento do protocolo. Nestes termos, cada protocolo foi testado com e sem ataque, obtendo-se os respectivos resultados em cada um dos critérios referidos.

Relativamente à cobertura, um primeiro aspecto avaliado foi a influência da posição da estação-base, tendo-se concluído que o Clean-Slate não é afectado pela posição, contrariamente a cobertura do INSENS melhora significativamente quando a posição é o centro da rede. A relação entre as dimensões da área e o número de nós da rede evidenciou-se também como um factor essencial, pois o Clean-Slate e o Flooding operam melhor em áreas com baixa concentração de nós, enquanto o INSENS se apropria a áreas com alta concentração de nós.

Na ausência de ataques o Flooding é o protocolo que adquire melhor taxa de cobertura, seguido do Clean-Slate que também apresenta resultados muito positivos, por fim o INSENS obtém valores de cobertura inferiores. Quanto aos ataques onde a cobertura registou alterações, os *blackholes* afectam pouco o INSENS mas este é muito afectado pelo ataque *hello-flooding*, ao invés o Clean-Slate é bastante afectado por *blackholes* mas não pelo *hello-flooding*. Com os ataques *wormhole-simples* e *wormhole-overlay* ambos os protocolos melhoram a cobertura, e com o *wormhole-mitm* são muito afectados.

Quanto à fiabilidade, analisaram-se vários cenários de congestionamento e concluiu-se que para qualquer cenário o protocolo Flooding é aquele que obtém melhor taxa de fiabilidade, sendo seguido pelo INSENS que se revelou também com valores razoáveis, o Clean-Slate foi o que apresentou a taxa de fiabilidade mais baixa. Relativamente aos ataques medidos, apenas o *wormhole-simples* e *wormhole-overlay* provocaram impacto na fiabilidade.

Na avaliação de latência sem ataques concluiu-se que no Flooding a latência é muito elevada, na ordem dos segundos, enquanto o Clean-Slate e o INSENS apresentam valores na ordem dos milissegundos, ainda assim o INSENS é o protocolo que tem a menor latência, este resultado aplica-se em qualquer cenário de congestionamento. O INSENS é também o protocolo que encaminha as mensagens com um menor número de saltos, e o Clean-Slate o que requer mais saltos. Quanto aos ataques, o *sinkhole* (com muitos atacantes) apenas afecta o INSENS; o *wormhole-simples* melhora a latência do Clean-Slate, enquanto no INSENS melhora e agrava a partir de um certo número de nós. No ataque *wormhole-overlay* tanto o Clean-Slate como o INSENS são afectados, principalmente este último, os restantes ataques medidos não provocam alterações.

Em relação à energia na fase de configuração do protocolo, o Clean-Slate requer um elevado dispêndio energético, o INSENS tem um consumo moderado, e o Flooding um baixo consumo. Durante o encaminhamento e sem ataques, o Flooding é o protocolo que mais energia consome, seguindo-se o Clean-Slate que requer bastante menos, e o INSENS é o que menos energia utiliza. Em relação aos ataques, no *wormhole-simples* apenas o INSENS baixa o seu consumo, no *wormhole-overlay* ambos os protocolos baixam, e os restantes ataques possíveis de medir não provocam alterações.

Analisaram-se ainda outras características dos protocolos que levaram a concluir que o INSENS com o multi-encaminhamento permite contornar relativamente bem o *blackhole* e razoavelmente o *wormhole-overlay*, enquanto o Clean-Slate não suporta este mecanismo. Foi também analisada a eficácia dos ataques à selecção de rotas e o *wormhole-simples* e *wormhole-overlay* mostraram ser os que mais efeitos produzem.

Atendendo aos resultados obtidos nos testes efectuados no capítulo 6, os protocolos estudados apresentam características distintas entre si, tanto na presença como na ausência de ataques, concluindo-se logo à partida que nenhum dos protocolos se superioriza totalmente, isto porque cada protocolo tanto se revela benéfico em determinadas situações, como se poderá tornar desvantajoso em outras.

Concluiu-se que o protocolo Flooding, sendo o protocolo de referência, é aquele que permite melhor cobertura e fiabilidade, no entanto a latência e o consumo energético durante o encaminhamento apresentam valores indesejáveis. O seu gasto energético

é tão elevado que demonstra logo ser muito desapropriado para este tipo de redes, e também por não incluir qualquer noção de segurança.

Relativamente aos protocolos de especial foco, o Clean-Slate obtém uma cobertura bastante boa em relação ao INSENS, contudo é o INSENS que em geral oferece melhor fiabilidade, latência e consumo energético. O Clean-Slate mostrou também ser uma solução mais económica por exigir um menor número de sensores para cobrir uma determinada área. Em termos de energia, a fase de configuração do protocolo Clean-Slate exige um alto consumo energético, então a utilização deste protocolo deve ser feita se as condições de energia forem as necessárias para garantir o seu bom funcionamento.

Na presença de ataques, o Clean-Slate mostrou conseguir lidar melhor com os ataques à selecção de rotas, por não utilizar nenhum critério específico de construção de caminhos, enquanto o INSENS mostrou lidar melhor com os ataques inseridos na fase de controlo de encaminhamento, tirando partido do seu encaminhamento redundante. Não obstante, embora alguns ataques não provoquem impacto nos critérios de medição, os nós atacantes continuam a ser utilizados para encaminhar tráfego, o que é ainda vulnerável a ataques arbitrários. Nesse sentido, estes protocolos precisariam de mecanismos de resiliência que permitissem a detecção e exclusão total de nós atacantes.

Portanto, se fosse pretendido implementar um destes protocolos num ambiente real, a escolha do protocolo a utilizar deverá ser feita segundo uma linha de orientação que tem por base as características dos protocolos, as condições que envolvem o meio onde a rede é implantada (que pressupõe as condições de segurança envolventes), bem como as características dos dispositivos, e ainda o propósito aplicacional da rede, para que se possa beneficiar das vantagens do protocolo escolhido.

7.2 Assuntos em aberto

Nesta dissertação fica em aberto a implementação do ataque *sybil* segundo uma abordagem mais intrusiva, em que os nós atacantes possam manter os segredos que foram capturados em outros nós, preparando a respectiva análise e avaliação do impacto provocado pelo ataque em tais condições.

Outro aspecto considerado pertinente passa por implementar o ataque de replicação de nós, o qual obriga a que os protocolos utilizem mecanismos adequados [PPG05]. O Clean-Slate refere na sua documentação este tipo de mecanismo, na medida em que ao ser detectada uma replicação possa seguidamente ser feita a recuperação da rede por intermédio do mecanismo *honey-bee*.

Fica também em aberto a possível implementação de um gestor de resultados que

permita gerir de um modo mais elaborado os resultados obtidos , bem como produzir e apresentar uma maior variedade de gráficos que facilitem o processo de análise e avaliação dos resultados. Considera-se também o desenvolvimento da camada MAC de modo a utilizar de uma forma mais realista o modelo de energia, por exemplo passando a preconizar efectivamente o estado *sleep*, e, por fim, que o tempo de transmissão de cada mensagem seja em função do seu tamanho, ao invés de se utilizar um valor médio.

7.3 Trabalho futuro

Uma vez que nesta dissertação é desenvolvida uma plataforma para análise e avaliação de sistemas de encaminhamento seguro, como trabalho futuro será interessante por intermédio da plataforma dar continuidade ao estudo já elaborado, no sentido em que se possa obter uma visão e análise mais extensiva dos sistemas de encaminhamento seguro quando sujeitos aos ataques contemplados, os quais incluem uma abordagem intrusiva.

O Clean-Slate e o INSENS são protocolos de referência na investigação recente, e a avaliação realizada permitiu obter uma visão concreta para ambos. Contudo, o facto de apenas serem testados estes protocolos torna, em certo modo, essa visão limitada, na medida em que é efectuada uma comparação que não abrange a análise e características de outros sistemas de encaminhamento seguro. Portanto, além dos protocolos de referência avaliados, outros protocolos na mesma linha de orientação (direccionados a intrusões) poderão ser implementados, permitindo concluir experimentalmente se o INSENS e o Clean-Slate, ao serem considerados protocolos de referência, são os melhores em termos de resiliência e operabilidade relativamente a outros, bem como verificar se algum outro protocolo se revela efectivamente melhor que os restantes.

Por outro lado, para complementar a sugestão anterior, será interessante considerar a possibilidade de os ataques contemplados serem também injectados durante as simulações, constituindo a possibilidade de injecção de código dinâmico. Isto tem como objectivo reflectir condições em que o atacante actua durante o funcionamento da rede.

Bibliografia

[41503] Ieee standard for information technology - telecommunications and information exchange between systems - local and metropolitan area networks specific requirements part 15.4: Wireless medium access control (mac) and physical layer (phy) specifications for low-rate wireless personal area networks (lr-wpans). *IEEE Std 802.15.4-2003*, pág. 1–670, 2003.

[BPC+07] Paolo Baronti, Prashant Pillai, Vince W.C. Chook, Stefano Chessa, Alberto Gotta, e Y. Fun Hu. Wireless sensor networks: A survey on the state of the art and the 802.15.4 and zigbee standards. *Computer Communications*, 30(7):1655 – 1695, 2007. Wired/Wireless Internet Communications.

[CFP+06] K. Chintalapudi, T. Fu, J. Paek, N. Kothari, S. Rangwala, J. Caffrey, R. Govindan, E. Johnson, e S. Masri. Monitoring civil structures with a wireless sensor network. *Internet Computing, IEEE*, 10(2):26 – 34, 2006.

[cro] Crossbow technology, http://www.xbow.com.

[dS11] Pedro Marques da Silva. Avaliação de condições de fiabilidade e segurança de protocolos de encaminhamento de dados em redes de sensores sem fios. Tese de Mestrado, Faculdade de Ciências e Tecnologia - Universidade Nova de Lisboa, 2011.

[DY83] D. Dolev e A. Yao. On the security of public key protocols. *Information Theory, IEEE Transactions on*, 29(2):198 – 208, Março 1983.

[GSSK05] G. Guimaraes, E. Souto, D. Sadok, e J. Kelner. Evaluation of security mechanisms in wireless sensor networks. In *Systems Communications, 2005. Proceedings*, pág. 428 – 433, 2005.

[HKL+06] Tian He, Sudha Krishnamurthy, Liqian Luo, Ting Yan, Lin Gu, Radu Stoleru, Gang Zhou, Qing Cao, Pascal Vicaire, John A. Stankovic, Tarek F. Abdelzaher, Jonathan Hui, e Bruce Krogh. Vigilnet: An integrated sensor network system for energy-efficient surveillance. *ACM Trans. Sen. Netw.*, 2:1–38, February 2006.

[JD02] Shivakant Mishra Jing Deng, Richard Han. Insens: Intrusion-tolerant routing in wireless sensor networks. Relatório técnico, Department of Computer Science, University of Colorado, Boulder, Colorado, USA, 2002.

[JPr] Jprowler, http://w3.isis.vanderbilt.edu/projects/nest/prowler/index.html.

[KSW04] Chris Karlof, Naveen Sastry, e David Wagner. Tinysec: a link layer security architecture for wireless sensor networks. In *Proceedings of the 2nd international conference on Embedded networked sensor systems*, SenSys '04, pág. 162–175, New York, NY, USA, 2004. ACM.

[KW03] C. Karlof e D. Wagner. Secure routing in wireless sensor networks: attacks and countermeasures. In *Sensor Network Protocols and Applications, 2003. Proceedings of the First IEEE. 2003 IEEE International Workshop on*, pág. 113 – 127, Maio 2003.

[Lev] Tossim system description, http://www.tinyos.net/nest/doc/tossim.pdf.

[LL] Philip Levis e Nelson Lee. Tossim: A simulator for tinyos networks, http://www.cs.berkeley.edu/ pal/pubs/nido.pdf.

[MFM05] V.B. Misic, Jun Fang, e J. Misic. Mac layer security of 802.15.4-compliant networks. In *Mobile Adhoc and Sensor Systems Conference, 2005. IEEE International Conference on*, pág. 8 pp. –854, 2005.

[MKKW08] Timothée Maret, Raphaël Kummer, Peter Kropf, e Jean-Frédéric Wagen. Freemote emulator: a lightweight and visual java emulator for wsn. In *Proceedings of the 6th international conference on Wired/wireless internet communications*, WWIC'08, pág. 92–103, Berlin, Heidelberg, 2008. Springer-Verlag.

[MOJ06] Ar Milenković, Chris Otto, e Emil Jovanov. Wireless sensor networks for personal health monitoring: Issues and an implementation. *Computer Communications (Special issue: Wireless Sensor Networks: Performance, Reliability, Security, and Beyond*, 29:2521–2533, 2006.

[PCG+05] Jeongyeup Paek, K. Chintalapudi, R. Govindan, J. Caffrey, e S. Masri. A wireless sensor network for structural health monitoring: Performance and experience. In *Embedded Networked Sensors, 2005. EmNetS-II. The Second IEEE Workshop on*, pág. 1 – 10, Maio 2005.

[PLGP06] Bryan Parno, Mark Luk, Evan Gaustad, e Adrian Perrig. Secure sensor network routing: a clean-slate approach. In *Proceedings of the 2006 ACM CoNEXT conference*, CoNEXT '06, pág. 11:1–11:13, New York, NY, USA, 2006. ACM.

[PPG05] B. Parno, A. Perrig, e V. Gligor. Distributed detection of node replication attacks in sensor networks. In *Security and Privacy, 2005 IEEE Symposium on*, pág. 49 – 63, Maio 2005.

[PSM+04] Joseph Polastre, Robert Szewczyk, Alan Mainwaring, David Culler, e John Anderson. *Analysis of wireless sensor networks for habitat monitoring*, pág. 399–423. Kluwer Academic Publishers, Norwell, MA, USA, 2004.

[Rit06] Ana Sofia Querido Rito. Redes de sensores sem fios. Tese de Mestrado, Faculdade de Ciências e Tecnologia - Universidade Nova de Lisboa, 2006.

[Shi00] R. Shirey. Internet security glossary, 2000.

[SKG+05] Vipul Singhvi, Andreas Krause, Carlos Guestrin, James H. Garrett, Jr., e H. Scott Matthews. Intelligent light control using sensor networks. In *Proceedings of the 3rd international conference on Embedded networked sensor systems*, SenSys '05, pág. 218–229, New York, NY, USA, 2005. ACM.

[Sta05] William Stallings. *Cryptography and Network Security (4th Edition)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2005.

[TZC06] V.W.S. Tang, Yuan Zheng, e Jiannong Cao. An intelligent car park management system based on wireless sensor networks. In *Pervasive Computing and Applications, 2006 1st International Symposium on*, pág. 65 –70, 2006.

[WFSH06] Anthony D. Wood, Lei Fang, John A. Stankovic, e Tian He. Sigf: a family of configurable, secure routing protocols for wireless sensor networks. In *Proceedings of the fourth ACM workshop on Security of ad hoc and sensor networks*, SASN '06, pág. 35–48, New York, NY, USA, 2006. ACM.

[WGE+05] A.S. Wander, N. Gura, H. Eberle, V. Gupta, e S.C. Shantz. Energy analysis of public-key cryptography for wireless sensor networks. In *Pervasive*

Computing and Communications, 2005. PerCom 2005. Third IEEE International Conference on, pág. 324 – 328, 2005.

[YCW06] Yu-Chee Tseng You-Chiun Wang. Attacks and defenses of routing mechanisms in ad hoc and sensor networks. In *Security in Sensor Networks*, pág. 3–23. Auerbach Publications, 2006.

Introdução às RSSF

A.1 Redes de Sensores sem Fios

As redes de sensores sem fios (RSSF) são redes de comunicação por radiofrequência (baseadas nas normas IEEE 802.15.4 [41503] ou Zigbee [BPC+07] com possíveis variantes associadas a protocolos de acesso ao meio, livres de colisões [Rit06]). As RSSF são formadas por pequenos dispositivos computorizados (vulgarmente apenas designados por sensores) de baixo custo (perspectivando-se que possam vir a custar cerca de 1 euro). Estes dispositivos podem ter dimensões mais ou menos reduzidas (da ordem de dezenas de milímetros cúbicos a dezenas de centímetros cúbicos), sendo dotados de capacidades computacionais e de comunicação. Normalmente um sensor é fabricado sobre um substrato integrado que contém bateria para funcionamento autónomo, conversor analógico-digital para capacidades de processamento de sinal, um transmissor e receptor de rádio, um micro-controlador, e um ou mais sensores propriamente ditos, que podem permitir medir ou monitorar fenómenos físicos do meio ambiente envolvente. Os sensores podem ser distribuídos por áreas geográficas de maior ou menor dimensão, podendo essas áreas serem eventualmente povoadas por sensores de forma mais ou menos densa, aproveitando o baixo custo dos dispositivos, a facilidade de instalação dos mesmos e a não necessidade de supervisão ou não necessidade de organização planeada.

Numa RSSF os sensores cooperam de forma a monitorizarem eventos que resultam da actividade de processamento de sinal e pré-processamento local de informação

ao nível dos nós, sendo esta posteriormente transmitida ao longo da rede. Os nós interagem entre si e com o meio ambiente, medindo valores associados aos fenómenos físicos. Actualmente, os nós sensores podem conter uma variedade de sensores, tais como: medição de temperatura, detecção de som ou ruído, medição de humidade, detecção de movimento, acelerómetros, medição de indicadores de poluição ou de níveis de concentração de certo tipo de substâncias no meio ambiente, medição de pressão, medição de vibração e medição de luminosidade.

Dependendo dos requisitos de escala e devido ao limitado alcance das comunicações rádio no espectro de operação das normas IEEE 802.15.4 ou Zigbee, a propagação de dados pela rede pode ter que exigir formas de difusão com base em encaminhamento par-a-par, de acordo com estruturas de encaminhamento de topologia *multi-hop* (múltiplos saltos).

Durante o encaminhamento dos dados ao longo da rede, estes podem ser processados e agregados de forma intermédia por outros sensores, o que permite evitar estratégias de encaminhamento por inundação e que disseminem todos os eventos gerados na rede, o que seria mau do ponto de vista do número de mensagens transmitidas na rede e dos custos energéticos associados à comunicação. Na verdade, em RSSF os custos de comunicação são determinantes no que diz respeito ao consumo de energia.

Os dados processados ou transmitidos ao longo da rede podem ser recolhidos em nós especiais de captura de dados, os quais podem possuir características mais ou menos especializadas. Estes nós são habitualmente designados por estações-base ou nós de recolha de dados (*base stations* ou *sink nodes* na terminologia de língua inglesa). Estes estão normalmente interligados com redes e sistemas externos (que podem ser baseados em tecnologias de redes locais ou infra-estruturas TCP/IP, por exemplo), onde possivelmente existem aplicações de tratamento final e visualização dos dados recolhidos bem como outras ferramentas de gestão e análise desses dados ou mesmo de monitorização externa das RSSF.

A integração de RSSF poderá ser feita com outras infra-estruturas de redes de comunicação convencionais, por exemplo através de tecnologias de redes locais com ou sem fios, ambientes de redes metropolitanas, redes GSM ou mesmo num ambiente de rede Internet interligando diferentes topologias de redes.

A.2 Dispositivos e suas limitações numa RSSF

Atendendo a características que permitam que os sensores sejam dispositivos de custo muito reduzido e aos componentes utilizados e requisitos da sua operação, os dispositivos possuem geralmente recursos muito limitados, quer ao nível da capacidade

computacional, quer ao nível do alcance e capacidade de comunicação (que podem limitar-se a distâncias de poucos metros a cerca de uma centena de metros em campo aberto, dependendo da tecnologia em causa, das condições de cobertura ou de factores, como por exemplo, a temperatura ambiente ou a humidade envolvente). Apresentam ainda limitações de memória, não apenas pela necessidade de miniaturização que limita a capacidade de armazenamento, mas sobretudo pela miniaturização dos micro-controladores e suas limitações para endereçamento da memória disponível.

Uma das importantes restrições diz respeito à energia, uma vez que por concepção os nós são desenvolvidos para poderem operar em condições de autonomia e sem supervisão. Em cenários onde não seja de todo possível o acesso ou supervisão por humanos (como por exemplo na monitorização de áreas geográficas remotas e hostis para acesso humano), a energia representa não apenas um recurso escasso como constitui na verdade um recurso finito, que limita o tempo de vida útil dos nós sensores e, por conseguinte, o de toda a rede. Assim, a energia numa RSSF deve ser utilizada com grande contenção, pois é um factor determinante na avaliação de soluções de *software* desenhadas para uma pilha de uma RSSF.

Devido às sérias limitações de energia, o número e a periodicidade de operações de detecção, processamento e, sobretudo, transmissão e recepção de informação, mas também processamento computacional muito exigente, podem ser restringidos na prática, sob pena de condicionarem fortemente o tempo de vida útil das redes [KW03], exemplo disso é o caso de certas operações criptográficas que utilizam criptografia assimétrica.

A.3 *Hardware* típico de um sensor

O hardware utilizado nos sensores pode ser diferente de acordo com a tecnologia usada. Alguns exemplos de sensores utilizados vulgarmente nas redes de sensores reais [cro] são apresentados de seguida:

- TelosB Motes (tecnologia da empresa *Crossbow*): Micro-processador TI MSP430F1611 de 16 bits a 8 MHz, 10 Kbytes de memória RAM, 48 Kbytes de memória dedicada a aplicações, 1024 Kbytes de memória para registar dados, duas baterias AA, permite alcance de 100 metros em campo aberto e uma velocidade de transmissão de 250 Kbps;

- MICAz (tecnologia da empresa *Crossbow*): Micro-processador ATMEL ATmega128L de 8 bits a 7.37 MHz, 128 Kbytes de memória dedicada a aplicações, 512 Kbytes

de memória para armazenamento de dados, duas baterias AA, permite alcance de 100 metros em campo aberto e uma velocidade de transmissão de 250 Kbps;

- MICA2DOT (tecnologia da empresa *Crossbow*): Micro-processador ATMEL ATmega128L de 8 bits a 4 MHz, 128 Kbytes de memória dedicada a aplicações, 512 Kbytes de memória para armazenamento de dados e uma bateria do tipo célula-moeda de 3V.

A figura A.1 representa, como referência, a assemblagem de componentes no caso de um sensor TelosB.

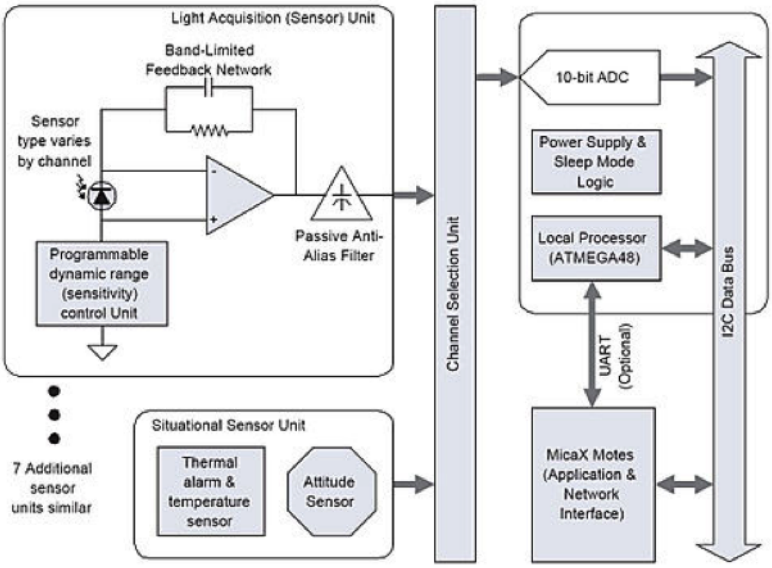


Figura A.1: Componentes num sensor *TelosB*.

A.4 Organização e operação de uma RSSF

Do ponto de vista externo aos sensores, uma RSSF forma um sistema distribuído capaz de capturar dados provenientes da actividade independente, autónoma e assíncrona dos sensores, pré-processar esses dados ao nível do nó origem, transmitir esses dados ou eventualmente agregar os mesmos em nós intermédios ao longo da rede, antes de serem enviados para uma estação-base.

Uma RSSF é geralmente um caso particular de uma rede *ad hoc* sem fios, contudo as RSSF têm características distintas de uma rede *ad hoc* sem fios mais convencional.

Assim, numa RSSF os nós possuem recursos muito mais limitados, não adquirem conhecimento sobre a topologia, não possuem geralmente a possibilidade de obtenção de informação de localização dos nós, são cobertas em geral por um elevado número e densidade de nós, operam sem supervisionamento humano, operam em regimes de intermitência no ciclo de vida de operação e são sujeitos a maiores taxas de erros e falhas de comunicação, bem como interferências externas.

Por norma, uma RSSF é formada por nós sensores comuns e por um ou vários nós *sink*. Dessa forma, habitualmente, o modelo de comunicação passa pelo encaminhamento da informação ao longo dos vários nós até à estação-base. No entanto, em determinadas circunstâncias podem-se ter outros modelos de comunicação tais como *broadcast* ou *multicast*.

A.4.1 Requisitos de escala e auto-organização

Uma ideia central na adopção destas redes é que estas podem ser utilizadas em cenários de grande escala, devendo neste caso ser formadas por um grande número de sensores (na ordem de milhares a dezenas de milhares, na abordagem da investigação actual). Assim, é possível cobrir vastas áreas geográficas desde que se garantam critérios adequados de densidade para essa cobertura, de modo a evitar um aumento significativo de colisões com repercussões no funcionamento da rede e também na gestão de energia.

Interessa-nos que o processo de formação (descoberta e auto-organização da rede) e a sua manutenção topológica seja pouco exigente de intervenção humana, dimensionamento e gestão operacional, ou que não o seja exigido de todo, sendo um aspecto particularmente importante para determinados cenários de utilização e suas aplicações.

A.4.2 Topologias da rede

No que diz respeito aos critérios de auto-organização das RSSF, estes podem condicionar a topologia de formação da rede. Diferentes topologias podem estar associadas a diferentes critérios de escala, requisitos de comunicação, estrutura do encaminhamento e processamento e agregação de dados ao longo da rede. A topologia de uma RSSF pode assim assumir diferentes formas, podendo adoptar estruturas planas, estruturas hierárquicas, estruturas em estrela, estruturas centradas em grupos de agregação (*group* ou *cluster based structures*) ou estruturas em malha (do tipo *mesh based structures*).

Deve notar-se que as diferentes topologias podem apresentar características com

maiores vantagens ou desvantagens, dependendo dos padrões de comunicação e requisitos particulares das aplicações a suportar. Por outro lado, a topologia de uma RSSF é fortemente influenciada pelas características que os seus nós constituintes apresentam, em termos de mobilidade ou requisitos específicos de cobertura.

Se estivermos perante uma rede dinâmica, na qual os nós suportam mobilidade, uma organização do tipo malha pode apropriar-se melhor. Nesta topologia, todos os nós apresentam normalmente as mesmas características de *hardware* e desempenham sempre as mesmas funções, podendo sempre actuar como nós de origem de eventos ou de encaminhamento. Assim, aumenta-se a flexibilidade e maximiza-se a área de cobertura da rede, visto poder aproveitar-se ao máximo qualquer possível conexão entre nós. Porém, o número de ligações entre os nós tende a ser elevado, o que pode ter um impacto considerável na memória ocupada e no consumo energético, resultando num grande número de mensagens trocadas entre os nós. Portanto, a relação entre conectividade e cobertura da rede constitui um factor com influência directa sobre o modo de funcionamento e tempo de vida da rede. A figura A.2 apresenta uma rede organizada segundo uma topologia de malha.

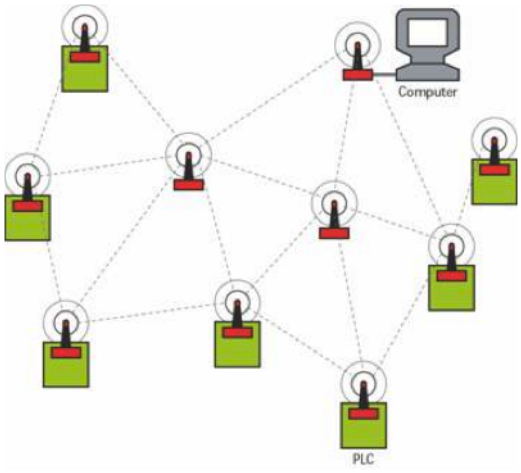


Figura A.2: Rede organizada segundo uma topologia de malha (*mesh-based*).

No caso de se tratar de uma rede estática, em que a posição dos nós é fixa e possa ser predefinida, pode-se optar por uma organização hierárquica ou em grupo. Cada grupo contém um nó agregador que pode ser seleccionado de forma aleatória ou com base nas suas características. Uma organização deste tipo, dependendo da finalidade da rede, pode oferecer algumas vantagens interessantes ao nível da memória ocupada, menor consumo de energia devido a menos conexões e facilitar a distribuição e substituição de chaves(refrescamento).

Por outro lado, os nós que funcionam como agregadores têm um período de vida

mais reduzido devido a realizarem mais processamento e transmissão de dados. Para contornar o problema, os nós agregadores podem ser mais robustos ao nível da capacidade de processamento, memória e energia disponível ou, como alternativa, a rede proceder a reconfigurações durante o seu período de vida, elegendo novos nós agregadores. As figuras A.3 e A.4 ilustram uma rede organizada hierarquicamente, com os nós a criarem grupos. Existem ainda topologias híbridas que tentam conjugar a flexibilidade obtida pelas redes do tipo malha, com o funcionamento estrutural das redes organizadas hierarquicamente.

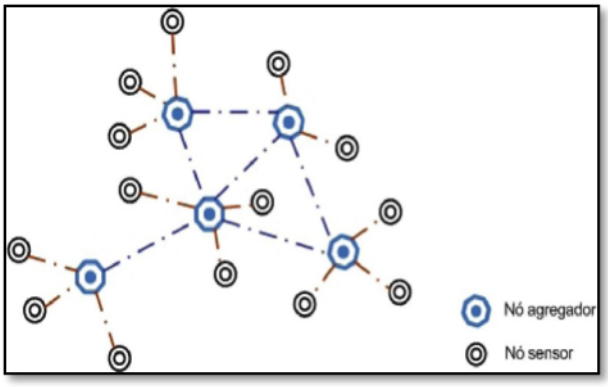


Figura A.3: Rede organizada segundo uma topologia hierárquica orientada a grupos.

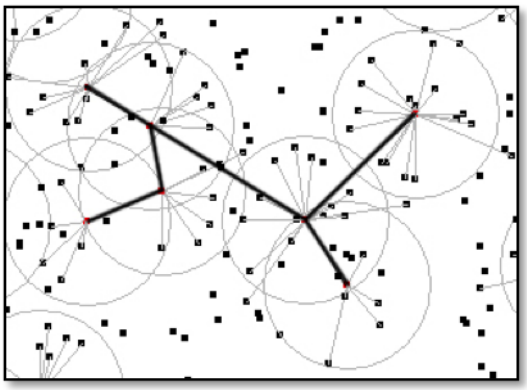


Figura A.4: Visão num simulador de uma rede orientada a grupos.

A.4.3 Suporte de comunicação

De acordo com requisitos de escala, das características específicas das aplicações e da topologia e organização da rede, poderão ocorrer diferentes padrões de comunicação. Um padrão habitual pode basear-se em difusão pura de um para N, habitualmente

com origem em nós especiais (ou na estação-base) com maior capacidade de energia, potência e alcance de comunicação. Outro padrão pode ser baseado em disseminação por reencaminhamento de dados nó a nó (encaminhamento par-a-par), sendo este o padrão mais adequado para topologias de cobertura em grande escala.

Dependendo dos requisitos aplicacionais e das diferentes topologias, o modelo de comunicação e os critérios para gestão eficiente dos recursos disponíveis podem ter que ser endereçados diferentemente para que sejam vantajosos para a aplicação. Assim, é vulgar que protocolos de nível MAC e encaminhamento, algoritmos de agregação ou processamento intermédio de dados, critérios de suporte de fiabilidade baseado na gestão de dados persistidos, e protocolos seguros de distribuição e estabelecimento de chaves criptográficas na rede, sejam concebidos de forma a adequarem-se minimamente às diferentes topologias.

A.4.4 Pilha de estruturação de serviços de *software* para RSSF

As arquitecturas de *software* para suporte de aplicações em RSSF são, em geral, estruturadas tendo em conta requisitos específicos de aplicações. A estruturação de serviços numa pilha de protocolos de rede segundo a norma IEEE 802.15.4 (subjacente às implementações usuais dos sensores vulgarmente utilizados para RSSF) não segue o mesmo modelo de uma rede convencional. A figura A.5 apresenta uma visão que corresponde a uma estrutura minimalista de suporte típico de uma aplicação de RSSF. Os requisitos das aplicações podem assim ser optimizados de forma mais pragmática como suporte do nível aplicação.

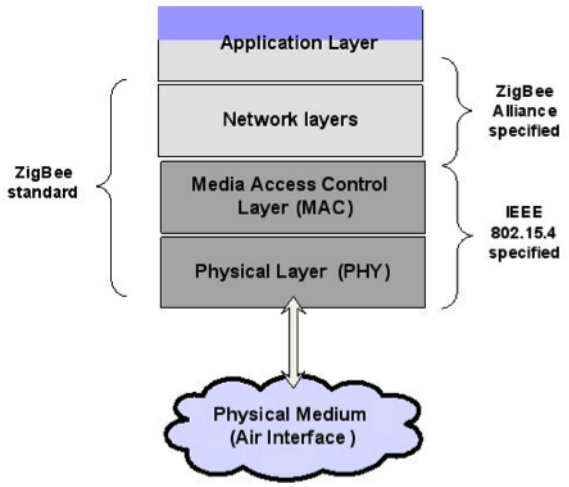


Figura A.5: Pilha de serviços de uma RSSF.

A.5 Aplicações das RSSF

Muitos desafios e aspectos que subsistem na realização de RSSF e suas aplicações permanecem em aberto. Estas redes têm sido ensaiadas, experimentadas ou exploradas para diversos tipos de aplicações. Algumas das aplicações da investigação são inovadoras, permitindo avaliar sobre a possibilidade das RSSF poderem constituir soluções únicas ou pelo menos mais económicas e funcionalmente mais vantajosas, quando comparadas com tecnologias de redes convencionais. As potenciais vantagens das RSSF tornam-se claramente visíveis em muitas dessas aplicações inovadoras, nomeadamente quando propostas para operarem em cenários de grande escala com auto-organização completa da rede.

Podemos tomar como exemplo diversas aplicações, tais como: aplicações na área da domótica para automação e monitorização de condições e vigilância de edifícios [SKG+05] ou de obras de engenharia civil [CFP+06], aplicações militares [HKL+06], controlo de subsistemas electromecânicos de automóveis [TZC06], monitorização de habitats naturais [PSM+04] e monitorização médica ou assistência remota e residencial de cuidados de saúde [MOJ06], entre muitas outras.

A.6 Problemática da segurança em RSSF

Muitas das características indicadas anteriormente implicam um acréscimo no risco de ataques à operação das RSSF, quando comparadas às redes convencionais. O não supervisionamento da rede pode afectar as condições de segurança, propiciando ataques que envolvam a captura física de nós. Admite-se que os sensores são dispositivos vulneráveis a ataques por intrusão (não apenas a tipologia de ataques a comunicações), permitindo a indução de comportamentos maliciosos que criam falhas ou ataques ao correcto funcionamento das redes. Isto alarga o leque de tipologias de ataques e potenciais hipóteses que podem ser exploradas por um adversário.

Além disso, os mecanismos e protocolos de segurança para RSSF têm que ser adequados às limitações e à capacidade de recursos existentes. Enquanto numa rede convencional o peso das computações relativas aos mecanismos de segurança e o volume de dados trocados durante os protocolos não são extremamente significativos, numa RSSF esse aspecto pode ser crucial, o que reduz de imediato uma série de hipóteses e soluções convencionais. Exemplo disso é o uso proibitivo de criptografia assimétrica, que é normalmente uma hipótese descartada, adoptando-se mecanismos que se baseiem somente em criptografia simétrica ou funções de sínteses seguras de mensagens.

Tal como nas redes *ad hoc* sem fios, a possibilidade de ocorrência de um ataque que se baseie num modelo de adversário Dolev-Yao [DY83] ou nas tipologias de ataques ao meio de comunicação inspiradas na *Framework* X.800 [Sta05] ou em modelos de hipóteses de adversário para ataques às comunicações na rede Internet [Shi00] continuam a verificar-se, sendo as condições de acesso ao meio de comunicação sem fios propícias a facilitar a vida ao adversário.

Enquanto nas redes *ad hoc* os dispositivos têm uma mínima vigilância por parte dos utilizadores, nas redes de sensores não há qualquer controlo, tornando a possibilidade de intrusão física em muitos cenários de aplicação uma realidade. Assim, considerando a estrutura minimalista da pilha de *software*, a tipologia de ataques a uma rede de sensores pode abarcar ataques por intrusão aos dispositivos, o que permitirá a um adversário ter condições mais favoráveis a quebrar as condições de segurança do dispositivo e ter acesso a eventuais segredos criptográficos instalados ou memorizados. Ataques podem ainda envolver formas de negação de serviço ao nível da comunicação (nível MAC), por indução de processamento incorrecto no protocolo de acesso ao meio. Consideram-se também ataques ao nível rede, incidindo neste caso em ataques ao encaminhamento, que podem ter em vista o controlo de parte significativa ou total da rede e da informação que nela flui ou que é processada.

Finalmente, considera-se que, devido às características das RSSF e seus dispositivos, um ataque com sucesso que consiga induzir falhas ou processamento incorrecto num nó da rede pode com maior ou menor facilidade ser replicado a uma vasta área da rede. Para tal, bastará que o adversário adquira, por exemplo, uma quantidade considerável de nós sensores (uma vez que possuem baixo custo), programe nesses sensores as condições de ataque e adicione esses nós em diferentes locais da rede, amplificando assim de forma simplificada o comportamento incorrecto e as condições do ataque em diferentes zonas da rede.