UNIVERSIDADE FEDERAL DO MARANHÃO
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE ELETRICIDADE

HIGO FELIPE SILVA PIRES

ABIDS-WSN: Um Framework de Detecção de Intrusão em Redes de Sensores sem Fio Orientado por Agentes Inteligentes

São Luís
2017

HIGO FELIPE SILVA PIRES

ABIDS-WSN: Um Framework de Detecção de Intrusão em Redes de Sensores sem Fio Orientado por Agentes Inteligentes

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia de Eletricidade da Universidade do Maranhão, para obtenção do título de Mestre em Engenharia de Eletricidade.

Área de concentração: Ciência da Computação

Orientador: Prof. Dr. Denivaldo Cicero Pavão Lopes

São Luís

2017

Ficha gerada por meio do SIGAA/Biblioteca com dados fornecidos pelo(a) autor(a).
Núcleo Integrado de Bibliotecas/UFMA

Pires, Higo Felipe Silva.
ABIDS-WSN : Um Framework de Detecção de Intrusão em Redes de Sensores sem Fio Orientado por Agentes Inteligentes / Higo Felipe Silva Pires. - 2017.
82 f.

Orientador(a): Denivaldo Cicero Pavão Lopes.
Dissertação (Mestrado) - Programa de Pós-graduação em Engenharia de Eletricidade/ccet, Universidade Federal do Maranhão, São Luís, 2017.

1. Agentes Inteligentes. 2. Redes de Sensores sem Fio. 3. Segurança da Informação. 4. Sistemas de Detecção de Intrusão. I. Lopes, Denivaldo Cicero Pavão. II. Título.

ABIDS-WSN: UM FRAMEWORK DE DETECÇÃO DE INTRUSÃO EM REDES DE SENSORES SEM FIO ORIENTADO POR AGENTES INTELIGENTES

Higo Felipe Silva Pires

Dissertação aprovada em 26 de janeiro de 2017.

Prof. Denivaldo Cicero Pavão Lopes, Dr.
(Orientador)

Profa. Karla Donato Fook, Dra.
(Membro da Banca Examinadora)

Prof. Osvaldo Ronald Saavedra Mendez, Dr.
(Membro da Banca Examinadora)

Ad Jesum per Mariam.

Agradecimentos

Em primeiro lugar, agradeço este trabalho a Deus, Nosso Senhor, que é o Senhor da História e foi – e é! – o Senhor da história deste trabalho desde sempre, me dando as graças necessárias para bem terminá-lo, por intercessão da Sua Santa Mãe, a Santíssima Virgem, e de toda a corte celeste.

À minha família, em especial meu pai Simião, minha mãe Rosa e meu irmão Arthur, que sempre estiveram comigo nesses anos, amorosa e pacientemente me incentivando nas alegrias e consolando nos momentos mais atribulados.

À minha amada Aline, que com seu amor e carinho nas boas horas e incentivo e compreensão nas horas difíceis foi fundamental nesse tempo de crescimento pessoal.

Um agradecimento especial ao meu orientador, o Professor Zair Abdelouahab (*in memoriam*), que nos deixou antes que esse trabalho viesse à luz. A ele meu eterno agradecimento pelas conversas, orientações, ensinamentos e conselhos e, sobretudo, por me ter servido de exemplo de Professor, ofício que, muito honrado, ora exerço. Seja a sua memória eterna!

Agradeço também ao Professor Denivaldo Lopes pela orientação após o falecimento do Professor Zair e pela aceitação dada ao trabalho. A ele, meus eternos agradecimentos pela compreensão, apontamentos, sugestões e conselhos para o aperfeiçoamento cada vez maior deste trabalho.

Aos colegas de Mestrado, em especial os amigos Breno, Débora, Silvano, Nilson e Tiago, por todos os momentos, conversas e ajuda mútua nos anos de Mestrado. A todos, meus mais sinceros votos de sucesso profissional e pessoal.

Aos colegas do LABSAC, em especial Mário Henrique, que foi de fundamental ajuda no processo de produção deste trabalho, Willian Ribeiro, Luan Oliveira e Natália Soeiro. A todos meus mais sinceros agradecimentos e votos de amizade. A cada um, meus reiterados agradecimentos e votos de amizade e sucesso.

Ao Instituto Federal de Educação, Ciência e Tecnologia do Maranhão - Campus Pinheiro, por, além de me prover o ambiente de trabalho, prover as condições tão favoráveis quanto possível para a consecução desse trabalho.

À Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (Capes), pela oferta das bolsas, que em muito ajudaram no sustento material necessário para a finalização do trabalho.

Aos professores e funcionários do Programa de Pós-Graduação em Engenharia de Eletricidade da Universidade Federal do Maranhão (PPGEE-UFMA), por toda a ajuda e apoio fornecidos no tempo do curso.

Aos amigos e irmãos de Fé da Associação Ad Maiorem Dei Gloriam, que com seu apoio espiritual, mediante preciosas orações, conseguiram do Céu esta Graça que agora se concretiza na minha vida.

Finalmente, meus agradecimentos a todos aqueles que, direta ou indiretamente, tornaram esse trabalho possível através da sua ajuda.

"Por isso aconselho a você, estudante, a não alegrar-se excessivamente por ler muitas coisas, mas por entender muitas coisas, e não somente entender mas poder memorizar. Do contrário, não adianta ler muito nem entender muito. Razão pela qual repito quanto disse acima, isto é, que as pessoas que se dedicam ao estudo necessitam de engenho e de memória."

Hugo de São Vítor (c. 1096 - 1141), A Arte de Ler

Resumo

PIRES, Higo Felipe Silva. **ABIDS-WSN: Um Framework de Detecção de Intrusão em Redes de Sensores sem Fio Orientado por Agentes Inteligentes**. 2017. 82 f. Dissertação (Mestrado em Engenharia de Eletricidade) – Programa de Pós-Graduação em Engenharia de Eletricidade, Universidade Federal do Maranhão, São Luís, 2017.

Ultimamente, houve um avanço significativo em várias tecnologias direta ou indiretamente correlatas à Computação Ubíqua. Entre elas, pode-se citar a tecnologia das Redes de Sensores sem Fio (*WSN*s). Tendo já o seu espaço no atual cenário, o uso dos sensores sem fio se estende em vários ramos da atividade humana: monitoramento industrial, *smart houses*, aplicações médicas e militares. Entretanto, várias deficiências e limitações em sensores sem fio podem ser notadas: recursos limitados de *hardware*, energia e capacidade computacional são pontos a sempre serem tratados por quem trabalha com tais dispositivos. Quanto a esses dispositivos há, além dos fatores já citados, uma preocupação importante referente à sua segurança. Assim como em outros dispositivos, para que essas ameaças sejam, ao menos, mitigadas é necessário criar camadas de segurança. Uma dessas camadas pode ser formada pelos Sistemas de Detecção de Intrusão (*IDS*). No entanto, devido à já mencionada restrição de *hardware* dos sensores, o desenvolvimento de *IDS*s — bem como qualquer outra aplicação — para esses dispositivos deve supor tais características. No que se refere, ainda, aos *IDS*s, há alguns aspectos que devem ser levados em conta, sobretudo flexibilidade, a eficiência e a capacidade de adaptação a novas situações. Um paradigma que facilita a implementação de tais capacidades são os Agentes Inteligentes. Sendo assim, este trabalho descreve a proposta de um *framework* para detecção de intrusões em *WSN*s baseado em agentes inteligentes.

Palavras-chaves: Segurança da Informação. Sistemas de Detecção de Intrusão. Redes de Sensores sem Fio. Agentes Inteligentes.

Abstract

PIRES, Higo Felipe Silva. **ABIDS-WSN: A Framework of Intrusion Detection in Wireless Sensor Networks Driven by Intelligent Agents**. 2017. 82 p. Dissertation (Master of Electricity Engineering) – Postgraduate Program in Electricity Engineering, Federal University of Maranhão, São Luís, 2017.

Lately, there has been a significant advance in several technologies directly or indirectly related to Ubiquitous Computing. Among them, the technology of Wireless Sensor Networks (WSNs) can be mentioned. Having its space in the current scenario, the use of wireless sensors extends into various branches of human activity: industrial monitoring, smart houses, medical and military applications. However, several shortcomings and limitations in wireless sensors can be noted: limited hardware, energy and computational capacity are points that are always treated by those who work with such devices. As for these devices, there is, besides the factors already mentioned, an important concern regarding their safety. As with other devices, for these threats to be at least mitigated, it is necessary to create layers of security. One of these layers may be formed by Intrusion Detection Systems (IDS). However, due to the aforementioned hardware restriction of the sensors, the development of IDSs - as well as any other application - for such devices should assume such characteristics. As for IDSs, there are some aspects that need to be taken into account, especially flexibility, efficiency and adaptability to new situations. A paradigm that facilitates the implementation of such capabilities is the Intelligent Agents. Therefore, this paper describes the proposition of a framework for intrusion detection in WSNs based on intelligent agents.

Keywords: Information security. Intrusion Detection Systems. Wireless Sensor Networks. Intelligent Agents.

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas e Siglas

CRC	*Cyclic Redundancy Check*
IDPS	*Intrusion Detection and Prevention System*
IDS	*Intrusion Detection System*
MAC	*Medium Access Control*
MANET	*Mobile ad hoc Network*
RFID	*Radio-Frequency IDentification*
VPN	*Virtual Private Network*
TDM	*Time Division Multiplexing*
WSN	*Wireless Sensor Network*

Sumário

1 Introdução

Neste primeiro capítulo, uma introdução sobre o trabalho será apresentada. Este primeiro capítulo trará trará uma contextualização e a problemática para a qual se visará propor uma solução, a motivação para tal trabalho, a solução proposta, os objetivos — gerais e específicos — do presente trabalho, a metodologia empregada para o desenvolvimento da solução proposta, finalizando com a apresentação da estrutura de todo o trabalho.

1.1 Contextualização e problemática

Com o advento da Computação Ubíqua, que foi pela primeira vez mencionada em (WEISER, 1991), a informática tornou-se cada vez mais presente nos ambientes nos quais estejam inseridos os seres humanos. Seja através de dispositivos vestíveis, dispositivos cientes do contexto ou sistemas embarcados, a presença desses dispositivos no meio humano está consolidada. Além disso, houve o surgimento da tecnologia dos sensores sem fio.

Sensores são dispositivos de tamanho reduzido, geralmente, dispostos em grandes quantidades (que variam das centenas aos milhares), cuja finalidade é captar informações do ambiente físico no qual estão inseridos, tais como temperatura, umidade, movimento, entre outras (XIE et al., 2011; BUTUN; MORGERA; SANKAR, 2014). No momento que estes dispositivos se localizam em uma dada região geográfica e possuem conexões entre si, tem-se uma rede de sensores sem fio (em inglês, *WSN*).

*WSN*s possuem uma ampla gama de aplicações: uso industrial, médico (*e-health*), militar, construção de *smart houses*, sensoriamento de eventos, *etc.* Segundo Patel e Aggarwal (2013), Xie et al. (2011), as *WSN*s possuem as seguintes características, que afetam de maneira decisiva o desempenho das *WSN*s e a maneira como elas devem ser trabalhadas:

- Pouca ou nenhuma infraestrutura;
- Nós sensores com recursos computacionais (memória, energia, largura de banda) limitados;
- Maior facilidade de comprometimento dos nós sensores;
- Topologia em constante mudança, devido a falhas ou mobilidade dos nós.

Por causa dos pontos elencados anteriormente, a segurança acaba se tornando um fator importante e complexo no uso das *WSN*s. Além de uma maior vulnerabilidade a dispositivos internos ou mesmo externos à rede, há também nos nós sensores uma vulnerabilidade contra a sua própria integridade física, dada a sua proximidade da fonte dos dados coletados. Portanto, faz-se necessário que sejam criadas medidas de segurança voltadas para amenização ou solução desses problemas de segurança. No entanto, devido ao menor poder computacional e energético dos sensores sem fio, o desenvolvimento de soluções de segurança para tais dispositivos deve ser diferenciado, lançando mão de técnicas e abordagens adequadas.

1.2 Motivação

A segurança é um fator vital em qualquer contexto onde haja uma rede de computadores. Visando lidar com esse fator, várias técnicas e abordagens foram desenvolvidas, sobretudo para redes cabeadas, porém essas abordagens, em geral, acabam por não ser tão funcionais em redes *wireless*, o que, por extensão, acaba por não ser tão funcional também em *WSN*s (BUTUN; MORGERA; SANKAR, 2014). Torna-se necessário, portanto, fazer uma abordagem compatível com as *WSN*s e os sensores sem fio que as compõem.

Segundo Khanum, Usman e Alwabel (2012), abordagens tradicionais, tais como criptografia, uso de *VPN*'s e *firewalls* acabam por se mostrar inadequadas para uso em *WSN*s, uma vez que elas provêm proteção apenas contra elementos externos à rede, além do grande consumo de recursos computacionais que elas demandam. Portanto, necessita-se de uma abordagem que seja, simultaneamente, eficiente o suficiente para a contenção em tempo real de atacantes não só externos à rede, mas também internos, e que demande menos recursos computacionais.

Para tanto, uma abordagem possível é o uso de sistemas de detecção de intrusão (*IDS*). Sistemas de detecção de intrusão podem ser definidos como sistemas que fazem uma detecção automática de invasões e ataques a uma rede, gerando alertas e relatórios do ocorrido (MOURABIT et al., 2014). Pelo fato de *IDS*s serem sistemas complexos, uma abordagem promissora é a decomposição dos módulos do sistema em vários nós da rede, tornando, assim, o *IDS* um sistema distribuído. Porém, também essa distribuição deve ser feita de maneira a não exaurir os escassos recursos computacionais dos sensores sem

fio. De maneira a se obter a consecução desse requisito, podem ser utilizados os agentes inteligentes.

Um agente é uma entidade computacional capaz de perceber o ambiente no qual esteja situada através de sensores, efetuar um processamento baseado na entrada provida pelos sensores e, de acordo com esse processamento, atuar no ambiente através de atuadores (NORVIG; RUSSELL, 2014). Quando os agentes trabalham em conjunto e de maneira coordenada – um sistema distribuído, portanto –, dá-se o paradigma dos sistemas multiagentes, que são sistemas que conseguem prover a flexibilidade, eficiência e adaptabilidade necessários para o bom funcionamento de um *IDS*.

Agentes são diferentes de outras abordagens, pois eles agem de maneira autônoma e baseada em objetivos. Além disso, eles apresentam as seguintes vantagens (MOURABIT et al., 2014):

- **Redução da latência de rede**: Agentes em um nó de rede, agindo de maneira autônoma, respondem e agem mais rapidamente do que se houvesse um coordenador central;
- **Execução autônoma**: Agentes continuam a atuar, mesmo com uma eventual perda ou danos na rede;
- **Atuação em ambientes heterogêneos**: Agentes são capazes de atuar em ambientes heterogêneos, criando, assim, um sistema multiplataforma;
- **Redução da carga de rede**: Em caso de necessidade, um agente poderá ter a capacidade de ser mover entre os *hosts* da rede, aumentando a distribuição de carga do sistema.

1.3 Hipótese e Objetivos

Dada a possibilidade do uso de agentes inteligentes para o desenvolvimento de sistemas de detecção de intrusão, auferida através de levantamento bibliográfico, este trabalho primeiramente buscará desenvolver a seguinte hipótese: *O uso de agentes inteligentes no desenvolvimento de soluções de detecção e prevenção de intrusões em redes de sensores sem fio é bastante adequado a tais equipamentos, pois aqueles favorecem o desenvolvimento de soluções cujo desempenho seja adequado para sensores sem fio. Além disso, o uso dos agentes inteligentes permite um melhor desenvolvimento desses sistemas, de maneira a*

prover desempenho computacional satisfatório, adaptabilidade, escalabilidade, menor tempo de latência e balanceamento de carga.

De maneira a comprovar a veracidade dessa hipótese, este trabalho dissertativo traça alguns objetivos gerais e específicos, descritos conforme a seguir.

O objetivo geral do presente trabalho de dissertação é propor um *framework*, baseado em agentes, de detecção de intrusão em redes de sensores sem fio, com a melhor adequação possível aos recursos dos dispositivos das redes.

Com vistas a auxiliar a consecução do objetivo geral, este trabalho possui os seguintes objetivos específicos:

- Conceber, projetar, implementar um *framework* baseado em agentes para detecção de intrusão em redes de sensores sem fio;
- Projetar e implementar um ambiente computacional a partir do *framework* proposto, que possibilite a construção de um cenário controlado;
- Projetar e implantar cenários de ataques simulados contra a rede de sensores sem fio implantada;
- Projetar e implantar cenários de medição, que possibilitem a aferição do consumo do hardware disponível para o framework;

1.4 Metodologia de Pesquisa

Para a consecução dos objetivos anteriormente citados, este trabalho lançará mão de alguns métodos, listados a seguir:

1. Revisão bibliográfica, feita através de levantamento de livros, artigos de jornal, anais de eventos e endereços *Web*;
2. Definição do escopo do problema, de maneira que sejam esclarecidos os limites do trabalho e sua posterior contribuição;
3. Levantamento dos requisitos necessários para o ambiente computacional construído a partir do framework proposto;
4. Levantamento dos requisitos necessários para os cenários de ataques simulados no ambiente computacional controlado;
5. Levantamento dos requisitos necessários para desenvolvimento e correta aplicação dos testes de medição.

1.5 Estrutura do Trabalho

A escrita deste trabalho distribui-se ao longo de seis capítulos, os quais serão detidamente descritos a seguir:

- **Capítulo 2**: Este capítulo apresenta um contexto tecnológico, o qual traz uma abordagem dos fundamentos teóricos cujo conhecimento é necessário para a compreensão deste trabalho. Neste capítulo, os conceitos referentes a Redes de Sensores sem Fio, Ameaças e Contramedidas em Redes de Sensores sem Fio, Sistemas de Detecção de Intrusão e Agentes Inteligentes são abordados. Por último uma síntese do capítulo é apresentada;
- **Capítulo 3**: Neste capítulo, um estado da arte é apresentado, ou seja, faz-se uma apresentação e uma discussão dos mais importantes e recentes trabalhos referentes aos conceitos discutidos no Capítulo 2, sendo feita posteriormente uma comparação entre aqueles que foram considerados mais importantes para este trabalho. Ao final do capítulo, uma síntese é apresentada;
- **Capítulo 4**: Neste capítulo, a proposta deste trabalho dissertativo é apresentada: um *Framework* de Detecção de Intrusão em *WSN*s. A arquitetura da proposta, sua modelagem e s sua prototipagem, ou seja, a exposição de um cenário de funcionamento do *framework* serão apresentados. Ao fim do capítulo, uma síntese é apresentada;
- **Capítulo 5**: Neste capítulo, as avaliações e discussões acerca dos testes feitos junto ao cenário de funcionamento do *framework* são apresentadas, assim como o ambiente e os dados usados nos testes, os resultados dos testes e comparações com os trabalhos relacionados. Por último uma síntese dos testes é apresentada;
- **Capítulo 6**: Neste capítulo, as conclusões do trabalho dissertativo serão apresentadas, bem como uma discussão sobre os objetivos alcançados, eventuais limitações apresentadas pelo *framework*, publicações acadêmicas referentes ao trabalho e sugestões de trabalhos futuros.

2 Contexto Tecnológico

Neste capítulo, um contexto tecnológico será apresentado, no qual serão abordados os principais conceitos que norteiam este trabalho, a saber: redes de sensores sem fio, ameaças e contramedidas em redes de sensores sem fio, detecção de intrusão, com enfoque em detecção de intrusão em redes de sensores sem fio e agentes inteligentes e seu uso em detecção de intrusão.

2.1 Redes de Sensores sem Fio

No contexto da Internet das Coisas, vários dispositivos que possuem, entre outras capacidades, funções de sensoriamento são apresentados: *tags RFID*, *GPS*, sensores infravermelho, *scanners laser*, entre vários outros. Um desses dispositivos apresentados são os sensores sem fio. Quando tais sensores atuam em conjunto, concretiza-se uma rede de sensores sem fio (LI et al., 2014).

Uma rede de sensores sem fio é composta por desde uma dezena ou uma dúzia até milhares de nós autônomos, espacialmente distribuídos cuja principal função (sem se resumir a ela) é monitorar e coletar informações físicas e/ou ambientais. Tal funcionamento se dá, em geral, através do envio dos dados por parte dos nós periféricos para os nós centrais (GANDHIMATHI; MURUGABOOPATHI, 2016). Pelo fato de tais sensores possuírem baixo custo e fácil capacidade de propagação e instalação, são amplamente utilizados com vários fins: médicos, militares, industriais, civis, etc. Um exemplo é uma rede de sensores localizada em uma fábrica; tais sensores são responsáveis por detectar presença de pessoal, eventos, materiais e condições perigosos à integridade dos funcionários, etc. Vê-se por esse exemplo, portanto, que a segurança é fator primordial em uma rede de sensores sem fio.

Uma arquitetura básica de redes sem fio possui os seguintes componentes (WANKHADE; CHAVHAN, 2013):

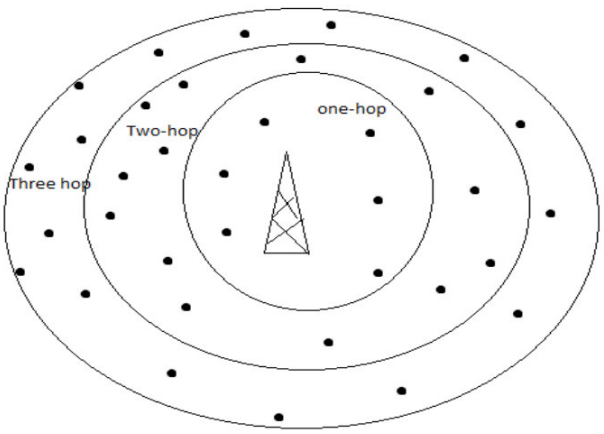
- **Nó sensor**: São os nós responsáveis pelo sensoriamento dos dados do ambiente. Podem também repassar dados para outros nós da rede;
- **Nó coletor**: São os nós responsáveis pela coleta e armazenamento dos dados absorvidos pelos nós sensores;

- ***Cluster head***: Em uma arquitetura de *cluster* (exposta a seguir), um nó *cluster head* é responsável por receber os dados dos nós sensores, o qual envia para um nó coletor.

Redes de sensores sem fio podem ser divididas em três categorias de arquitetura (WANKHADE; CHAVHAN, 2013):

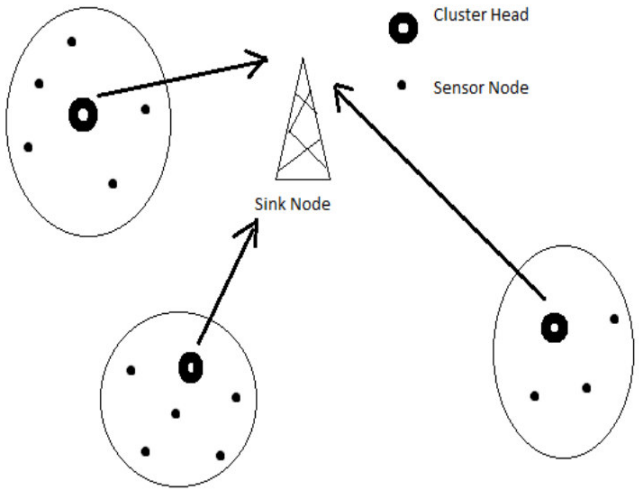
- **WSN em arquitetura de camadas**: Arquitetura de rede onde os nós são separados por camadas, separadas entre si por *hops* (saltos de rede). Nelas só há um nó coletor. Esta arquitetura é construída conforme na Figura 1;
- **WSN em arquitetura de *cluster***: Arquitetura composta por aglomerados (*clusters*) de sensores, cada um comandado por um nó *cluster head*, que repassa os dados para um nó coletor que age como estação-base. Tal arquitetura se dá como exposto na Figura 2;
- **WSN com nó coletor móvel**: Nesta arquitetura, o nó coletor se desloca ao longo da rede, para coletar os dados aferidos pelos nós sensores. Na Figura 3 é possível ver essa arquitetura.

Figura 1 – WSN em arquitetura de camadas



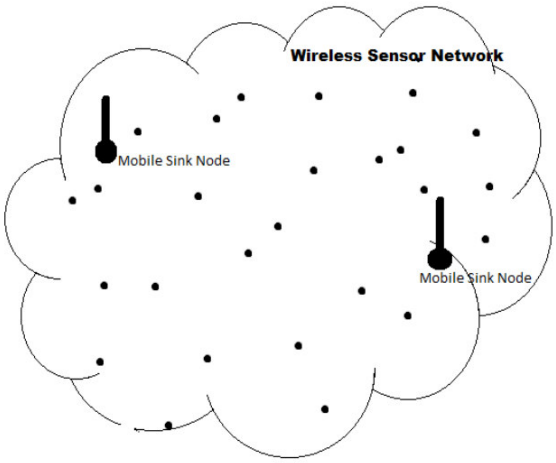
Fonte: (WANKHADE; CHAVHAN, 2013)

Figura 2 – WSN em arquitetura de *cluster*



Fonte: (WANKHADE; CHAVHAN, 2013)

Figura 3 – WSN com nó coletor móvel



Fonte: (WANKHADE; CHAVHAN, 2013)

2.1.1 Segurança em Redes de Sensores sem Fio

Sensores sem fio possuem importantes restrições quanto aos seus recursos: a capacidade de processamento, armazenamento e energia desses dispositivos é menor do que em computadores *desktop* ou outros dispositivos móveis. Além disso, os sensores estão distribuídos ao longo de uma ampla área. Portanto, a rede deve ser flexível, confiável, segura, organizada e tolerante a falhas (CAN; SAHINGOZ, 2015). Tais restrições exigem mecanismos de segurança robustos e tornam a segurança um ponto ainda mais relevante nas redes de sensores sem fio.

A princípio, existem dois tipos de abordagem quanto à segurança: abordagem estática e abordagem dinâmica. Pode-se dizer que exemplos de abordagem estática são as *VPN*'s, *firewalls*, métodos de autenticação e criptografia. Tais abordagens compõem uma primeira linha de defesa contra eventuais atacantes. No entanto, tais abordagens mais tradicionais acabam não sendo adequadas para uso em *WSN*'s, uma vez que as referidas técnicas só proveem defesa contra defesas externas, além de serem por natureza, grandes consumidoras de recursos (KHANUM; USMAN; ALWABEL, 2012). Além disso, geralmente os dados trafegam abertamente em *WSN*'s o que pode causar comprometimentos na segurança (LI et al., 2014).

2.2 Ameaças e Contramedidas em Redes de Sensores sem Fio

Como exposto anteriormente, a segurança é importante preocupação quanto às redes de sensores sem fio. Pela potencial importância de uma rede de sensores, atacantes podem lançar mão de vários tipos de ataques citados na literatura, contra os quais podem ter tomadas várias contramedidas possíveis.

Há vários tipos de ataques contra redes de sensores sem fio. Entre os mais conhecidos e citados na literatura, podemos citar:

- ***Jamming***: Neste ataque, os atacantes buscam causar interferência nos canais de sinal sem fio usados pelos nós (JAN; KHAN, 2013).
- ***Tampering***: Ataque que resulta em acesso físico ao sensor por parte de um atacante, com o intento de recolher dados ou descobrir artefatos como chaves de criptografia (MESSAI, 2014);
- **Colisão**: Neste tipo de ataque, o atacante faz com que haja uma colisão em um dos octetos do *frame* na rede, gerando a perda deste e forçando a sua posterior retransmissão. Outro tipo de estrutura que pode ser vitimada por um ataque de colisão é o pacote *ACK*, gerando atrasos exponenciais na rede (JAN; KHAN, 2013);
- **Exaustão**: Ataque que pode ser tanto feito por um atacante externo quanto por um sensor interno comprometido; nesse ataque, o sensor, através de um código malicioso, faz várias requisições à rede, consumindo seus recursos de hardware e energia (JAN; KHAN, 2013);

- ***Black hole***: Ataque no qual um nó comprometido da rede falsifica as diretivas de roteamento da rede, de maneira a forçar a passagem dos demais pacotes da rede por aquele nó e não repassá-los aos seus destinatários, criando um "buraco negro" na rede (MESSAI, 2014);
- ***Selective forwarding***: Ataque no qual um nó malicioso da rede atua como um roteador, não repassando certos tipos de mensagens (MESSAI, 2014);
- ***Sybil***: Ataque no qual um dispositivo malicioso adota, de maneira ilegítima, várias identidades. É um ataque que possui grande efetividade, sobretudo em ludibriar algoritmos de roteamento e alocação de recursos (WALTERS et al., 2007);
- ***HELLO Flood***: O pacote *HELLO* é bastante usado por protocolos de roteamento. Sua principal função é descobrir os nós vizinhos na rede e estabelecer uma topologia da rede. A versão mais simples desse ataque consiste em inundar a rede com pacotes *HELLO*, de maneira a impedir a troca de mensagens (MESSAI, 2014);
- ***Wormhole***: Em um ataque *wormhole*, é criado um túnel de baixa latência entre dois pontos da rede, pelo qual o atacante recebe mensagens e as retransmite para uma parte diferente (KARLOF; WAGNER, 2003);
- ***Sinkhole***: Neste ataque, um atacante entra em acordo com um nó da rede ou introduz um falso nó na rede e o usa para levar a cabo um ataque. Após essa manobra, o atacante ouve requisições de rotas e tenta convencer os nós vítimas de que o nó comprometido/nó falso possui o caminho mais rápido até a estação-base da rede (CAN; SAHINGOZ, 2015);
- ***Spoofed, altered and Replay***: Neste ataque, os pacotes dos protocolos de roteamento são alterados, modificados (gerando perda do princípio da integridade) ou mesmo reenviados, causando envio de pacotes maliciosos para a estação-base, *loops* indesejados na rede, aumento da latência, geração de falsas mensagens de erro e atração ou expulsão de tráfego de rede (JAN; KHAN, 2013).
- ***Homing***: É um tipo de ataque que busca mirar suas atividades em nós mais especiais, que tenham tarefas mais nobres como gerenciar chaves criptográficas ou monitorar o tráfego da rede (JAN; KHAN, 2013);
- **Negação de serviço**: Ataques por negação de serviço visam sobrecarregar um sistema para torná-lo indisponível. Em sistemas com sensores sem fio, que possuem, por natureza, menor capacidade de processamento, isso acaba por se tornar um ponto ainda mais sensível (CAN; SAHINGOZ, 2015). Ataques de negação de serviço em

geral lançam mão dos protocolos TCP, UDP e ICMP. Alguns exemplos de ataques de negação de serviço:

- *SYN Flood*: ataque no qual o atacante lança mão do processo de *three-way handshake* para enviar à vítima uma inundação de pacotes *TCP* com a *flag SYN*, de maneira a não tornar possível a resposta *SYN-ACK*, sobrecarregando, assim, a vítima (RAO et al., 2014).
- *LAND*: ataque no qual é enviado um pacote *TCP* com a *flag SYN* para uma porta aberta com mesmos endereços e portas de origem e destino (UNDERSTANDING. . . , 2015).
- *ICMP Flood*: Ataque no qual são enviados vários pacotes *ICMP* "*echo request*" para a vítima, sobrecarregando seus recursos (DEGIRMENCIOGLU et al., 2016).
- *Smurf*: Ataque no qual são enviados vários pacotes *ICMP* "*echo request*" para o *broadcast*, o qual envia as respostas para toda a rede (SURISETTY; KUMAR, 2010).
- *UDP Flood*: São enviados vários pacotes *UDP* à vítima, de maneira a diminuir seu desempenho e causar congestionamento na rede (ZARGAR; KABIRI, 2009).

Assim como ocorre com os ataques, também, na literatura, várias contramedidas que visem coibir as ações maliciosas são propostas. Contra o ataque *black hole*, é muito útil o uso de medidas estatísticas para a detecção de ataques (XIE et al., 2011). Em Bysani e Turuk (2011) são apresentados dois esquemas que visam combater o *selective forwarding*: um que visa detectar os nós maliciosos e removê-los das tabelas de roteamento (seja através de detecção baseada em reconhecimento, seja baseada em informações dos nós vizinhos) e outro que vise apenas mitigar os ataques. Em Can e Sahingoz (2015) são apresentadas duas soluções para o *HELLO flood*: verificação bidirecional de *links* e uso de múltiplas estações-base. Contra o *wormhole*, Hu, Perrig e Johnson (2003) propõem um mecanismo que restringe a vida útil de um pacote qualquer, restringindo, como consequência, a distância que tal pacote pode viajar e Triki, Rekhis e Boudriga (2009) apresentam um mecanismo que preconiza que haja nós investigadores na rede, monitorando a topologia da rede e os datagramas passantes. Em Dallas, Leckie e Ramamohanarao (2007) é proposto um sistema de detecção de anomalias, que tem por principal funcionalidade analisar a magnitude de contagem de saltos em uma tabela de roteamento, em busca de padrões que

demonstrem a presença de um ataque *sinkhole*. Em Newsome et al. (2004) são apresentadas duas abordagens para conter os efeitos do ataque *Sybil*: a validação direta, na qual um nó testa diretamente a validade da identidade de outro nó e a validação indireta, na qual nós já validados também podem participar da validação de nós cuja identidade ainda não foi certificada.

2.3 Sistemas de Detecção de Intrusão

Segundo Scarfone e Mell (2007), detecção de intrusão é o processo de monitoramento de eventos em um computador ou em uma rede de computadores orientado a uma busca por violações a sistema, ameaças às políticas de segurança do sistema ou práticas de segurança já padronizadas. Tais eventos geram alertas e relatórios, que serão enviados ao administrador de rede. Um sistema de detecção de intrusão é, portanto, um sistema computacional que torna automático o processo de detecção de intrusão.

De acordo com Srivastava et al. (2011), os IDS, quanto à cobertura dos eventos, se dividem em duas categorias:

- ***Host-based Intrusion Detection Systems* (*HIDS*)**: HIDS são projetados para detectar intrusões em apenas uma máquina na rede;
- ***Network-based Intrusion Detection Systems* (*NIDS*)**: NIDS visam a detecção de intrusão em toda uma rede de computadores.

Outra classificação importante é feita quanto à forma de detecção, que pode ser feita também em duas categorias (SRIVASTAVA et al., 2011):

- **IDS baseado em assinaturas**: Também chamado de **IDS baseado em abusos**, nele é armazenado um catálogo com padrões dos ataques contra os quais se deseja proteger. Caso a rede apresente um padrão condizente com alguma assinatura, o alerta de ataque é gerado. É bastante funcional contra ataques conhecidos, porém não é tão efetivo contra ataques contra os quais não há assinatura pré-definida;
- **IDS baseado em anomalias**: Eficiente em identificar novos padrões de ataques através de treinamento e comparações entre padrões esperados e padrões desviantes (SCARFONE; MELL, 2007), porém às vezes é falho em descobrir ataques de assinatura mais conhecida, pela ausência de um banco de assinaturas;

Ainda segundo Scarfone e Mell (2007), um *IDS* possui basicamente quatro componentes:

- **Sensor ou agente**: Responsável direto pela análise do tráfego de rede;
- **Servidor de gerenciamento**: Dispositivo responsável por gerenciar os sensores e receber as informações por eles geradas. Alguns destes servidores são capazes de interpretar as informações recebidas pelos sensores e identificar a presença de eventos na rede;
- **Servidor de banco de dados**: Trata-se de um repositório das informações úteis (informações de pacotes de rede ou histórico de eventos, por exemplo) para o IDS;
- **Console**: Interface entre o IDS e o administrador de rede, provendo a possibilidade de monitoramento do IDS, bem como dos eventos e informações por ele geridos;

2.3.1 Detecção de Intrusão em Redes de Sensores sem Fio

Os IDS apresentam várias técnicas para detecção e contenção dos ataques, porém várias vezes esses IDS são projetados para atuar em meios como redes cabeadas ou ambientes de computação em nuvem, tornando-se, portanto, de aplicação inadequada em uma rede de sensores sem fio (CAN; SAHINGOZ, 2015). Tal adequação é a principal motivação para detecção de intrusão em WSN's, e não se aplica mesmo às redes sem fio ou às MANETS, de topologia mais semelhante às WSN's.

Além do exposto acima, outra importante motivação para o desenvolvimento de IDS's é o volume de ataques já catalogados e apresentados em diversos trabalhos, como o exposto por na Tabela 1.

Portanto, segundo Can e Sahingoz (2015), é necessário desenvolver os IDS em conformidade com as restrições apresentadas pelas WSN's, devido às suas peculiaridades: bateria limitada, ambiente aberto, memória e capacidade computacional limitadas.

De acordo com Rassam, Maarof e Zainal (2012), especificamente em WSN's, os IDS's representam uma segunda camada de defesa contra ataques, uma vez que a primeira camada, composta por mecanismos de criptografia e autenticação, não consegue conter ataques vindos de dentro da rede.

Para o desenvolvimento dos IDS's, são usadas várias técnicas, apresentadas na literatura; abordagens baseadas em assinaturas e/ou baseadas em anomalias combina-

Tabela 1 – Taxonomia dos ataques contra redes de sensores sem fio

Taxonomia	Tipos de ataque
Ataques de comunicação	Ataques de repetição
	Ataques de negação de serviço
	Ataques Sybil
Ataques contra a privacidade	Espionagem
	Representação
	Análise de tráfego
Ataques com nós específicos como alvo	Captura de nó sensor
	Destruição de nó sensor
Ataques por consumo de energia	
Ataques contra políticas	
Ataques contra gerenciamento de chaves criptográficas	

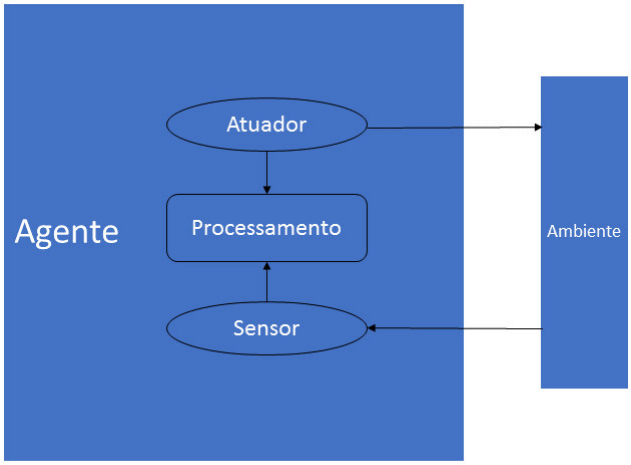
Fonte: (HAN et al., 2006)

das com uso de técnicas como: Agentes Inteligentes, Mineração de Dados, Inteligência Computacional, Teoria dos Jogos, detecção baseada em estatísticas, etc.

2.4 Agentes Inteligentes

Segundo Norvig e Russell (2014), um agente inteligente (também chamado de agente racional) é qualquer entidade que seja, através de sensores, capaz de perceber o ambiente no qual esteja inserido e, baseado nesse sensoriamento, agir nesse ambiente através de atuadores, conforme pode ser ilustrado na Figura 4. Agentes podem incorporar a si várias capacidades, como tomadas de decisão baseadas em lógica fuzzy ou aprendizado de máquina ou mesmo mobilidade entre nós em uma rede. São usados nos mais variados campos da sociedade: Inteligência Artificial, Robótica, Psicologia, Sociologia, Linguística e muitos outros.

Figura 4 – Estrutura de um agente inteligente e fluxograma de suas relações com o ambiente



Fonte: Adaptado de (NORVIG; RUSSELL, 2014)

Segundo Weiss (1999), as arquiteturas de agentes podem ser categorizadas em 4 tipos:

- **Agentes baseados em lógica**: agentes nos quais as decisões são tomadas baseadas em deduções lógicas;
- **Agentes reativos**: agentes nos quais as tomadas de decisão são implementadas com base em mapeamentos feitos de situação para ação;
- **Agentes baseados em crença-desejo-intenção**: agentes cujas decisões são tomadas através da manipulação de estruturas de dados que representam as crenças, desejos e intenções dos agentes;
- **Agentes baseados em camadas de arquiteturas**: agentes cujas decisões são tomadas com base em camadas de software, cada uma com maior ou menor abstração e conhecimento do ambiente computacional no qual se encontra o agente.

Quanto aos tipos de agente, Norvig e Russell (2014) apresentam quatro tipos, a saber:

- **Agentes reativos simples**: Agente que executa as suas ações baseadas na sua percepção atual, em um processamento baseado em condição-ação, independente das percepções anteriores;
- **Agentes reativos baseados em modelos**: Agente que possui um estado interno que é baseado em um histórico de percepções, bem como um modelo do ambiente no

qual ele se encontra. Essas estruturas fornecerão um panorama do ambiente para o agente, influenciando, portanto, as suas ações nele;

- **Agentes baseados em objetivos**: Agente que pauta as suas ações não apenas e um modelo e um histórico de percepções, mas também em objetivos (situações desejáveis);
- **Agentes baseados na utilidade**: Agente orientado, além das ações, modelos e percepções, por utilidade, que nada mais é senão uma medida de maior ou menor satisfação de certas métricas.

Quando vários agentes compartilham o mesmo ambiente computacional, trabalham em conjunto e trocam mensagens e informações, dá-se a presença de um sistema multi-agentes. Os agentes que compõem um sistema multi-agentes podem ser cientes da situação dos outros agentes do sistema, o que permite que os agentes se comportem de uma tal maneira que pareçam módulos de um maior sistema (PANAIT; LUKE, 2005).

2.4.1 Uso de Agentes Inteligentes em Detecção de Intrusão

No que se refere aos sistemas de detecção de intrusão, os agentes inteligentes permitem que sejam melhor desenvolvidos, uma vez que o processo de detecção de intrusão deve ser ágil, de maneira a minimizar os danos, e escalável, para que seja possível trabalhar com cargas de trabalho variáveis. Além disso, eles fornecem as seguintes vantagens a estes sistemas (PATIL et al., 2008; MOURABIT et al., 2014):

- **Redução da latência de rede**: Agentes localizados em um determinado host, no qual alguma ação deva ser tomada, têm uma resposta mais rápida do que ocorreria caso no sistema fosse usada alguma entidade centralizada;
- **Execução autônoma**: Agentes são mais independentes; eles permanecem funcionais, mesmo após uma falha parcial do sistema, o que fornece uma maior tolerância a falhas;
- **Ambiente heterogêneo**: Plataformas de agentes permitem que o sistema opere nas mais diversas plataformas, fornecendo uma camada independente de sistema operacional;

- **Redução de carga de rede**: Agentes, caso seja necessário, podem implementar mobilidade de todo o processo para outros hosts da rede, permitindo, assim, uma distribuição da carga do sistema;
- **Adaptação dinâmica**: pela natureza dinâmica dos comportamentos de um agente, um sistema de detecção de intrusão baseado em agentes pode ser reconfigurado em tempo de execução;
- **Adaptação estática**: Quando for necessário adicionar uma nova assinatura de ataque ao IDS, os algoritmos do sistema de detecção podem ser atualizados sem a reinicialização do sistema inteiro;
- **Escalabilidade**: Caso seja necessário, é possível distribuir os agentes por toda a extensão de um sistema, permitindo que a carga do sistema seja distribuída em vários hosts.

2.5 Síntese

Neste capítulo, os principais conceitos introdutórios foram apresentados. Estes são necessários para uma correta compreensão e implementação do *framework* apresentado neste trabalho dissertativo: Redes de Sensores sem Fio, Detecção de Intrusão e Agentes Inteligentes.

Os sensores sem fio possuem várias particularidades, principalmente quanto às suas capacidades computacionais (memória, processamento, quantidade de armazenamento disponível, carga de bateria, entre outros). Além disso, conforme exposto neste capítulo, possui tamanho razoável o rol de ameaças que visam atacar o reto funcionamento dessas estruturas. Portanto, faz-se necessário que haja implementações de soluções de segurança que levem em conta tais particularidades.

Visando atender tais particularidades, várias abordagens foram propostas na literatura. Uma delas, citada nesse trabalho e escolhida como abordagem principal deste trabalho, é a dos Agentes Inteligentes. Tal escolha se deve a algumas vantagens já citadas e descritas neste capítulo, tais como: desempenho computacional satisfatório, adaptabilidade, escalabilidade, menor tempo de latência e balanceamento de carga.

3 Estado da Arte

Este capítulo apresenta os trabalhos propostos pela comunidade científica cujos estudos foram considerados bastante relevantes para a consecução do framework proposto.

3.1 Ameaças e Contramedidas em Redes de Sensores sem Fio

Com a expansão do uso das redes de sensores sem fio, aumentou, paralelamente, a quantidade de ameaças que visam atentar contra a segurança dessas redes. Com esse aumento indesejado, surgiu a necessidade de encontrar meios de combater tais ameaças. Com efeito, a literatura oferece bastantes trabalhos sobre tais ameaças e contramedidas.

Um importante trabalho no início desses esforços foi o trabalho apresentado por Han et al. (2006) – cuja taxonomia já foi apresentada neste trabalho, conforme a Tabela 1 –, embora careça de uma apresentação das contramedidas apropriadas para cada tipo dos ataques citados. Trabalhos posteriores, no entanto, se encarregaram de tal tarefa.

Malik (2013) primeiramente apresenta os principais requisitos de segurança em WSN's: confidencialidade dos dados entre emissário e receptor das mensagens, de maneira a proteger dados sensíveis do acesso de pessoas indevidas; integridade dos dados, de maneira a providenciar que a integridade dos dados não seja afetada por atacantes; autenticação dos dados, com o fim de restringir as atividades efetuadas por nós não autorizados; atualidade dos dados, ou seja, uma garantia de que os dados serão os mais recentes possíveis, para impedir que mensagens antigas não sejam reaproveitadas; disponibilidade, importante requisito referente não só às capacidades de funcionamento da rede, mas também aos gastos computacionais da mesma; auto-organização, que se refere à adaptabilidade e flexibilidade da topologia dos nós; localização segura, que provê referências geográficas para melhoramento dos protocolos de segurança. Na Tabela 4 são apresentados os ataques e contramedidas propostos neste trabalho.

Semelhante trabalho é apresentado por Panigrahi, Sharma e Ghose (2013). Além de citar as ameaças mencionadas pelo trabalho anterior, são citadas outras contramedidas, tais como: gerenciamento de chaves, com o fim de estabelecer comunicações mais seguras com uma chave válida entre os pares envolvidos na conexão e IDS's, visando identificar e combater intrusos.

Tabela 2 – Ataques e contramedidas propostos por Malik (2013)

Tipo de ataque	**Contramedidas propostas**
Jamming	Uso de múltiplas frequências
Ataques físicos em geral	Proteção física extra
	Camuflagem
	Escondimento
Colisões	Detecção e evitação de colisões
	Uso de CRC
Exaustão	Uso de TDM
Inequidade	Uso de pacotes de pequeno tamanho
Negligência e ganância	Definição de rotas alternativas
Homing	Criptografia do cabeçalho e do conteúdo
Spoofing de informações de roteamento	Uso de esquemas CRC ou MAC
Black hole	Troca de pacotes de roteamento apenas entre nós autorizados
	Criptografia de chave pública
Flooding	Limitar número de conexões
	Uso de captchas
Desincronização	Autenticação das partes críticas
Sybil	Limitação das conexões estabelecidas pelos sensores
Selective forwarding	Monitoramento regular da rede, em busca de atividade suspeita

No trabalho desenvolvido por Jan e Khan (2013), são apresentados especificamente os ataques de negação de serviço (DoS) e suas respectivas contramedidas, no escopo das WSN's. Segundo o autor, os ataques de negação de serviço se dividem em três classes:

- Ataques na Camada Física
  - *Jamming*;
  - *Tampering*;
- Ataques na Camada MAC
  - Colisão;
  - Exaustão;
- Ataques na Camada de Rede
  - *Spoofed, altered and Replay*;
  - *Sybil*;
  - *Sinkhole*;
  - *Black hole*;
  - *Worm hole*;
  - *Selective forwarding*;
  - *Homing.*

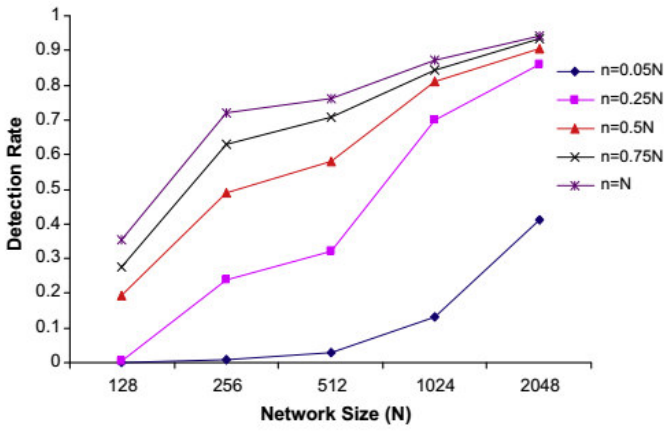
3.2 Sistemas de Detecção de Intrusão em Redes de Sensores sem Fio

No escopo das redes de sensores sem fio, várias abordagens de detecção de intrusão foram feitas. A seguir são apresentadas algumas delas.

Em Li, He e Fu (2008) é proposto um esquema no qual os nós de uma rede de sensores são divididos em grupos. Nesses grupos, os sensores ficam fisicamente próximos uns dos outros, para maior precisão da detecção das intrusões, e os atacantes são detectados através dos valores atribuídos a certas variáveis, tais como: dados coletados pelos sensores, taxa de envio de pacotes, taxa de perda de pacotes, taxa de pacotes nos quais foi detectado algum tipo de alteração, taxa de recebimento de pacotes e energia dispendida no envio de pacotes. No trabalho é concluído que tal abordagem é capaz de detectar com sucesso os ataques com uma baixa taxa de falsos positivos e baixo consumo de energia.

Em Baig (2011) é apresentado um modelo para prevenção de ataques baseada em fluxo de tráfego de rede, além da definição de um mecanismo distribuído para a detecção dos mesmos ataques. Para tanto, são especificados modelos, específicos para cada topologia de rede, de tráfego normal de rede. A partir do ciência do que seja um tráfego normal, são feitas, de maneira bem facilitada, as detecções do tráfego dito anormal. Para testes da efetividade e performance do sistema, foi elaborada uma simulação do sistema, mediante o uso de um ataque de exaustão distribuído. Como resultado o sistema logrou sucesso na detecção dos ataques (com taxas de detecção entre 86 a 92%), sem um maior overhead, tão custoso a dispositivos de poucos recursos computacionais. Foi observada também uma maior taxa de detecções em redes de sensores menos densas, conforme a Figura 5.

Figura 5 – Taxa de detecção de ataques em função do tamanho/densidade da rede (BAIG, 2011)



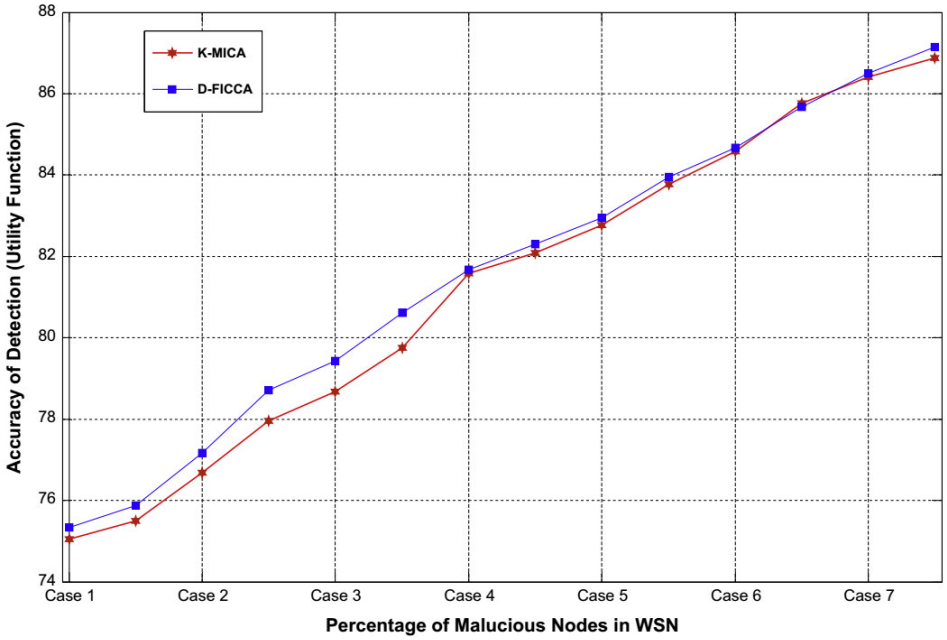
Fonte: (BAIG, 2011)

Em Moon, Kim e Cho (2014) são apresentados esquemas de métodos de roteamento eficientes energeticamente, bem como de detecção de intrusão, cujo princípio é o baixo overhead. O funcionamento da proposta ocorre ao longo de três fases: construção inicial, na qual as tabelas de roteamento de toda a rede são construídas; sensoriamento da transmissão de dados, na qual um nó sensor gera e retransmite relatórios de eventos para a estação-base; reconstrução, na qual a topologia de rede e as tabelas de roteamento são reconstruídos, após a sinalização de um alerta. Concluiu-se, ao fim do trabalho, que o objetivo de alcançar baixo overhead foi alcançado.

No trabalho desenvolvido por Wang et al. (2011), é apresentado um mecanismo integrado de detecção de intrusão voltado para clusters de sensores sem fio. Tal sistema integrado agrega três subsistemas principais: *Intelligent Hybrid Intrusion Detection System* (*IHIDS*), *Hybrid Intrusion Detection System* (*HIDS*) e IDS baseado em abusos. Tais componentes são aplicados de maneiras diferentes em cada um dos diferentes níveis da WSN (desde as cluester heads até os nós sensores). Com o fim de mitigar a quantidade de falsos positivos, é também agregado um módulo de detecção através de abusos e outro de detecção através de anomalias. Acoplado a todos estes elementos, um módulo de tomada de decisão para integrar os ataques detectados por cada um dos módulos separadamente.

Em Shamshirband et al. (2014a), é apresentada uma proposta de um algoritmo que lança mão da lógica fuzzy para efetuar detecção de intrusão em clusters redes de sensores sem fio, de maneira a aprimorar a precisão da detecção de atividade maliciosa. Para a avaliação do sistema, foi usado um dataset com dados reais, recolhidos por sensores do Intel Berkeley Research Lab, e sua perfomance foi comparada com outros algoritmos semelhantes já consagrados, como o K-MICA, o K-mean e o DBSCAN, demonstrando uma precisão na detecção acima dos 87%, conforme a Figura 6.

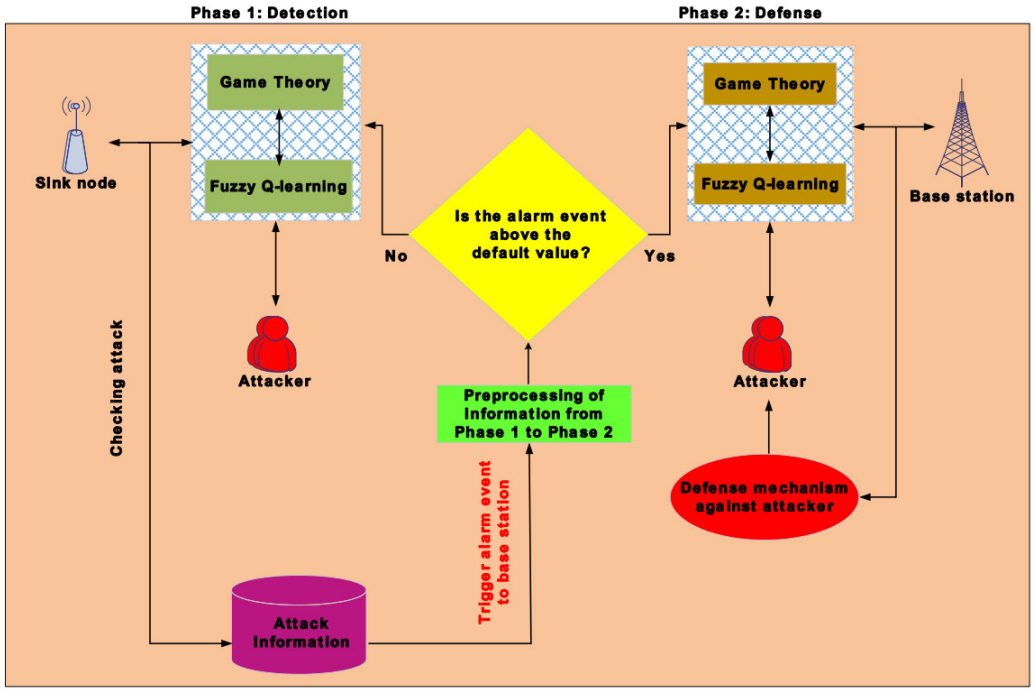
Figura 6 – Comparação da precisão entre os algoritmos K-MICA e D-FICCA



Fonte: (SHAMSHIRBAND et al., 2014a)

Em Shamshirband et al. (2014b), é apresentado um modelo de detecção e prevenção de intrusão (principalmente ataques de negação de serviço distribuída) em WSN's baseado em Teoria dos Jogos. Para avaliação, o modelo é comparado com outros modelos considerados leves e, portanto, adequados para uso em WSN's, tais como: controlador de lógica fuzzy *NS-2*, *Q-learning* e *fuzzy Q-learning*. Nos resultados, o modelo proposto demonstrou melhor precisão na detecção, tomada de contramedidas, tempo de vida da rede e precisão da defesa. Na Figura 7, é apresentado o modelo do IDPS; nele, uma estrutura composta por detecção baseada em Teoria do Jogos e aprendizado de máquina é responsável pela detecção dos ataques. Neste mecanismo, a estação-base e os nós coletores se adaptam, através de aprendizado baseado em *fuzzy Q-learning* (*FQL*), para selecionar a melhor estratégia e, a partir desta, tomar as contramedidas necessárias. Depois dessa etapa, o nó coletor transmite um alarme à estação-base, informando a ocorrência do ataque. A partir daí, na segunda fase, a estação-base usa o algoritmo *FQL* como meio de defesa contra o atacante.

Figura 7 – Atacante e modelagem do IDPS baseado em Teoria dos Jogos



Fonte: (SHAMSHIRBAND et al., 2014b)

No trabalho proposto por Ngai, Liu e Lyu (2007), é proposto um algoritmo para detecção do ataque *sinkhole* (escolhido por sua importante seriedade e potenciais danos causados a uma WSN). O algoritmo proposto busca um conjunto de nós suspeitos e constrói uma lista deles, através de recursos de checagem de consistência de dados. A partir daí, mediante análise do fluxo de rede, são identificados os reais nós nocivos à rede. Além disso, o algoritmo goza de robustez suficiente para lidar também com nós que, agindo de maneira cooperativa, mascaram a presença do verdadeiro intruso. Os resultados demonstraram a eficiência e a eficácia esperados na detecção dos ataques, além da presença de um overhead baixo o suficiente para uma WSN.

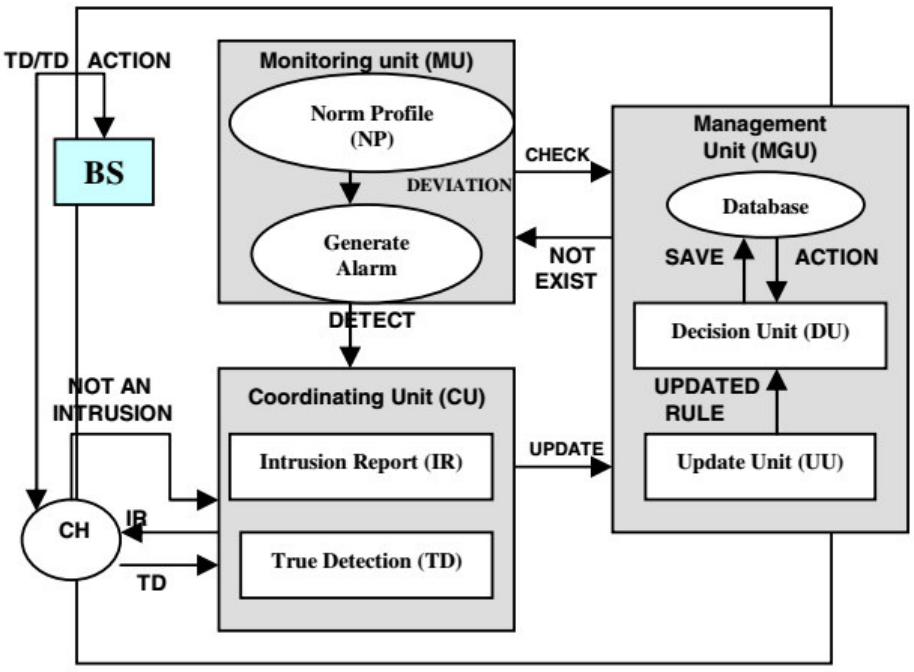
3.3 Uso de Agentes Inteligentes em Sistemas de Detecção de Intrusão

Na presente seção, serão apresentados os trabalhos que, quer pelas suas limitações, quer pelas aberturas detectadas para melhoramentos ou abordagens alternativas, foram considerados basilares para a elaboração do presente trabalho. Ao fim, será apresentada uma tabela comparativa entre os trabalhos citados na seção.

3.3.1 Energy-Efficient Intrusion Detection System for Wireless Sensor Network Based on MUSK Architecture

No trabalho de Khanum et al. (2010) é apresentada a arquitetura *MUSK*, um mecanismo que visa dinamismo e segurança de redes de sensores sem fio que faça uso otimizado do uso de energia. Essa arquitetura em baseada em três agentes: *Monitoring Unit*, *Management Unit* e *Coordinating Unit*. O agente *Monitoring Unit* tem como responsabilidade monitorar as intrusões: quando ele detecta alguma intrusão, é enviada uma mensagem para o agente *Management Unit*, onde será aferido se aquele padrão de ataque informado existe no banco de dados; se não, é enviada uma mensagem de volta ao *Monitoring Unit*, informando a não existência do padrão no repositório. Se existe, o *Management Unit* envia uma mensagem ao *Coordinating Unit* para que seja trabalhada a terminação da intrusão. É, por fim, informado no trabalho que a principal vantagem obtida no trabalho é redução da carga de trabalho. Na Figura 8 é apresentada uma ilustração da arquitetura apresentada pelo trabalho.

Figura 8 – Arquitetura de agentes MUSK

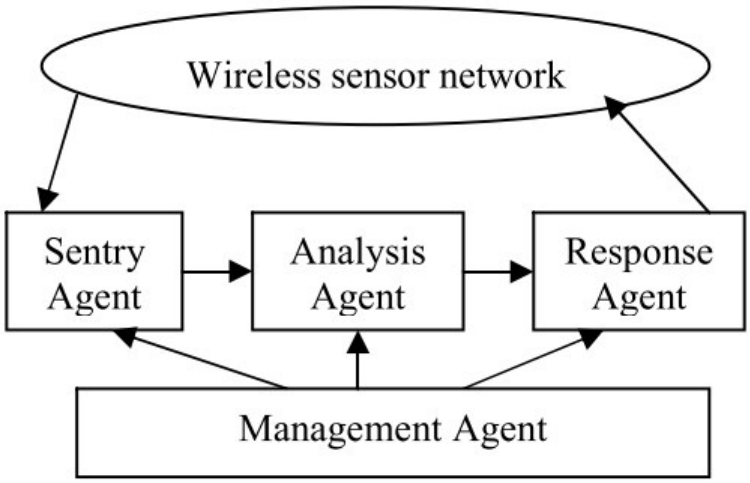


Fonte: (KHANUM et al., 2010)

3.3.2 Intrusion Detection for Wireless Sensor Networks Based on Multi-agent and Refined Clustering

Neste trabalho (WANG; YUAN; WANG, 2009) é apresentado um modelo de sistema de detecção de intrusão baseado em multi-agentes que usa um novo método de detecção, chamado clusterização refinada. Para a consecução desse método são usados algoritmos de redes neurais e algoritmos de classificação. Este trabalho se baseia em quatro agentes: Sentry Agent, Analysis Agent, Response Agent e Management Agent. O Sentry Agent é alocado em cada um dos nós da rede de sensores sem fio, e é responsável por monitorar as atividades nos nós. Os dados coletados por ele são submetidos para análise pelo Analysis Agent, que é o responsável por julgar se há ou não intrusão, baseado no que for eventualmente recebido pelo Sentry Agent. O agente Response Agent, localizado em cada cluster de sensores, é responsável por receber os resultados das análises feitas no Analysis Agent e tomar as contramedidas necessárias, baseadas no que foi recebido do Analysis Agent. O Management Agent é instalado em cada um dos nós de cada cluster de sensores. Cada um desses nós toma parte no gerenciamento da rede, tornando, assim, a administração da rede descentralizada (evitando, assim, o colapso da rede inteira, caso o nó centralizado falhe); as tarefas do Management Agent envolvem gerenciar, manter e harmonizar os demais agentes. Tal arquitetura demonstrou uma ótima taxa de detecção, ainda que também uma maior taxa de falsos positivos. A Figura 9 retrata um diagrama com os relacionamentos entre os agentes apresentados.

Figura 9 – Relacionamento entre os agentes Sentry, Analysis, Response e Management

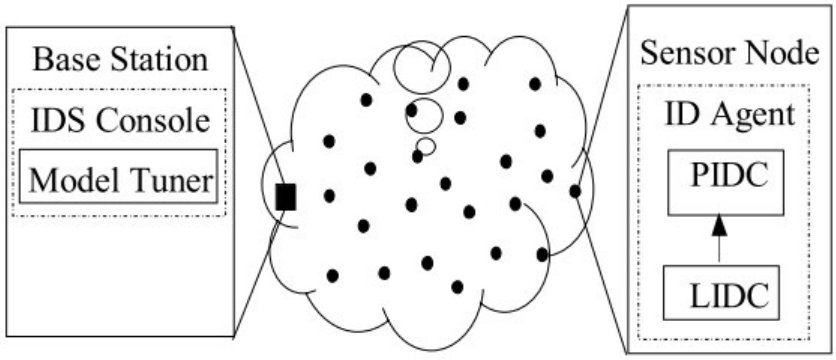


Fonte: (WANG; YUAN; WANG, 2009)

3.3.3 A Framework of Machine Learning Based Intrusion Detection for Wireless Sensor Networks

Aqui é apresentado (YU; TSAI, 2008) um *framework* de sistema de detecção de intrusão em redes de sensores sem fio baseado em aprendizado de máquina. Nesse framework é proposto um agente: o *Intrusion Detection Agent*, o qual é executado em cada um dos nós da rede de sensores. O agente usa algoritmos de aprendizado de máquina para, baseado em aprendizado anterior, aferir as características dos pacotes e detectar ou não os ataques. O sistema proposto é capaz de monitorar todos os nós sensores, porém apenas um por slot de tempo. Na Figura 10 encontra-se uma representação do agente responsável pelo *framework* proposto.

Figura 10 – Diagrama representado o agente Intrusion Detection



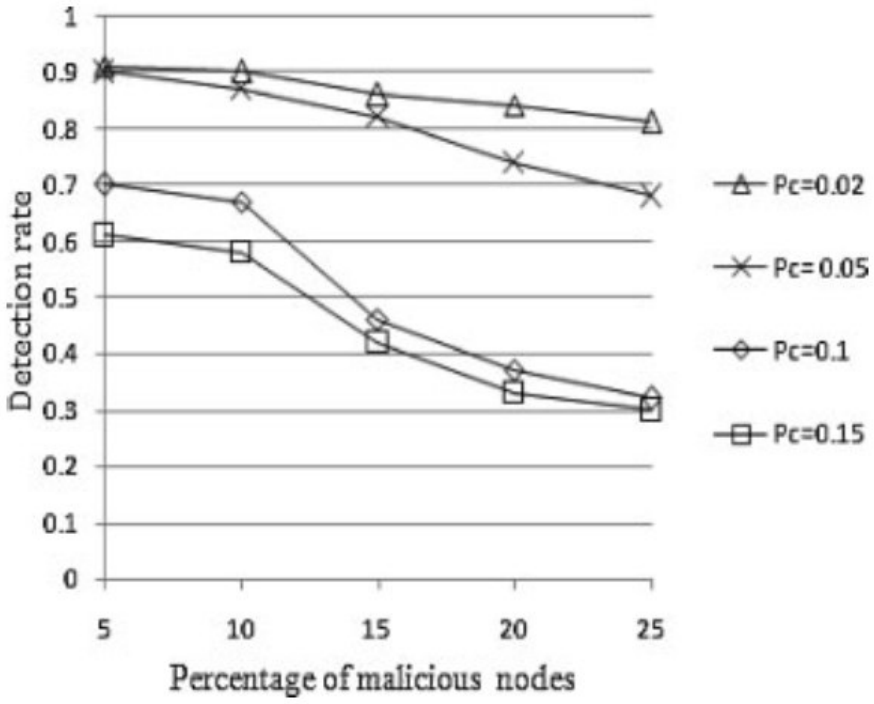
Fonte: (YU; TSAI, 2008)

3.3.4 A lightweight intrusion detection framework for wireless sensor networks

Neste trabalho (HAI; HUH; JO, 2010) é proposto um *framework* leve (consumo otimizado de energia) integrado para redes de sensores clusterizados. Tal *framework* é composto por dois agentes: *Local IDS Agent* e *Global IDS Agent*. Uma característica importante do quesito economia de energia é que desses agentes, só um fica ativo por vez, para que haja economia do uso de bateria e demais recursos limitados dos sensores. O *Local IDS Agent* é responsável por monitorar as informações enviadas e recebidas pelos sensores da rede. Quando a rede é inicialmente configurada, os sensores não possuem qualquer conhecimento sobre nós maliciosos, conhecimento este que vai se consolidando de maneira gradual ao longo do tempo de vida do sistema, e sendo propagado para os outros nós sensores da rede. Já o *Global IDS Agent* é responsável por monitorar a comunicação

estabelecida nos nós vizinhos a ele. Nesse agente é feita a investigação propriamente dita dos pacotes, de maneira a buscar as intrusões. Quanto à performance desta proposta, foi concluído que o número de nós monitores da rede de sensores é inversamente proporcional à possibilidade de surgirem falsos positivos: quanto maior o número de nós monitores na rede, menor a possibilidade de serem alertados falsos positivos. Por outro lado, foi detectado que quando o número de nós monitores aumenta, aumenta também a possibilidade de emissão de falsos negativos. Na Figura 11 é possível observar um gráfico que demonstra essa tendência.

Figura 11 – Taxa de detecção de falsos positivos no framework proposto por Hai, Huh e Jo (2010)



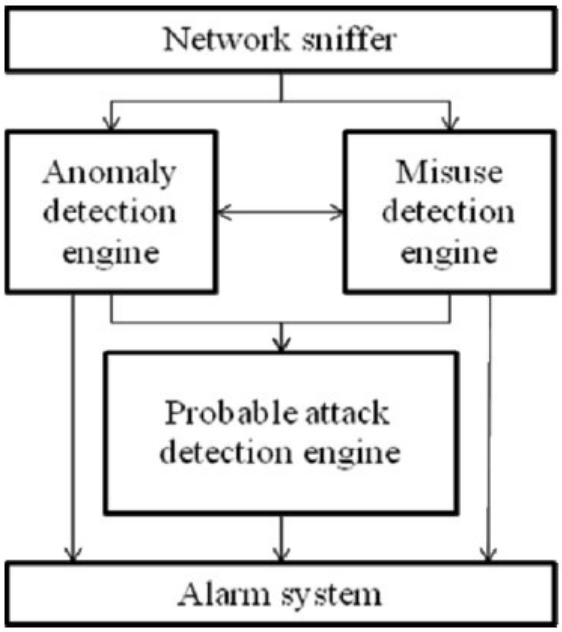
Fonte: (HAI; HUH; JO, 2010)

3.3.5 Wireless Intrusion Detection System Using a Lightweight Agent

Neste trabalho (HADDADI; SARRAM, 2010) é proposto um sistema de detecção de intrusão orientado por um agente, o IDS Agent. Esse agente é dividido em cinco módulos, a saber: Network sniffer, responsável por "farejar" a rede; *Misuse detection engine*, responsável por investigar se certos padrões de tráfego são semelhantes a padrões de ataques; *Anomaly detection engine*, que analisa os pacotes e investiga se há anomalias estabelecidas em uma dada regra (pacotes de rede suspeitos acima de um número limite,

por exemplo); *Probable attack detection*, que tem como função investigar se está havendo um ataque ou não, dadas as informações coletadas; *Alarm system*, que entra em ação caso uma anomalia ou ataque sejam detectados, informando o administrador através de um simples recurso de log. Nos testes, o sistema foi bem-sucedido na detecção de ataques como descoberta de rede, representação falsa de nós de rede, ataque do homem no meio e negação de serviço, porém não detectou ataques eavesdropping. Na Figura 12 é possível observar uma imagem que retrata o sistema proposto.

Figura 12 – Sistema de agentes proposto por Haddadi e Sarram (2010)



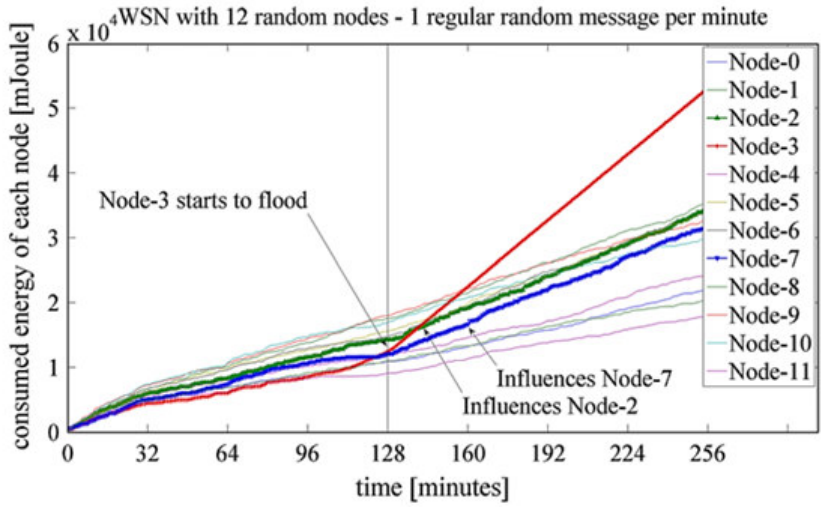
Fonte: (HADDADI; SARRAM, 2010)

3.3.6 Lightweight energy consumption-based intrusion detection system for wireless sensor networks

Neste trabalho (RIECKER et al., 2015) é proposto um IDS baseado em agentes móveis capaz de detector intrusões baseando-se no consumo de energia dos sensores (conforme apresentado no gráfico na Figura 13), uma vez que ataques perturbam o consumo de energia normal nos nós. O IDS é formado por apenas um agente móvel. O sistema proposto se demonstrou confiável na detecção das intrusões, porém há nele algumas limitações: não adequação a todo e qualquer cenário de uso de WSN's, mudança do código do agente por um atacante que tenha acesso físico aos nós, diminuição de aplicabilidade

causada por baixa quantidade de armazenamento e, por fim, ataques que não alterem os níveis de consumo de energia permanecem indetectáveis.

Figura 13 – Medidas de energia em uma WSN com 12 nós arranjados em topologia mesh

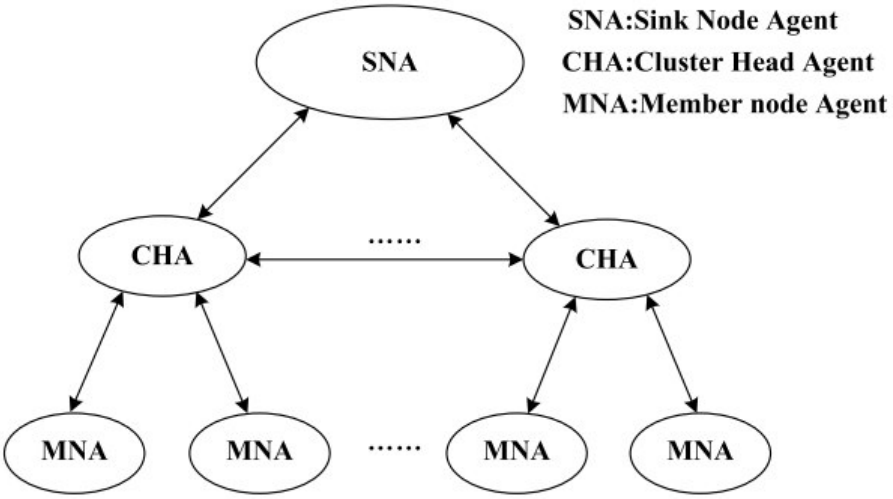


Fonte: (RIECKER et al., 2015)

3.3.7 Agent-based intrusion detection and self-recovery system for wireless sensor networks

Aqui é apresentado (SUN; LIU, 2013) um modelo com três agentes: *Member Node Agent* (*MNA*), responsável por monitorar, coletar e analisar dados e *Cluster Head Agent* (*CHA*), responsável por tomar decisões baseadas na análise feita pelo *MNA*, decisões estas que são executadas pelo *Sink Node Agent* (*SNA*). A principal vantagem do uso de agentes por parte do modelo é a adaptação dinâmica do sistema e a otimização dos recursos de sistema. Nos resultados, é possível notar que a efetividade do IDS, porém não sem ter como retorno uma alta taxa de dados transmitidos e energia consumida. Na Figura 14 abaixo, é exposto o modelo proposto pelos autores.

Figura 14 – Modelo de detecção de intrusão proposto por Sun e Liu (2013)



Fonte: (SUN; LIU, 2013)

Segue abaixo, na Tabela 3 com os trabalhos citados na atual seção:

Tabela 3 – Tabela comparativa de principais trabalhos relacionados

Abordagem	**N.A**	**A.B.An**	**A.B.Ab**	**NIDS**	**HIDS**	**T.M.A**	**U.D.R**
(KHANUM et al., 2010)	3	✓		✓		✓	✓
(WANG; YUAN; WANG, 2009)	4		✓	✓		✓	✓
(YU; TSAI, 2008)	1	✓			✓		✓
(HAI; HUH; JO, 2010)	2	✓		✓			
(HADDADI; SARRAM, 2010)	1	✓	✓	✓			✓
(RIECKER et al., 2015)	1		✓	✓			
(SUN; LIU, 2013)	3	✓		✓		✓	
ABIDS-WSN (Solução proposta)	4		✓	✓		✓	✓

Abaixo, uma legenda sobre os termos abreviados usados na Tabela 3:

- **N.A**: Número de agentes;
- **A.B.An**: Abordagem baseada em anomalias;
- **A.B.Ab**: Abordagem baseada em abusos;
- **NIDS**: *Network-based Intrusion Detection*;
- **HIDS**: *Host-based Intrusion Detection*;
- **T.M.A**: Troca de mensagens entre os agentes;
- **U.D.R**: Uso de dispositivos reais;

A proposta do ABIDS-WSN vislumbra primeiramente oferecer um *framework* leve computacionalmente para detecção em redes de sensores sem fio e que, além disso, torne possível que o sistema de detecção das intrusões tenha baixo tempo de latência e

balanceamento de carga. O ABIDS-WSN visa também atacar algumas limitações aferidas nos trabalhos relacionados, sobretudo taxas reduzidas de verdadeiros-positivos e alto consumo de energia, através da proposição de um sistema com distribuição de carga (o que acaba por proporcionar consumo de energia reduzido em cada um dos nós) e uso de detecção orientada por assinaturas obtidas através de cenários de testes.

3.4 Síntese

Neste capítulo, alguns trabalhos encontrados na literatura sobre as Redes de sensores sem Fio, principais ataques e contramedidas foram apresentados, além de abordagens diversas de detecção e/ou prevenção de intrusão em *WSN*s e, principalmente abordagens de IDSs voltados para *WSN*s construídos através de agentes inteligentes.

Quanto a estas últimas, mais importantes por sua correlação mais próxima com este trabalho dissertativo, uma tabela comparativa foi apresentada, com uma exposição das suas principais características, bem como as características do atual trabalho. Além disso, as conclusões fornecidas pelos próprios autores foram apresentadas, bem como as limitações dos mesmos trabalhos. O conhecimento de tais resultados e limitações forneceu um importante insumo para o atual trabalho dissertativo.

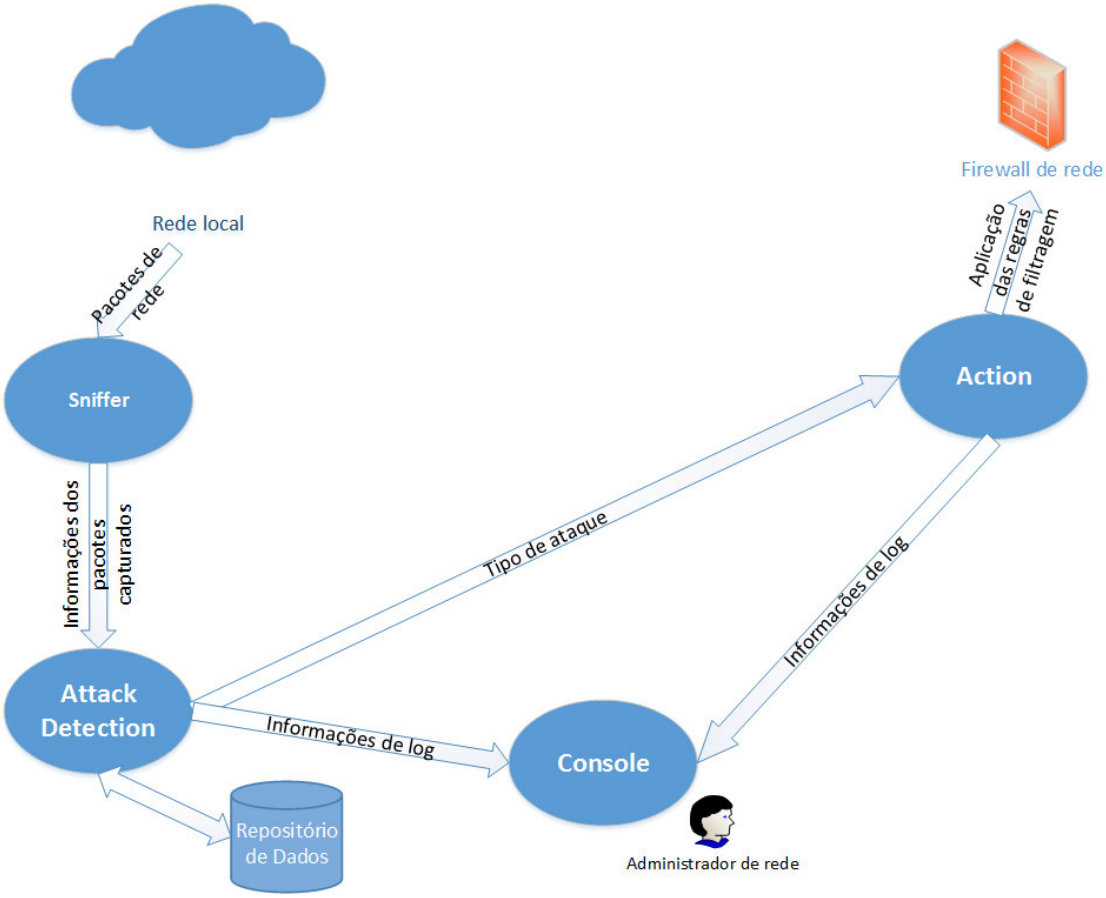
4 Proposta de um Framework de Detecção de Intrusão em WSNs

Neste capítulo, a proposta do *Framework* para Detecção de Intrusão em Redes de Sensores sem Fio será apresentada, sendo daqui em diante chamado de **ABIDS-WSN** (*Agent-based Intrusion Detection System for Wireless Sensor Networks*). A arquitetura do *framework*, a modelagem dos requisitos do sistema, bem como sua prototipagem serão apresentados. Por fim, este capítulo será encerrado como uma síntese.

4.1 Arquitetura

A arquitetura do ABIDS-WSN é composta por 4 agentes inteligentes: *Sniffer*, *Attack Detection*, *Action* e *Console*; todos foram desenvolvidos através do JAVA Agent DEvelopment Framework (JADE)[1], uma plataforma *open-source* para desenvolvimento de sistemas multi-agentes, e sua representação é como a demonstrada na Figura 15:

Figura 15 – Arquitetura do Framework ABIDS-WSN



Fonte: O Autor (2017)

[1] http://jade.tilab.com/

O *framework*, orientado por abusos, possui um repositório de dados, hospedado junto ao agente *Attack Detection*, e no qual serão armazenadas as assinaturas dos ataques que ele visará combater e as informações dos pacotes capturados. Neste repositório, podem ser inseridas quantas assinaturas sejam consideradas necessárias; além disso, não há um foco definido para um tipo de ataque e, portanto, podem ser inseridas diversas assinaturas de quaisquer tipos de ataques contra os quais se queira tomar contramedidas, o que acaba por favorecer uma maior diversidade dos tipos de ataque que o *framework* irá combater.

O agente *Sniffer* é responsável pela captura dos pacotes da rede; o agente *Attack Detection* será responsável pela detecção dos endereços IP pertencentes à rede e pela análise do tráfego capturado pelo agente *Sniffer*, em busca de padrões que correspondam às assinaturas armazenadas pelo ABIDS-WSN; *Action* será o agente responsável por receber os alertas de detecção do agente *Attack Detection* e tomar as medidas necessárias para a contenção das ameaças e *Console*, responsável por receber informes dos agentes *Attack Detection* e *Action*, e emiti-los ao administrador de rede.

Como primeira etapa do funcionamento do IDS, o agente *Attack Detection* faz uma varredura na rede, em busca dos endereços IP pertencentes à rede. Tendo cumprido essa tarefa, o agente *Sniffer* se dedica à captura dos pacotes; tal operação é subsidiada pela biblioteca Jpcap[2], através de funções de captura e filtragem de pacotes. Tal captura se dá por iterações, de maneira a permitir a cada dado número de segundos a ação do agente *Attack Detection*. Durante cada uma dessas iterações, informações de cada um dos pacotes capturados são armazenadas em um banco de dados.

Após cada iteração de funcionamento do agente *Sniffer* – que agora está em estado de espera –, entra em ação o agente *Attack Detection*. Munido com as informações do que pode ou não ser um ataque, ele busca no banco de dados povoado pelo agente *Sniffer* elementos que apontem para a presença de um ou mais ataques (inclusive ataques que tenham ocorrido de maneira simultânea): caso não haja detecção, conclui-se que há tráfego normal e os registros são apagados (o que acaba por ser também uma medida de otimização do armazenamento, que, no caso das WSN's, é um recurso também escasso); caso haja a detecção de ao menos um ataque por parte do *Attack Detection*, é enviada uma mensagem *ACL* (*Agent Communication Language*, protocolo de envio de mensagens assíncronas usado pelo JADE e baseado no padrão *FIPA*[3]) para o agente *Action*, informando detalhes sobre

[2] http://jpcap.gitspot.com/index.html
[3] http://www.fipa.org/

os ataques concretizados, e outra mensagem ACL para o agente Console, para permitir que o administrador de rede seja informado do evento. Além disso, é adicionado no banco de dados uma ocorrência relativa aos ataques.

Quando o agente *Action* recebe a mensagem *ACL*, ele implementa regras de filtragem de pacotes no *firewall* de rede, as quais são específicas para cada tipo de ataque. Tais regras visam bloquear o tráfego invasor, bem como as máquinas que executam os ataques. Assim como no agente *Attack Detection*, os relatos desses eventos são também enviados para o agente *Console*. Depois do envio desta mensagem, reinicia-se o ciclo, e são novamente identificados os endereços IP da rede.

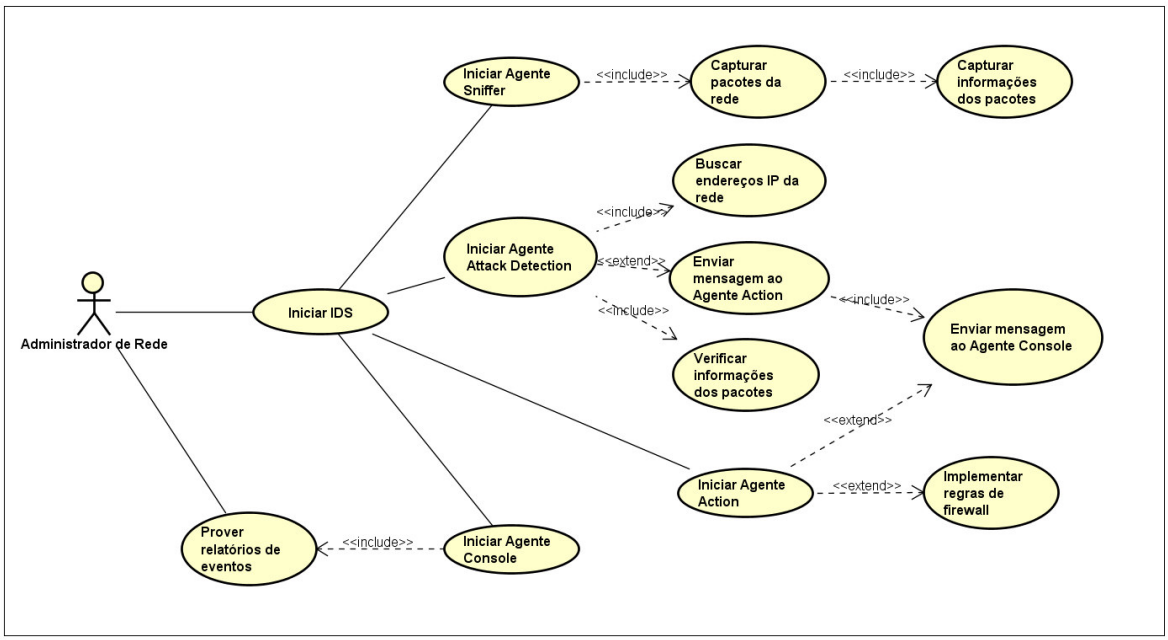
4.2 Modelagem

O projeto do ABIDS-WSN pode ser formalizado em alguns tipos de diagramas UML, como os que serão apresentados nas próximas subseções.

4.2.1 Diagrama de Caso de Uso

O processo de funcionamento do *framework*, descrito na seção anterior, pode ser formalizado de acordo com o diagrama de caso de uso apresentado na Figura 16:

Figura 16 – Digrama de Caso de Uso do Framework ABIDS-WSN



Fonte: O Autor (2017)

O administrador de rede dá início ao funcionamento do IDS; nesse instante são inicializados os quatro agentes, simultaneamente. O agente *Sniffer* e o agente *Attack Detection* dão início às suas duas atribuições: a busca pelos endereços IP da rede e a captura dos pacotes passantes na rede. Este último passo conduz à ação de armazenar as informações dos pacotes.

No momento que a captura dos pacotes é temporariamente suspensa, tem início a verificação do cabeçalho dos pacotes. Em caso de detecção, é enviada uma mensagem para o agente *Action*, com as informações dos ataques, além de uma mensagem para o agente *Console*, com um relatório da ocorrência.

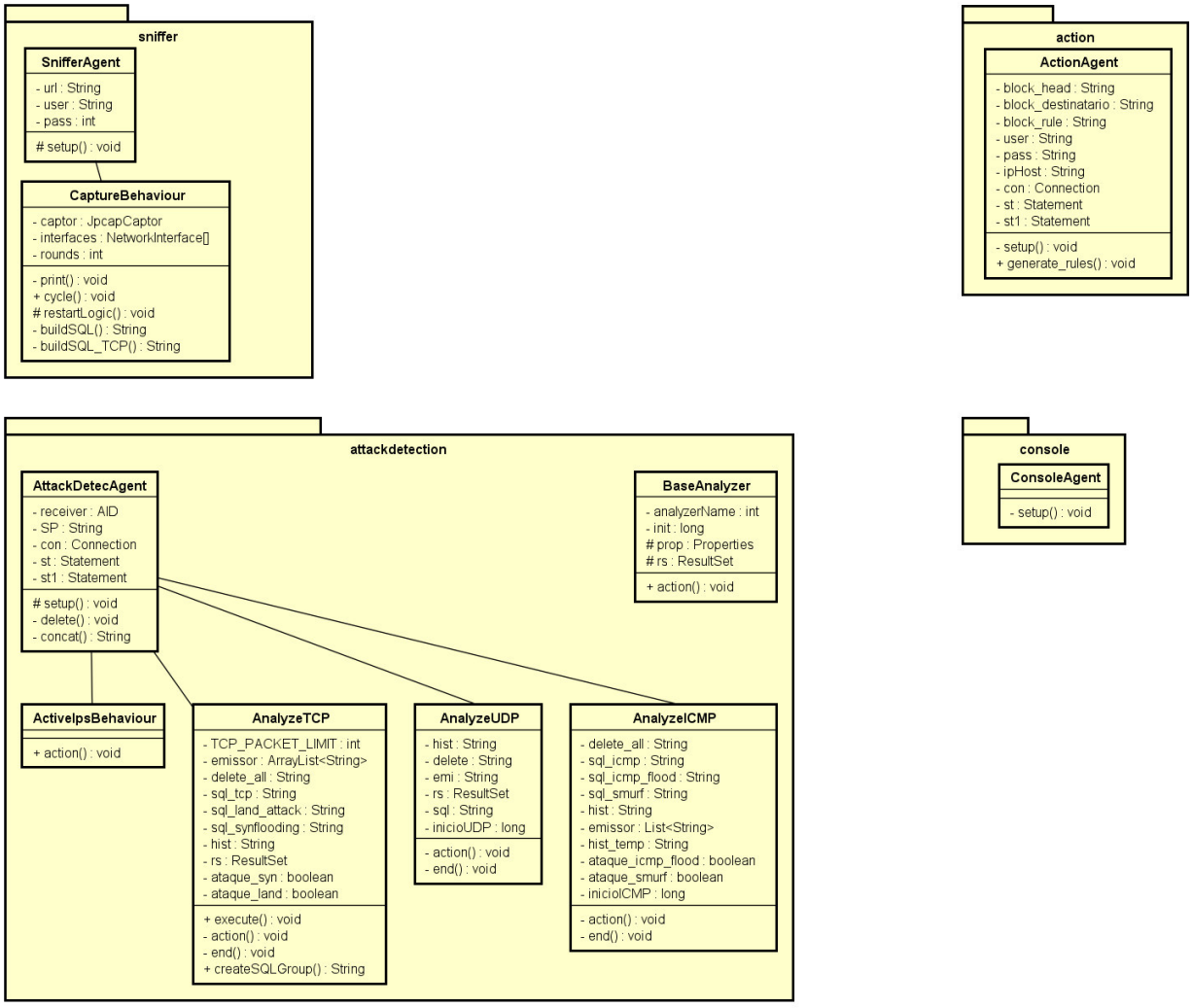
O agente *Action*, por sua vez, ao receber as informações sobre os ataques detectados pelo agente *Attack Detection*, implementa as regras de *firewall* necessárias para a contenção dos ataques e defesa da rede como um todo, além de também enviar relatório para o agente *Console* sobre os eventos por ele gerados.

4.2.2 Diagrama de Classe

O framework ABIDS-WSN é composto por 4 pacotes: *sniffer*, *attackdetection*, *action* e *console*. O pacote *sniffer* detém duas classes: *SnifferAgent* e *CaptureBehaviour*;

o pacote *attackdetection* engloba cinco classes: *AttackDetecAgent*, *ActiveIpsBehaviour*, *AnalyzeTCP*, *AnalyzeUDP*, *AnalyzeICMP* e *BaseAnalyzer*; o pacote *action* contém uma classe, *ActionAgent*; igualmente, o pacote *console* possui apenas uma classe, *ConsoleAgent*. Tal distrubuição dos pacotes e agentes é apresentada no Diagrama de Classe na Figura 17 a seguir:

Figura 17 – Digrama de Classe do Framework ABIDS-WSN



Fonte: O Autor (2017)

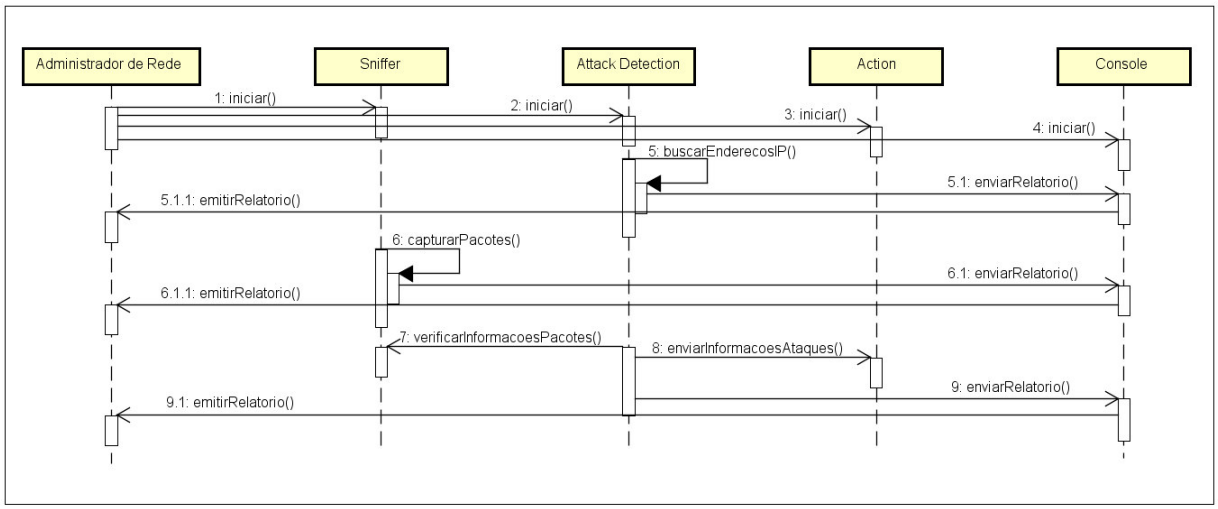
Os 4 agentes estão especificados nas classes *SnifferAgent*, *AttackDetecAgent*, *ActionAgent* e *ConsoleAgent*. Para a implementação de cada um dos agentes foi herdada a superclasse do JADE chamada *Agent*. Cada uma das classes possui um método chamado setup() – um método sobrescrito do JADE –, responsável pelas inicializações necessárias para o funcionamento do agente.

Outra funcionalidade fornecida pelo JADE e aproveitada neste trabalho é o uso de *behaviours* (comportamentos). *Behaviours* são tarefas que são designadas para que um agente as cumpra, podendo ser implementadas em qualquer parte do código da classe do agente. Um dos métodos necessários para ser implementados em uma classe *Behaviour* é o método action() – assim como o setup(), também é um método sobrescrito do JADE –, que define especificamente as ações a serem tomadas por aquele comportamento (CAIRE, 2003). As classes que implementam os comportamentos do ABIDS-WSN são: *CaptureBehaviour*, *ActiveIpsBehaviour*, *AnalyzeTCP*, *AnalyzeUDP* e *AnalyzeICMP*; os comportamentos *CaptureBehaviour* e *ActiveIpsBehaviour* são encarregados da captura dos pacotes e da detecção dos endereços IP ativos na rede, respectivamente, enquanto os comportamentos *AnalyzeTCP*, *AnalyzeUDP* e *AnalyzeICMP* são responsáveis pela análise do tráfego dos protocolos TCP, UDP e ICMP, respectivamente.

4.2.3 Diagrama de Sequência

A sequência de atividades desenvolvidas pelos agentes e as relações e mensagens trocadas em função do tempo entre os 4 agentes do ABIDS-WSN pode ser descrita através da Figura 18:

Figura 18 – Digrama de Sequência do Framework ABIDS-WSN

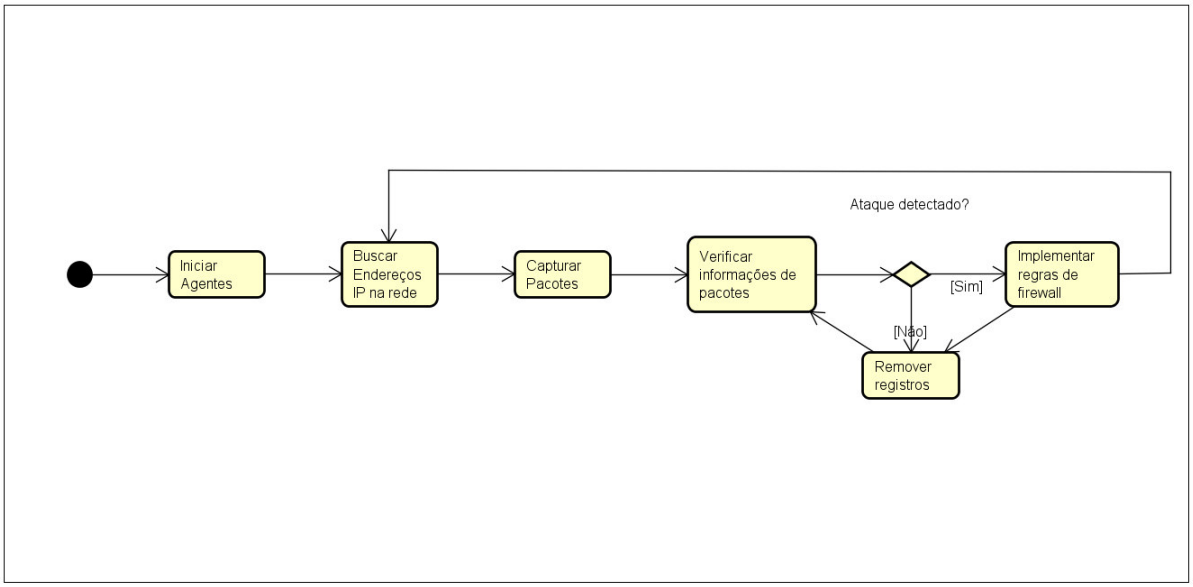


Fonte: O Autor (2017)

4.2.4 Diagramas de Atividades

A sequência de atividades e tomadas de decisão necessárias para a consecução destas é retratada no diagrama de atividades exposto na Figura 19:

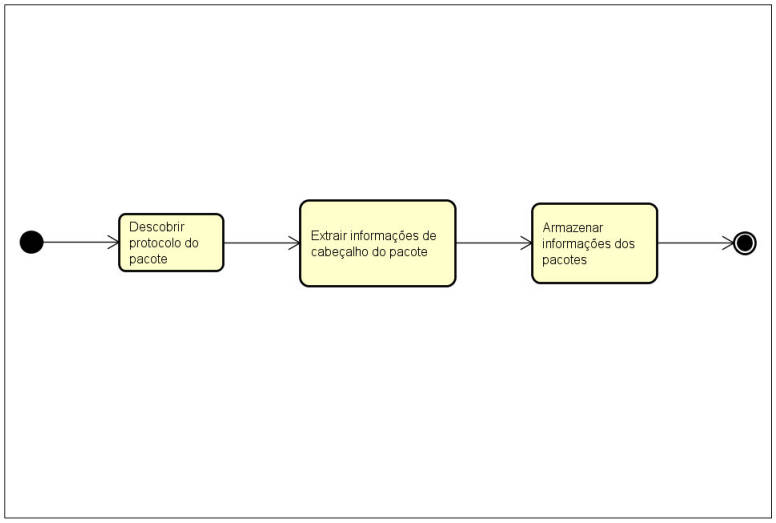
Figura 19 – Digrama de Atividades do Framework ABIDS-WSN



Fonte: O Autor (2017)

Quando à captura dos pacotes, uma das principais funções exercidas pelo *framework*, pode ser descrita através do diagrama de atividades da Figura 20:

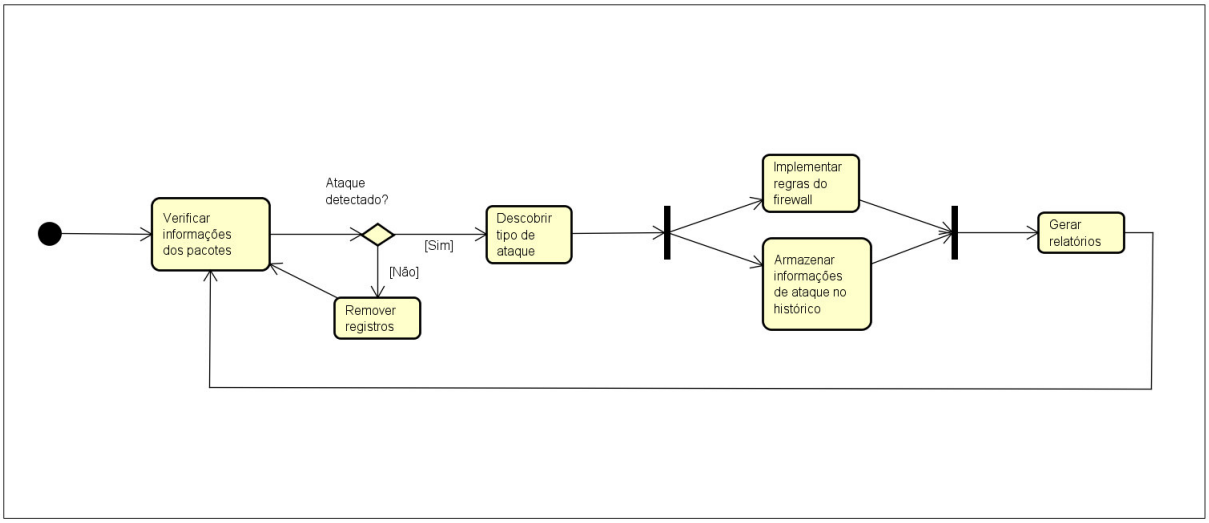
Figura 20 – Digrama de Atividades da captura de pacotes do Framework ABIDS-WSN



Fonte: O Autor (2017)

Quanto ao processo de investigação dos ataques, o diagrama de atividades da Figura 21 apresenta a formalização:

Figura 21 – Digrama de Atividades da análise de ataques do Framework ABIDS-WSN



Fonte: O Autor (2017)

4.3 Prototipagem

Passadas as etapas de modelagem e implementação, tornou-se necessário o desenvolvimento de um protótipo para validação do funcionamento e comportamento do *framework*. O ambiente no qual foi feita a prototipagem possui as seguintes características:

- Plataforma *Raspberry Pi 3 B*;
- Sistema operacional *Raspbian*;
- *JADE* (*Java Agent DEvelopment Framework*), versão 4.4;
- Linguagem *Java*, versão 1.7;
- *Jpcap*, versão 0.7;
- *MySQL*, versão 14.14;
- Ambiente de desenvolvimento IDE *Eclipse*, versão 3.8 (para *Raspbian*).

No Anexo B são apresentadas imagens de etapas de funcionamento da instância do *framework* executada. Nelas são apresentados a inicialização dos quatro agentes que compõem o *framework*, o carregamento dos recursos e bibliotecas do JADE, a primeira atividade *per se* do sistema de detecção de intrusão, que é a detecção dos endereços IP

ativos na rede, executada pelo *Attack Detection*, a primeira rodada de captura de pacotes e a posterior verificação destes, a ação do agente *Attack Detection* e as verificações das informações de cabeçalho separadas para cada tipo de pacote (TCP, UDP e ICMP).

4.4 Síntese

Uma visão geral do *framework* foi apresentada, incluindo sua arquitetura formada por 4 agentes, a modelagem do *framework*, na qual foram expostos alguns diagramas UML para descrição das principais atividades e funcionalidades exercidas. Um protótipo do *framework*, no qual foi apresentada uma instância funcional, também foi descrito.

5 Testes

Neste capítulo, testes sobre o comportamento e o desempenho do *framework* proposto são apresentados. O ambiente, os dados de testes, os resultados obtidos e algumas comparações com os trabalhos relacionados listados no Capítulo 3 serão também apresentados.

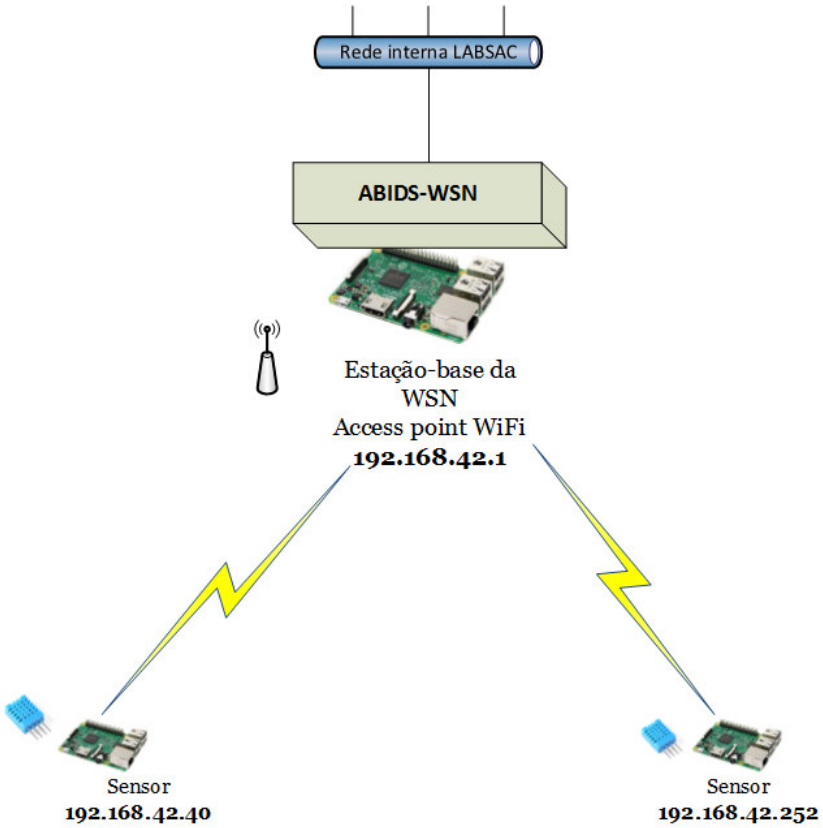
5.1 Ambiente e Dados dos Testes

Para tornar possível os testes do *framework* ABIDS-WSN, foi necessário projetar e implantar um ambiente computacional que os tornasse possíveis. Para o ambiente então, considerou-se necessário a presença mínima de 3 sensores sem fio, o que foi concretizado com 3 *Raspbery Pi 3 B*, com 2 deles equipados com sensores *DHT11* e outro funcionando com uma estação-base. O *Raspberry Pi 3 B* é dotado das seguintes configurações:

- CPU ARMv8, 1.2GHz 64-bit *quad-core*;
- 802.11n *Wireless LAN*;
- 1GB *RAM*;
- 4 portas *USB*;
- Porta *HDMI*;
- Porta *Ethernet*;
- 40 pinos *GPIO*;
- Armazenamento em cartão *Micro SD*.

O sistema embarcado nos aparelhos é o *Raspbian*, que é um sistema operacional baseado na distribuição Linux *Debian* e otimizado para uso no *Raspberry Pi*. A Figura 22 demonstra como se dispõe o ambiente acima descrito:

Figura 22 – Ambiente de testes do Framework ABIDS-WSN



Fonte: O Autor (2017)

Como exposto, a estação-base também funciona como ponto de acesso de uma rede sem fio, através da conectividade fornecida pela rede do LABSAC (Laboratório de Sistemas e Arquiteturas Computacionais), localizado nas dependências da Universidade Federal do Maranhão (UFMA).

Para as simulações dos ataques, o nó sensor de endereço IP 192.168.42.40 foi escolhido para ser o atacante, ao passo que o nó sensor de IP 192.168.40.252 foi selecionado como vítima, em um cenário onde o primeiro é uma máquina usada por um atacante para executar ataques. A ferramenta usada para lançamento dos ataques é o *hping3*[1], um *assembler* de pacotes IP inspirado no comando *ping* que possui suporte a vários protocolos e que possui capacidade de emular situações onde IDS's e firewalls sejam exigidos. O *firewall* a ser usado durante os testes será o *iptables*[2], que é o padrão no Linux e suas distribuições.

Para condução dos testes, os seguintes ataques foram executados contra o ABIDS-WSN:

[1] http://www.hping.org/manpage.html
[2] http://www.netfilter.org/projects/iptables/index.html

- *SYN Flood*;
- *LAND*;
- *ICMP Flood*;
- *Smurf*;
- *UDP Flood*.

Para complemento dos testes e obtenção dos seus resultados, dois *scripts* na linguagem *Python* foram criados e executados de maneira simultânea aos testes com os ataques: um, chamado *DHT11.py*, que coleta dados, a cada três segundos, de temperatura e umidade e envia esses dados a um banco de dados *MySQL* remoto, e outro, chamado *consultaBD.py*, que lê, a cada dez segundos, as coletas armazenadas no banco de dados e as imprime na tela. Estes *scripts* desempenham atividades consideradas normais para a rede de sensores sem fio e sua presença nestes testes tem fundamental importância para a detecção de falsos-positivos (eventualmente disparados por essas operações normais da rede).

Os códigos-fonte dos *scripts* encontram-se no Apêndice A deste trabalho dissertativo, e algumas telas de funcionamento são apresentadas no Anexo C.

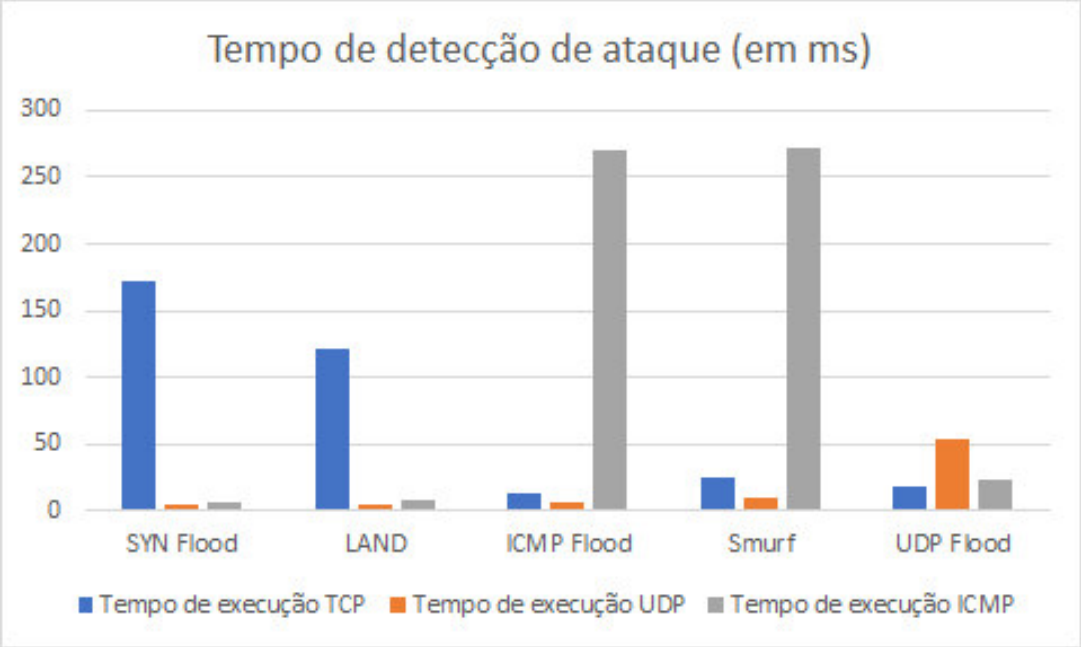
5.2 Resultados dos Testes

Nesta seção, os resultados dos testes executados com o ABIDS-WSN são apresentados. Foram feitas simulações dos ataques citados anteriormente, e para avaliação do desempenho foram escolhidas três métricas: o tempo de detecção dos ataques, calculado pelo próprio *framework* (tempo que o *framework* leva para identificar a presença de um padrão de ataque na leitura das informações dos pacotes), e o volume de entrada e saída de dados (uma vez que o ABIDS-WSN baseia suas operações em I/O), feito através do uso da ferramenta *iotop*[3], um programa em *Python*, inspirado no comando *top* do Linux, que visa apresentar, além dos processos executados em tempo real, o volume de entrada e saída gerado por cada um deles e o volume de uso da CPU, aferido pelo uso do comando *top*, disponível nas distribuições *Linux*.

[3] http://guichaz.free.fr/iotop/

Quanto ao primeiro teste, quanto ao tempo de detecção dos ataques, foram executados 10 ataques de cada um dos ataques citados, e foram extraídas as médias aritméticas dos tempos de detecção. As médias estão apresentadas conforme a Figura 23 e a Tabela 4:

Figura 23 – Gráfico de barras com valores médios dos tempos de detecção dos ataques contra a WSN



Fonte: O Autor (2017)

Tabela 4 – Tabela com valores médios dos tempos de detecção dos ataques contra a WSN (em ms)

	SYN Flood	***LAND***	***ICMP Flood***	***Smurf***	***UDP Flood***
Tempo de execução *TCP*	172	121,7	13	25,5	17,4
Tempo de execução *UDP*	4,1	3,8	5,4	9,2	53,2
Tempo de execução *ICMP*	6,4	7	270,2	271,3	23,5

Quanto à efetividade da detecção de cada um dos ataques, o ABIDS-WSN foi bem sucedido em cada um dos ataques testados; a matriz de confusão da Tabela 5 apresenta os resultados:

O valor do número de pacotes representa a assinatura do ataque e diz respeito ao número de pacotes correspondentes a um padrão de ataque que o ABIDS-WSN considerará seguro, antes de emitir um alerta de ataque; qualquer tráfego correspondente a um padrão de uma assinatura conhecida pelo ABIDS-WSN cujo número de pacotes ultrapasse esse número, o framework irá considerar como um ataque, emitirá um alerta e tomará as devidas contramedidas. A extração desses valores foi aplicada de maneira empírica, através

Tabela 5 – Matriz de confusão dos ataques testados e número limite de pacotes passantes considerados inseguros

	Verdadeiros-positivos	**Falsos-positivos**	**Número máximo de pacotes**
SYN Flood	10	0	180
LAND	10	0	180
ICMP Flood	10	0	60
Smurf	10	0	60
UDP Flood	10	0	300

de casos de testes e cenários de uso do *framework*, nos quais eram obtidos números de pacotes que poderia se dizer, com segurança, que correspondiam a um ataque. A partir desses valores, tornou-se possível determinar uma margem mínima na determinação do que seja um ataque ou não.

Para a simulação dos ataques, foi usado o programa *hping3*. O programa funciona no Linux através de comandos do Terminal, e cada um dos ataques foi lançado a partir do sensor de endereço IP 192.168.42.40, cada um com duração de 10 segundos (tornado possível através do comando timeout[4]). Os comandos usados para os ataques foram os seguintes:

- *SYN Flood*: timeout -s9 10s hping -V -c 1000000 -S -w 64 -p 135 -s 135 –flood -a [IP_DO_ATACANTE] 192.168.42.252;
- *LAND*: timeout -s9 10s hping -V -c 1000000 -S -w 64 -p 135 -s 135 –flood -a 192.168.42.252 192.168.42.252;
- *ICMP Flood*: timeout -s9 10s hping -a [IP_DO_ATACANTE] -1 192.168.42.252 –flood;
- *Smurf*: timeout -s9 10s hping -a 192.168.42.40 -1 192.168.42.255 –flood;
- *UDP Flood*: timeout -s9 10s hping3 -2 -a [IP_DO_ATACANTE] 192.168.42.252 –flood;

O *-s9 10s* significa que ao fim do tempo de 10 segundos será dado o comando *kill(9)* do Linux, terminando abruptamente o comando; *-V* se refere à emissão de uma saída verbosa; *-c [NÚMERO]* se refere à contagem de pacotes que serão enviados ao destino; *-d [NÚMERO]* se refere ao tamanho de cada um dos pacotes a serem enviados; *-s* e *-p* se referem às portas de origem e destino, respectivamente; *–flood* é uma opção que faz com que os pacotes sejam enviados à vítima na maior velocidade possível, sem esperar por respostas; *-a [IP_DO_ATACANTE]* é uma opção que permite que o pacote IP tenha como origem um endereço IP falsificado. Além disso, faz com que a vítima tente responder, sem

[4] http://man7.org/linux/man-pages/man1/timeout.1.html

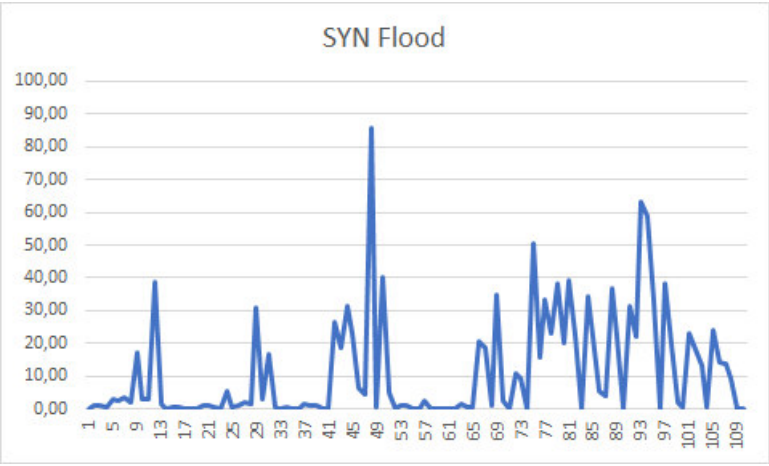
sucesso, ao IP falsificado; *-1* faz com que o *hping3* se comporte no modo *ICMP* e *-2* faz com que o *hping3* se comporte no modo *UDP*.

Para defesa da WSN contra cada um dos ataques simulados acima, foras implementadas regras de *firewall* para bloquear os ataques. O *firewall* usado nos testes foi o nativo no *Raspbian* e nas demais distribuições *Linux*, o *iptables*. Contra cada uma dos ataques, o ABIDS-WSN gerou uma regra do iptables. Tal regra é apresentada da seguinte maneira: iptables -A FORWARD -s [IP_DO_ATACANTE] -j DROP.

O *-A* significa que tal regra deve ser adicionada ao conjunto de regras de filtragem de pacotes gerida pelo *iptables*; a *chain FORWARD* diz respeito a o que fazer com os pacotes que estiverem de passagem pelo *host* onde está o ABIDS-WSN (tendo ele ou outra máquina como destinatário); *-s* se refere à fonte dos pacotes sobre os quais agirá a regra; *-j* se refere ao *host* destino dos pacotes sobre os quais agirá a regra: no ABIDS-WSN, para uma maior segurança, tal campo é deixando em branco, resultando que não apenas o *host* vítima seja protegido, mas todos os demais *hosts* da rede serão protegidos de um nó que se sabe, dali em diante, que é um atacante; *DROP* é a ação a ser tomada em relação a todos os pacotes que atendem à descrição feita pelos outros argumentos; no caso em tela, *DROP* faz com que os pacotes sejam perdidos.

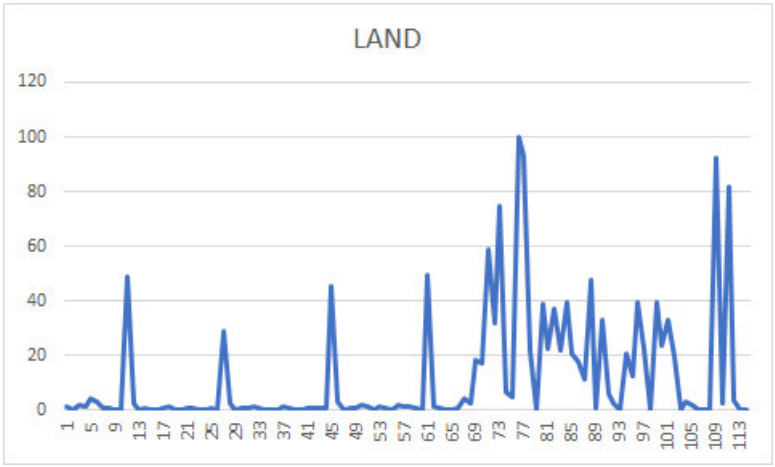
Quanto ao consumo dos recursos de entrada e saída (em %) registrados pelo *iotop* registrados durante os ataques, a coleta dos dados revelou os seguintes gráficos:

Figura 24 – Consumo de recursos de I/O - SYN Flood (em %)



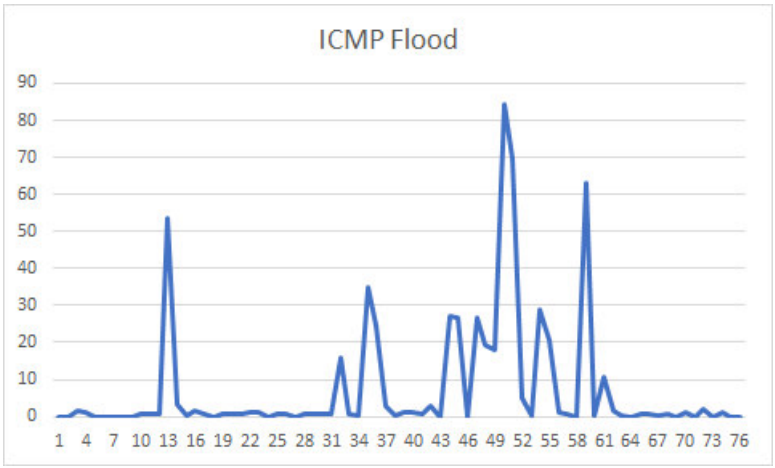
Fonte: O Autor (2017)

Figura 25 – Consumo de recursos de I/O - LAND (em %)



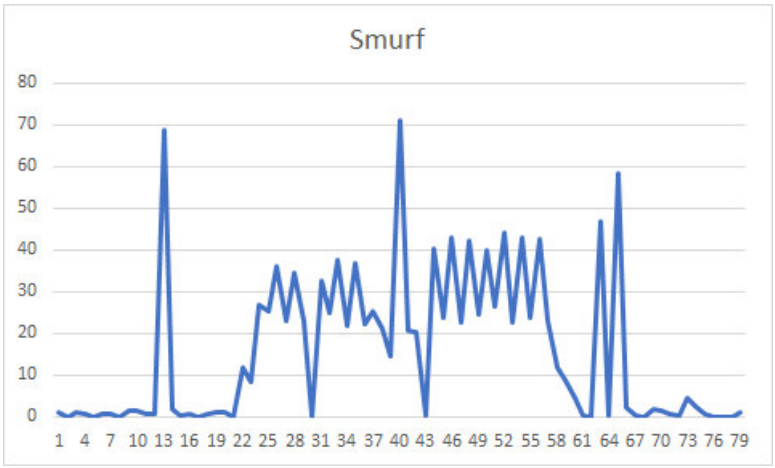
Fonte: O Autor (2017)

Figura 26 – Consumo de recursos de I/O - ICMP Flood (em %)



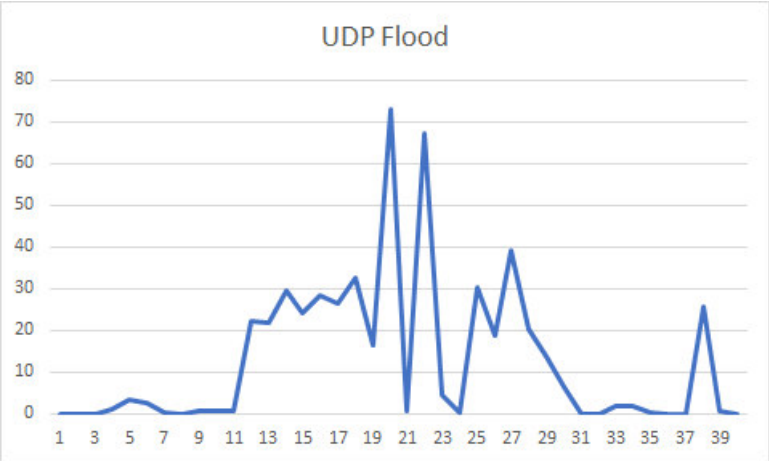
Fonte: O Autor (2017)

Figura 27 – Consumo de recursos de I/O - Smurf (em %)



Fonte: O Autor (2017)

Figura 28 – Consumo de recursos de I/O - UDP Flood (em %)

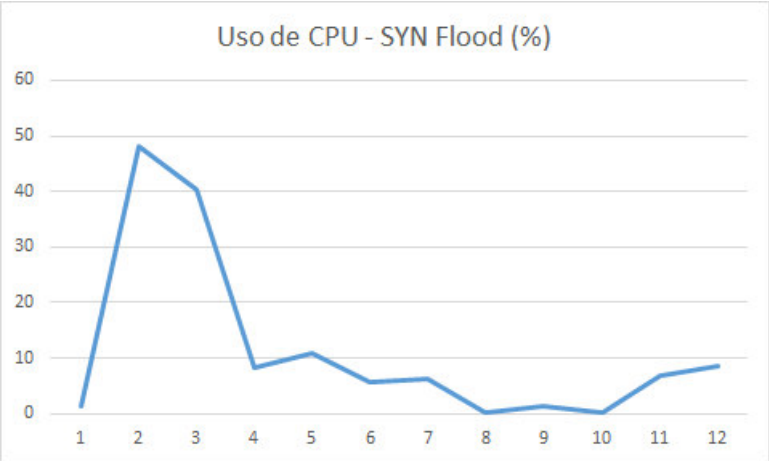


Fonte: O Autor (2017)

Os picos de consumo, geralmente registrados do início da segunda metade da execução do *framework* até perto do final, demonstram a atividade mais intensa do ABIDS-WSN, que é o uso intenso de bancos de dados (leitura e escrita), ocorridos principalmente após a fase de captura de pacotes.

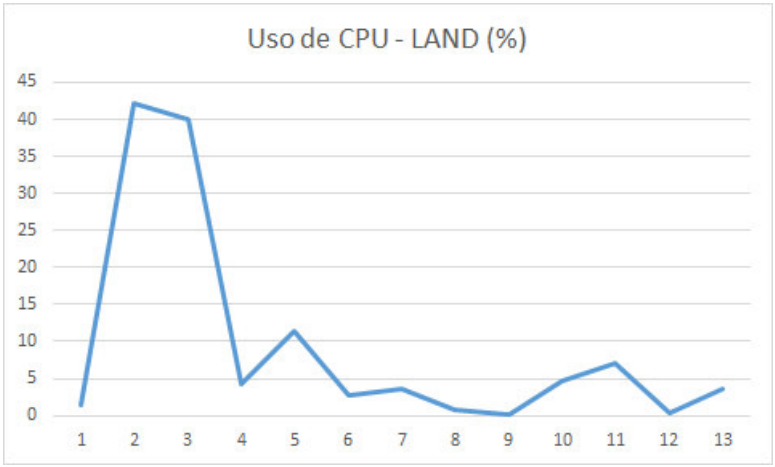
Quanto ao consumo dos recursos de CPU (em %) registrados pelo comando *top* durante os ataques testados, foram extraídos valores que são representados conforme os gráficos abaixo:

Figura 29 – Consumo de recursos de CPU - SYN Flood (em %)



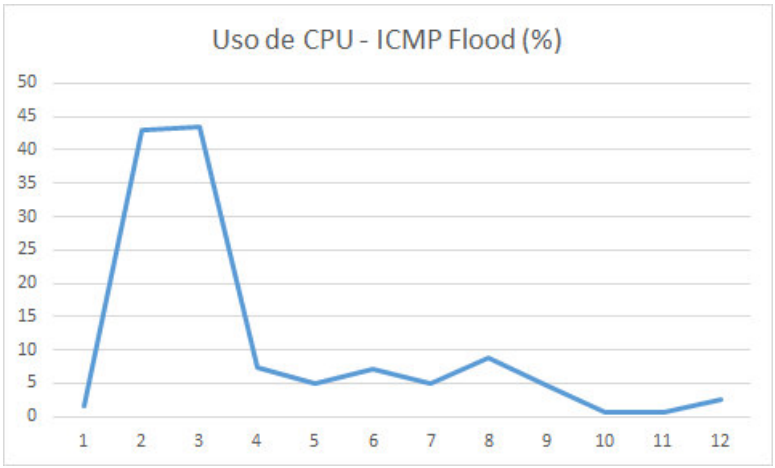
Fonte: O Autor (2017)

Figura 30 – Consumo de recursos de CPU - LAND (em %)



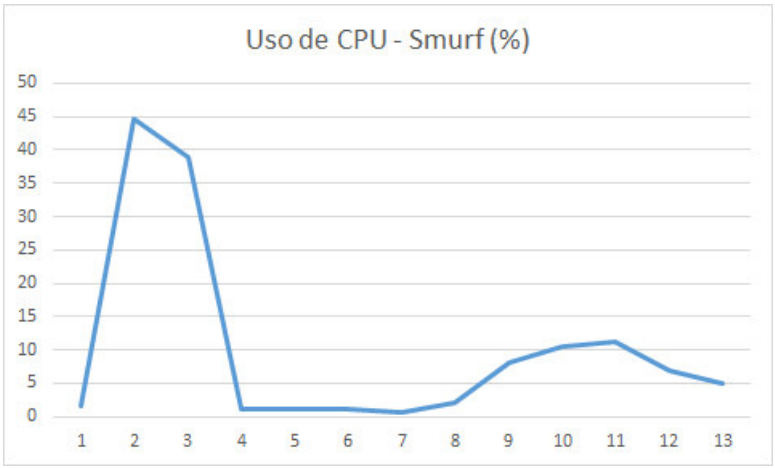
Fonte: O Autor (2017)

Figura 31 – Consumo de recursos de CPU - ICMP Flood (em %)



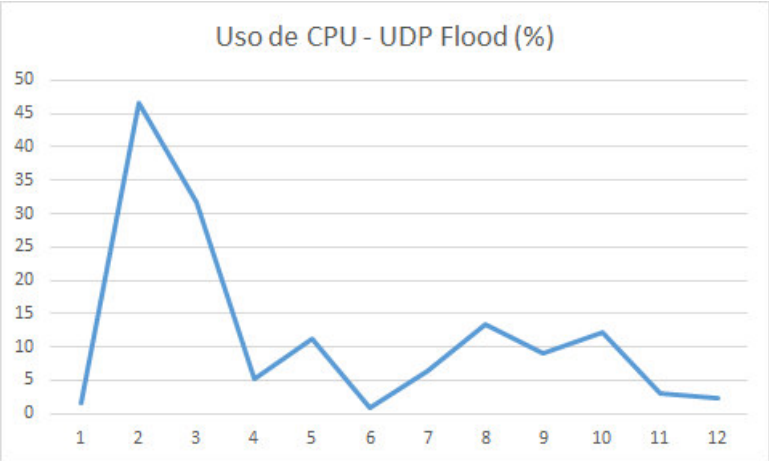
Fonte: O Autor (2017)

Figura 32 – Consumo de recursos de CPU - Smurf (em %)



Fonte: O Autor (2017)

Figura 33 – Consumo de recursos de CPU - UDP Flood (em %)



Fonte: O Autor (2017)

Foi possível perceber, através dos valores extraídos nos testes de medição de uso de CPU, que o ABIDS-WSN mantém baixo uso de CPU, de maneira praticamente constante ao longo do seu funcionamento, havendo apenas um pico na sua inicialização.

5.3 Comparativo com Trabalhos Relacionados

Os trabalhos relacionados, além de fornecerem recursos teóricos e sugestões para este trabalho, podem fornecer parâmetros para o trabalho proposto. Portanto, nessa seção algumas comparações entre os trabalhos relacionados e o trabalho proposto serão apresentados.

No trabalho apresentado por Khanum et al. (2010), os principais resultados obtidos foram: diminuição de requisições ao nó estação-base, minimizações de mensagens de erro, redução da carga de rede e diminuição dos recursos usados pelos nós. O *framework* proposto também logrou um menor consumo dos recursos dos nós, conforme apresentado nas Figuras 23-23, as quais retratam um consumo mais intenso dos recursos mais intensos apenas na fase de tomada de contramedidas.

No trabalho proposto por Wang, Yuan e Wang (2009), o uso de dois algoritmos em um IDS orientado por anomalias é apresentado: o algoritmo SOM e o S-K. Durante os experimentos, concluiu-se que o algoritmo S-K possui uma melhor taxa de detecção de intrusões, chegando muito perto de uma taxa de 100%. No *framework* proposto neste trabalho, foi alcançada uma taxa de detecção de 100%, através das assinaturas determinadas para o *framework*. Além disso, nos testes do trabalho proposto por Wang, Yuan e Wang

(2009) há um maior número de nós (doze, mais precisamente) do que os usados no presente trabalho dissertativo; tal discrepância acaba por não comprometer a comparação, uma vez que os agentes, ainda que sendo executados de maneira distribuída, rodam em uma plataforma que permite que o desempenho dos agentes seja semelhante caso estivessem sendo executados localmente. Também foi notada outra semelhança entre os dois trabalhos, que foi o teste de um tipo de ataque cada vez, sem testes com ataques simultâneos.

No trabalho proposto por Yu e Tsai (2008), o IDS é composto por apenas um agente em cada *host* da rede, o que pode ser uma desvantagem, uma vez que a capacidade de mobilidade, a qual permite um melhor balanceamento de carga. O *framework* ABIDS-WSN, através da sua capacidade de mobilidade e uso das mensagens *ACL*, pode trabalhar com mobilidade, e esta pode ser aprimorada através das comunicações inter-agentes permitidas pela linguagem *ACL*.

No trabalho proposto por Hai, Huh e Jo (2010), foi observado que quanto maior o número de nós na rede, maior será o número de falsos positivos; pelo observado nos testes do ABIDS-WSN, através de implantação de casos de teste, nos quais mais nós foram implantados na rede, não há indícios de que isto ocorra; percebeu-se que a única grandeza que pode ser influenciada pelo número de nós da rede é o tempo levado para aferição dos endereços IP ativos na rede.

No trabalho proposto por Haddadi e Sarram (2010) percebeu-se que, dentre outros tipos de ataque, houve uma satisfatória resposta contra ataques de negação de serviço. O ABIDS-WSN, testado em um ambiente análogo ao deste trabalho (ambos com máquinas gerando tráfego de rede para os demais sensores; além disso, nos dois casos, nestes elementos de rede geradores de tráfego há a passagem dos pacotes nocivos), também mostrou desempenho satisfatório em desenvolver contramedidas bem sucedidas contra alguns destes tipos de ataques.

No trabalho proposto por Sun e Liu (2013) ocorre um fenômeno semelhante: o tempo de processamento dos ataques aumenta, à medida que o número de nós afetados por ataques também aumenta, fato este que também não mostrou evidências de ocorrer no funcionamento do *framework* ABIDS-WSN.

Por fim, apresenta-se a Tabela 6 na qual as propostas são comparadas em função dos ataques contra os quais cada uma demonstrou sucesso:

Abaixo, uma legenda sobre os termos abreviados usados na Tabela 6:

Tabela 6 – Tabela comparativa dos resultados do *framework* proposto e dos trabalhos relacionados

Ataque	1	2	3	4	5	6	7	**ABIDS-WSN**
SYN Flood	N/I		*					✓
LAND	N/I		*					✓
ICMP Flood	N/I		*					✓
Smurf	N/I	✓	*					✓
UDP Flood	N/I		*					✓
Portsweep	N/I	✓	*					
Nmap	N/I	✓	*					
Wormhole	N/I		*	✓				
Sinkhole	N/I		*	✓				
Selective Forwading	N/I		*	✓				
HELLO Flood	N/I		*	✓				
Impersonação	N/I		*		✓			
Network discovery	N/I		*		✓			
Negação de serviço	N/I		*		✓		✓	
Flooding	N/I		*			✓		
Blackhole	N/I		*			✓		

- **1**: *Energy-Efficient Intrusion Detection System for Wireless Sensor Network Based on MUSK Architecture*;
- **2**: *Intrusion Detection for Wireless Sensor Networks Based on Multi-agent and Refined Clustering*;
- **3**: *A Framework of Machine Learning Based Intrusion Detection for Wireless Sensor Networks*;
- **4**: *A lightweight intrusion detection framework for wireless sensor networks*;
- **5**: *Wireless Intrusion Detection System Using a Lightweight Agent*;
- **6**: *Lightweight energy consumption-based intrusion detection system for wireless sensor networks*;
- **7**: *Agent-based intrusion detection and self-recovery system for wireless sensor networks*.

Embora tenha sido um trabalho que expôs resultados, o trabalho de Khanum et al. (2010) não apresentou maiores detalhes sobre quais ataques ele tenha tomado contramedidas. Além disso, o trabalho proposto por Yu e Tsai (2008) não citou ataque nenhum em específico, pelo fato de o framework proposto neste trabalho ser voltado para identificar e detectar ataques através de aprendizado de máquina. Percebe-se também que as soluções relacionadas não são testadas com uma grande gama de ataques, sendo apenas

testadas com um pequeno conjunto de ataques, ao passo que o ABIDS-WSN foi testado com cinco ataques, havendo a possibilidade de extensão desse rol.

5.4 Síntese

Neste capítulo, os testes com o *framework* proposto foram apresentados. Neste ambiente, dispositivos *Raspberry Pi 3 B* foram usados, atuando como sensores de umidade e temperatura, através de circuitos sensores *DHT11*. Como teste desses sensores e estabelecimento de um cenário de normal funcionamento da rede, foram desenvolvidos dois *scripts* na linguagem *Python* que capturassem dados de umidade e temperatura e executassem operações com esses dados em um banco de dados *MySQL*.

Os testes em um cenário onde foram usados três dispositivos *Raspberry Pi 3 B* foram apresentados, no qual dois atuavam como nós sensores e outro atuava como estação-base e ponto de acesso sem fio. Primeiramente, os tempos de detecção para cada tipo de ataque foram medidos, seguido de uma matriz de confusão de cada ataque efetivado. Logo após foi apresentada uma descrição de como foram simulados os ataques, bem como a construção das defesas através da geração de regras de *firewall*. Em seguida uma matriz de confusão das detecções do ABIDS-WSN, enquanto eram executados os *scripts Python* foi apresentada. Finalmente, os dados de consumo dos recursos de CPU e entrada e saída dos dispositivos foram apresentados.

6 Conclusão

O atual capítulo, enquanto conclusão deste trabalho dissertativo, apresentará uma discussão sobre os objetivos alcançados, as limitações apresentadas pelo sistema ABIDS-WSN, publicações acadêmicas relacionadas ao trabalho e discussões acerca de prováveis trabalhos futuros.

6.1 Objetivos alcançados

Tendo por base um referencial bibliográfico sobre as mais diversas abordagens a respeito de sistemas de detecção de intrusão em redes de sensores sem fio, foi proposto neste trabalho dissertativo um *framework*, chamado ABIDS-WSN (*Agent-based Intrusion Detection System for Wireless Sensor Networks*). Foi adotada a abordagem dos agentes inteligentes, devido a facilidades providas como: execução autônoma, possibilidade de desenvolvimento em ambientes heterogêneos e distribuição de carga, um fator essencial em sistemas de recursos computacionais escassos. Para medição dessas vantagens foi modelado e implantado um cenário de testes, no qual o ABIDS-WSN viria a ser testado nos seguintes aspectos:

- Precisão das detecções de ataques;
  - Presença de verdadeiros-positivos e falsos-positivos;
- Tempo de detecção dos ataques, em milissegundos;
- Consumo de recursos de entrada e saída;
- Consumo de recursos de CPU;

Após essa etapa, foi trazido a lume um protótipo de uso do ABIDS-WSN, contra o qual foram lançados alguns tipos de ataques, cada um deles gerando um efeito de contra-ataque e dados numéricos quanto ao tempo de detecção dos ataques. Além disso, dois *scripts* foram usados como testes de comportamento do *framework* ABIDS-WSN em uma situação de uso normal.

Os resultados dos testes, sobretudo as medições de consumo de recursos de entrada e saída e uso de CPU, demonstraram que o ABIDS-WSN lidou de maneira satisfatória com restrições oferecidas, mantendo um baixo consumo de recursos computacionais (como I/O e

CPU) e não exaurindo os poucos recursos computacionais ofertados pelos dispositivos. Uma vez que houve esse comportamento considerado satisfatório, o uso de agentes inteligentes no desenvolvimento de uma solução de detecção de intrusão teve a sua adequação comprovada.

A principal contribuição deste trabalho foi a proposta de um *framework* de desempenho computacional condizente com as redes de sensores sem fio, adaptável e escalável, com baixo tempo de latência e capaz de balanceamento de carga. Os resultados obtidos no Capítulo anterior são considerados satisfatórios, tendo em conta o hardware fornecido pelos dispositivos.

6.2 Limitações

Algumas limitações foram constatadas na proposição deste trabalho, a saber:

- O número de pacotes suspeitos a serem tolerados pelo agente *Sniffer* é pré-fixado, não permitindo uma atualização em tempo de execução e, portanto, uma adaptação às diversas circunstâncias e critérios de uma rede de sensores sem fio;
- Com o número de pacotes a serem tolerados sendo pré-fixado, é possível que em outros cenários diferentes do apresentado neste trabalho, haja uma menor taxa de sucesso do *framework*;
- O *framework* usa basicamente recursos de entrada e saída (operações em um banco de dados *MySQL*), o que acaba por ser um impeditivo para um melhor desempenho computacional, uma vez que as unidades de armazenamento secundário usadas pelos sensores possui um *overhead* baixo;
- O cenário de ataques foi reduzido, com apenas um *host* atacante e um *host* sendo atacado.

6.3 Trabalhos Futuros

Quando a trabalhos futuros relacionados ao apresentado através deste trabalho, há alguns pontos que podem ser explorados:

- Uso de técnicas que permitam adaptações no número de pacotes a serem tolerados pelo agente *Sniffer*, para cálculo do número de pacotes em tempo de execução; dentre dessas técnicas, é possível citar: redes neurais, redes bayesianas, aprendizado baseado

em árvores de decisão, *SVM* (*Support Vector Machine*), lógica *fuzzy* e ciência de contexto.

- Desenvolvimento de cenários de ataques com mais nós sensores na rede;
- Criação de protótipos que trabalhem com ataques não abordados neste trabalho, como, por exemplo, ataques contra protocolos de roteamento;
- Tornar possível que o *framework* trabalhe com o protocolo 6LoWPAN, protocolo definido pela IEEE 802.15.4 e voltado para dispositivos de baixa energia;
- Tornar possível que o *framework* trabalhe com ataques baseados além dos três abordados no presente trabalho;
- Trocar os recorrentes acessos a I/O por acessos à memória, através de, por exemplo, algoritmos como o produtor-consumidor.

6.4 Publicações

Este trabalho dissertativo ensejou a produção de um artigo aprovado na *Second International Conference on Internet of Things, Data and Cloud Computing* (ICC 2017), a ser realizada no Churchill College, da Universidade de Cambridge, localizada na cidade de mesmo nome no Reino Unido, nos próximos dias 22 e 23 de março de 2017. O título do artigo aceito é *A Framework for Agent-based Intrusion Detection in Wireless Sensor Networks*.

6.5 Considerações Finais

Ao fim deste trabalho, pode-se concluir, através dos resultados obtidos, que a pesquisa ratificou a hipótese e fez cumprir os objetivos gerais e específicos levantados inicialmente; o uso de agentes inteligentes se demonstrou propício para a proposta e posterior desenvolvimento de um *framework* para detecção de intrusão em redes de sensores sem fio. Além disso, a metodologia proposta também foi considerada satisfatória para o cumprimento dos objetivos e procedimentos necessários para a confirmação da hipótese. Além disso, a bibliografia levantada tornou possível vários levantamentos que deram origem à arquitetura final do *framework*.

Foi considerado também que, embora possam colaborar de maneira bastante relevante no desenvolvimento de soluções de segurança em redes de sensores sem fio, não

apenas os agentes podem demonstrar resultados importantes, mas também outros tipos de abordagem, como Teoria dos Jogos ou algoritmos de aprendizado. Considerou-se também que a combinação destas técnicas entre si, e com outras técnicas, também tem potencial para gerar soluções igualmente importantes.

Com a condução dos testes, foi possível perceber com maior detalhe a importância de manter o uso dos recursos computacionais em redes de sensores sem fio em níveis satisfatórios, o que reforçou, também, a importância do desenvolvimento de ferramentas leves no que se refere a consumo de *hardware*, previamente estudada no referencial bibliográfico.

Referências

BAIG, Z. A. Pattern recognition for detecting distributed node exhaustion attacks in wireless sensor networks. *Computer Communications*, Elsevier, v. 34, n. 3, p. 468–484, 2011. Citado 2 vezes nas páginas 10 e 36.

BUTUN, I.; MORGERA, S. D.; SANKAR, R. A survey of intrusion detection systems in wireless sensor networks. *IEEE Communications Surveys & Tutorials*, IEEE, v. 16, n. 1, p. 266–282, 2014. Citado 2 vezes nas páginas 17 e 18.

BYSANI, L. K.; TURUK, A. K. A survey on selective forwarding attack in wireless sensor networks. In: IEEE. *Devices and Communications (ICDeCom), 2011 International Conference on*. [S.l.], 2011. p. 1–5. Citado na página 27.

CAIRE, G. Jade tutorial: Jade programming for beginners. *Documentación de JADE*, v. 3, 2003. Citado na página 53.

CAN, O.; SAHINGOZ, O. K. A survey of intrusion detection systems in wireless sensor networks. In: IEEE. *Modeling, Simulation, and Applied Optimization (ICMSAO), 2015 6th International Conference on*. [S.l.], 2015. p. 1–6. Citado 4 vezes nas páginas 24, 26, 27 e 29.

DALLAS, D.; LECKIE, C.; RAMAMOHANARAO, K. Hop-count monitoring: Detecting sinkhole attacks in wireless sensor networks. In: IEEE. *2007 15th IEEE International Conference on Networks*. [S.l.], 2007. p. 176–181. Citado na página 27.

DEGIRMENCIOGLU, A. et al. A classification approach for adaptive mitigation of syn flood attacks: Preventing performance loss due to syn flood attacks. In: IEEE. *Network Operations and Management Symposium (NOMS), 2016 IEEE/IFIP*. [S.l.], 2016. p. 1109–1112. Citado na página 27.

GANDHIMATHI, L.; MURUGABOOPATHI, G. Cross layer intrusion detection and prevention of multiple attacks in wireless sensor network using mobile agent. In: IEEE. *Information Communication and Embedded Systems (ICICES), 2016 International Conference on*. [S.l.], 2016. p. 1–5. Citado na página 22.

HADDADI, F.; SARRAM, M. A. Wireless intrusion detection system using a lightweight agent. In: IEEE. *Computer and Network Technology (ICCNT), 2010 Second International Conference on*. [S.l.], 2010. p. 84–87. Citado 4 vezes nas páginas 10, 43, 44 e 67.

HAI, T. H.; HUH, E.-N.; JO, M. A lightweight intrusion detection framework for wireless sensor networks. *Wireless Communications and mobile computing*, Wiley Online Library, v. 10, n. 4, p. 559–572, 2010. Citado 4 vezes nas páginas 10, 42, 43 e 67.

HAN, S. et al. Taxonomy of attacks on wireless sensor networks. In: *EC2ND 2005*. [S.l.]: Springer, 2006. p. 97–105. Citado 2 vezes nas páginas 30 e 34.

HU, Y.-C.; PERRIG, A.; JOHNSON, D. B. Packet leashes: a defense against wormhole attacks in wireless networks. In: IEEE. *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*. [S.l.], 2003. v. 3, p. 1976–1986. Citado na página 27.

JAN, M. A.; KHAN, M. Denial of service attacks and their countermeasures in wsn. *IRACST–International Journal of Computer Networks and Wireless Communications (IJCNWC)*, v. 3, 2013. Citado 3 vezes nas páginas 25, 26 e 35.

KARLOF, C.; WAGNER, D. Secure routing in wireless sensor networks: Attacks and countermeasures. *Ad hoc networks*, Elsevier, v. 1, n. 2, p. 293–315, 2003. Citado na página 26.

KHANUM, S.; USMAN, M.; ALWABEL, A. Mobile agent based hierarchical intrusion detection system in wireless sensor networks. *International Journal of Computer Science Issues (IJCSI)*, Citeseer, v. 9, n. 1, p. 101–108, 2012. Citado 2 vezes nas páginas 18 e 25.

KHANUM, S. et al. Energy-efficient intrusion detection system for wireless sensor network based on musk architecture. In: *High Performance Computing and Applications*. [S.l.]: Springer, 2010. p. 212–217. Citado 3 vezes nas páginas 40, 66 e 68.

LI, G.; HE, J.; FU, Y. Group-based intrusion detection system in wireless sensor networks. *Computer Communications*, Elsevier, v. 31, n. 18, p. 4324–4332, 2008. Citado na página 36.

LI, W. et al. A new intrusion detection system based on knn classification algorithm in wireless sensor network. *Journal of Electrical and Computer Engineering*, Hindawi Publishing Corporation, v. 2014, 2014. Citado 2 vezes nas páginas 22 e 25.

MALIK, M. Y. An outline of security in wireless sensor networks: threats, countermeasures and implementations. *arXiv preprint arXiv:1301.3022*, 2013. Citado 3 vezes nas páginas 12, 34 e 35.

MESSAI, M.-L. Classification of attacks in wireless sensor networks. In: *International Congress on Telecommunication and Application14*. [S.l.: s.n.], 2014. Citado 2 vezes nas páginas 25 e 26.

MOON, S. Y.; KIM, J. W.; CHO, T. H. An energy-efficient routing method with intrusion detection and prevention for wireless sensor networks. In: IEEE. *16th International Conference on Advanced Communication Technology*. [S.l.], 2014. p. 467–470. Citado na página 37.

MOURABIT, Y. E. et al. Intrusion detection system in wireless sensor network based on mobile agent. In: IEEE. *Complex Systems (WCCS), 2014 Second World Conference on*. [S.l.], 2014. p. 248–251. Citado 3 vezes nas páginas 18, 19 e 32.

NEWSOME, J. et al. The sybil attack in sensor networks: analysis & defenses. In: ACM. *Proceedings of the 3rd international symposium on Information processing in sensor networks*. [S.l.], 2004. p. 259–268. Citado na página 28.

NGAI, E. C.; LIU, J.; LYU, M. R. An efficient intruder detection algorithm against sinkhole attacks in wireless sensor networks. *Computer Communications*, Elsevier, v. 30, n. 11, p. 2353–2364, 2007. Citado na página 39.

NORVIG, P.; RUSSELL, S. *Inteligência Artificial, 3a Edição*. [S.l.]: Elsevier Editora Ltda, 2014. Citado 3 vezes nas páginas 19, 30 e 31.

PANAIT, L.; LUKE, S. Cooperative multi-agent learning: The state of the art. *Autonomous agents and multi-agent systems*, Kluwer Academic Publishers, v. 11, n. 3, p. 387–434, 2005. Citado na página 32.

PANIGRAHI, R.; SHARMA, K.; GHOSE, M. Wireless sensor networks–architecture, security requirements, security threats and its countermeasures. In: *The Third International Conference on Computer Science & Information Technology (CCSIT), Jan Zizka (Eds): CCSIT, SIPP, AISC, PDCTA*. [S.l.: s.n.], 2013. p. 107–115. Citado na página 34.

PATEL, M. M.; AGGARWAL, A. Security attacks in wireless sensor networks: A survey. In: IEEE. *Intelligent Systems and Signal Processing (ISSP), 2013 International Conference on*. [S.l.], 2013. p. 329–333. Citado na página 17.

PATIL, N. et al. Analysis of distributed intrusion detection systems using mobile agents. In: IEEE. *2008 First International Conference on Emerging Trends in Engineering and Technology*. [S.l.], 2008. p. 1255–1260. Citado na página 32.

RAO, G. S. et al. Security assessment of computer networks-an ethical hacker's perspective. In: IEEE. *Computer and Communications Technologies (ICCCT), 2014 International Conference on*. [S.l.], 2014. p. 1–5. Citado na página 27.

RASSAM, M. A.; MAAROF, M.; ZAINAL, A. A survey of intrusion detection schemes in wireless sensor networks. *American Journal of Applied Sciences*, Science Publications, v. 9, n. 10, p. 1636, 2012. Citado na página 29.

RIECKER, M. et al. Lightweight energy consumption-based intrusion detection system for wireless sensor networks. *International Journal of Information Security*, Springer, v. 14, n. 2, p. 155–167, 2015. Citado 2 vezes nas páginas 44 e 45.

SCARFONE, K.; MELL, P. Guide to intrusion detection and prevention systems (idps). *NIST special publication*, v. 800, n. 2007, p. 94, 2007. Citado 2 vezes nas páginas 28 e 29.

SHAMSHIRBAND, S. et al. D-ficca: A density-based fuzzy imperialist competitive clustering algorithm for intrusion detection in wireless sensor networks. *Measurement*, Elsevier, v. 55, p. 212–226, 2014. Citado 2 vezes nas páginas 37 e 38.

SHAMSHIRBAND, S. et al. Cooperative game theoretic approach using fuzzy q-learning for detecting and preventing intrusions in wireless sensor networks. *Engineering Applications of Artificial Intelligence*, Elsevier, v. 32, p. 228–241, 2014. Citado 2 vezes nas páginas 38 e 39.

SRIVASTAVA, S. S. et al. A survey on mobile agent based intrusion detection system. In: *International Symposium on Devices MEMS, Intelligent Systems & Communication (ISDMISC)*. [S.l.: s.n.], 2011. p. 19–24. Citado na página 28.

SUN, T.; LIU, X. Agent-based intrusion detection and self-recovery system for wireless sensor networks. In: IEEE. *Broadband Network & Multimedia Technology (IC-BNMT), 2013 5th IEEE International Conference on*. [S.l.], 2013. p. 206–210. Citado 4 vezes nas páginas 10, 45, 46 e 67.

SURISETTY, S.; KUMAR, S. Is mcafee securitycenter/firewall software providing complete security for your computer? In: IEEE. *Digital Society, 2010. ICDS'10. Fourth International Conference on*. [S.l.], 2010. p. 178–181. Citado na página 27.

TRIKI, B.; REKHIS, S.; BOUDRIGA, N. Digital investigation of wormhole attacks in wireless sensor networks. In: IEEE. *Network Computing and Applications, 2009. NCA 2009. Eighth IEEE International Symposium on*. [S.l.], 2009. p. 179–186. Citado na página 27.

UNDERSTANDING Land Attacks - Technical Documentation - Support - Juniper Networks. 2015. ⟨https://www.juniper.net/documentation/en_US/junos12.3x48/topics/concept/denial-of-service-network-land-attack-understanding.html⟩. (Acessado em 15/12/2016). Citado na página 27.

WALTERS, J. P. et al. Wireless sensor network security: A survey. *Security in distributed, grid, mobile, and pervasive computing*, CRC Press: Boca Raton, FL, USA, v. 1, p. 367, 2007. Citado na página 26.

WANG, H.-b.; YUAN, Z.; WANG, C.-d. Intrusion detection for wireless sensor networks based on multi-agent and refined clustering. In: IEEE. *Communications and Mobile Computing, 2009. CMC'09. WRI International Conference on*. [S.l.], 2009. v. 3, p. 450–454. Citado 3 vezes nas páginas 41, 66 e 67.

WANG, S.-S. et al. An integrated intrusion detection system for cluster-based wireless sensor networks. *Expert Systems with Applications*, Elsevier, v. 38, n. 12, p. 15234–15243, 2011. Citado na página 37.

WANKHADE, S. R.; CHAVHAN, N. A. A review on data collection method with sink node in wireless sensor network. *International Journal of Distributed and Parallel Systems (IJDPS)*, v. 4, n. 1, 2013. Citado 3 vezes nas páginas 22, 23 e 24.

WEISER, M. The computer for the 21st century. *Scientific American*, Nature Publishing Group, v. 265, n. 3, p. 94–104, 1991. Citado na página 17.

WEISS, G. *Multiagent systems: a modern approach to distributed artificial intelligence*. [S.l.]: MIT press, 1999. Citado na página 31.

XIE, M. et al. Anomaly detection in wireless sensor networks: A survey. *Journal of Network and Computer Applications*, Elsevier, v. 34, n. 4, p. 1302–1325, 2011. Citado 2 vezes nas páginas 17 e 27.

YU, Z.; TSAI, J. J. A framework of machine learning based intrusion detection for wireless sensor networks. In: IEEE. *Sensor Networks, Ubiquitous and Trustworthy Computing, 2008. SUTC'08. IEEE International Conference on*. [S.l.], 2008. p. 272–279. Citado 3 vezes nas páginas 42, 67 e 68.

ZARGAR, G. R.; KABIRI, P. Identification of effective network features to detect smurf attacks. In: IEEE. *Research and Development (SCOReD), 2009 IEEE Student Conference on*. [S.l.], 2009. p. 49–52. Citado na página 27.

Anexo A – Códigos-fonte dos *scripts* utilizados nos testes

1 DHT11.py - Script para captura da temperatura e umidade ambiente e armazenamento em banco de dados

```
# Programa : Sensor de temperatura DHT11 com Raspberry Pi B+
# Autor : FILIPEFLOP

# Carrega as bibliotecas
import Adafruit_DHT
import RPi.GPIO as GPIO
import time
import MySQLdb

# Define o tipo de sensor
sensor = Adafruit_DHT.DHT11

GPIO.setmode(GPIO.BOARD)

# Define a GPIO conectada ao pino de dados do sensor
pino_sensor = 2

# Define o nome do sensor
nome_sensor = "Sensor_1"

# Informacoes iniciais
print ("*** Lendo os valores de temperatura e umidade e acessando o banco
    de dados");

# Conexao ao BD
conn = MySQLdb.connect(host= "192.168.42.1",
                       user="root",
                       passwd="java",
                       db="sensoriamento")
x = conn.cursor()

while(1):
    # Efetua a leitura do sensor
    umid, temp = Adafruit_DHT.read_retry(sensor, pino_sensor);
```

```
    # Caso leitura esteja ok, mostra os valores na tela
35  if umid is not None and temp is not None:
      print ("Temperatura = {0:0.1f}  Umidade = {1:0.1f}\n").format(temp,
    umid);
37    x.execute("INSERT INTO registros (umid, temp, nome_sensor) VALUES (\%s
    , \%s, \%s)",(umid, temp, nome_sensor))
      conn.commit()
39    print ("Aguarde 1 segundo para efetuar nova leitura...\n");
      time.sleep(1)
41  else:
      # Mensagem de erro de comunicacao com o sensor
43    print ("Falha ao ler dados do DHT11 !!!")
```

2 consultaBD.py - Script para consulta aos registros de temperatura e umidade

```
1 # Programa : Script para consulta aos registros de temperatura e umidade
  # Autor : Adaptacao do codigo de autoria de FILIPEFLOP
3 import MySQLdb
  import time
5
  # Conexao ao BD
7
  conn = MySQLdb.connect(host= "192.168.42.1",
9                        user="root",
                         passwd="java",
11                       db="sensoriamento")
  x = conn.cursor()
13
  while(1):
15      x.execute("SELECT umid, temp, nome_sensor from registros")
        for row in x :
17    print row
        print ("Aguarde 10 segundos para efetuar nova leitura...\n");
19      time.sleep(10)
```

Anexo B – Capturas de tela do funcionamento do *framework* ABIDS-WSN

1 Captura de tela da instância do Framework ABIDS-WSN - Início de funcionamento dos agentes

```
jan 03, 2017 9:34:34 PM jade.core.Runtime beginContainer
INFORMAÇÕES: ----------------------------------
    This is JADE 4.4.0 - revision 6778 of 21-12-2015 12:24:43
    downloaded in Open Source, under LGPL restrictions,
    at http://jade.tilab.com/
----------------------------------------
jan 03, 2017 9:34:35 PM jade.imtp.leap.LEAPIMTPManager initialize
INFORMAÇÕES: Listening for intra-platform commands on address:
- jicp://192.168.42.1:1099

jan 03, 2017 9:34:37 PM jade.core.BaseService init
INFORMAÇÕES: Service jade.core.management.AgentManagement initialized
jan 03, 2017 9:34:37 PM jade.core.BaseService init
INFORMAÇÕES: Service jade.core.messaging.Messaging initialized
jan 03, 2017 9:34:37 PM jade.core.BaseService init
INFORMAÇÕES: Service jade.core.resource.ResourceManagement initialized
jan 03, 2017 9:34:38 PM jade.core.BaseService init
INFORMAÇÕES: Service jade.core.mobility.AgentMobility initialized
jan 03, 2017 9:34:38 PM jade.core.BaseService init
INFORMAÇÕES: Service jade.core.event.Notification initialized
jan 03, 2017 9:34:38 PM jade.mtp.http.HTTPServer <init>
INFORMAÇÕES: HTTP-MTP Using XML parser com.sun.org.apache.xerces.internal.jaxp.SAXParserImpl$JAXPSAXParser
jan 03, 2017 9:34:38 PM jade.core.messaging.MessagingService boot
INFORMAÇÕES: MTP addresses:
http://raspberrypi:7778/acc
Agente ACTION [action] iniciando execucao
Agente ATK DETC [attack] iniciando execucao
Agente SNIFFER [sniffer] iniciando captura de pacotes.

jan 03, 2017 9:34:38 PM jade.core.AgentContainerImpl joinPlatform
INFORMAÇÕES: ----------------------------------
Agent container Main-Container@192.168.42.1 is ready.
------------------------------------------------
```

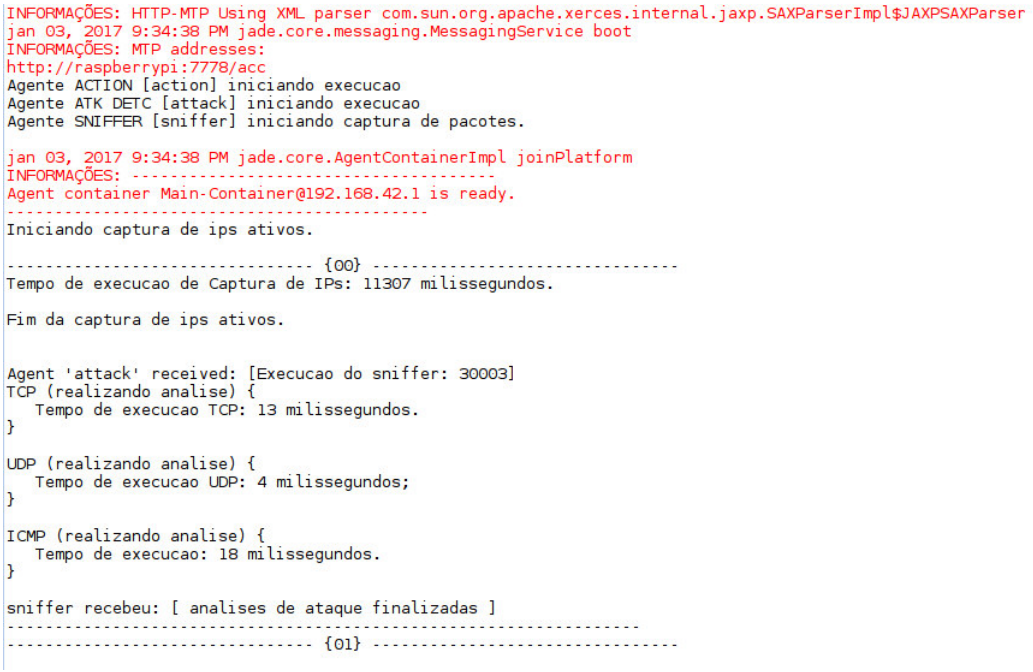
Fonte: O Autor (2017)

2 Captura de tela da instância do Framework ABIDS-WSN - Início e fim da etapa de captura dos endereços IP ativos

```
    at http://jade.tilab.com/
----------------------------------------
jan 03, 2017 9:34:35 PM jade.imtp.leap.LEAPIMTPManager initialize
INFORMAÇÕES: Listening for intra-platform commands on address:
- jicp://192.168.42.1:1099

jan 03, 2017 9:34:37 PM jade.core.BaseService init
INFORMAÇÕES: Service jade.core.management.AgentManagement initialized
jan 03, 2017 9:34:37 PM jade.core.BaseService init
INFORMAÇÕES: Service jade.core.messaging.Messaging initialized
jan 03, 2017 9:34:37 PM jade.core.BaseService init
INFORMAÇÕES: Service jade.core.resource.ResourceManagement initialized
jan 03, 2017 9:34:38 PM jade.core.BaseService init
INFORMAÇÕES: Service jade.core.mobility.AgentMobility initialized
jan 03, 2017 9:34:38 PM jade.core.BaseService init
INFORMAÇÕES: Service jade.core.event.Notification initialized
jan 03, 2017 9:34:38 PM jade.mtp.http.HTTPServer <init>
INFORMAÇÕES: HTTP-MTP Using XML parser com.sun.org.apache.xerces.internal.jaxp.SAXParserImpl$JAXPSAXParser
jan 03, 2017 9:34:38 PM jade.core.messaging.MessagingService boot
INFORMAÇÕES: MTP addresses:
http://raspberrypi:7778/acc
Agente ACTION [action] iniciando execucao
Agente ATK DETC [attack] iniciando execucao
Agente SNIFFER [sniffer] iniciando captura de pacotes.

jan 03, 2017 9:34:38 PM jade.core.AgentContainerImpl joinPlatform
INFORMAÇÕES: ----------------------------------
Agent container Main-Container@192.168.42.1 is ready.
------------------------------------------------
Iniciando captura de ips ativos.

------------------------------ {00} ------------------------------
Tempo de execucao de Captura de IPs: 11307 milissegundos.

Fim da captura de ips ativos.
```

Fonte: O Autor (2017)

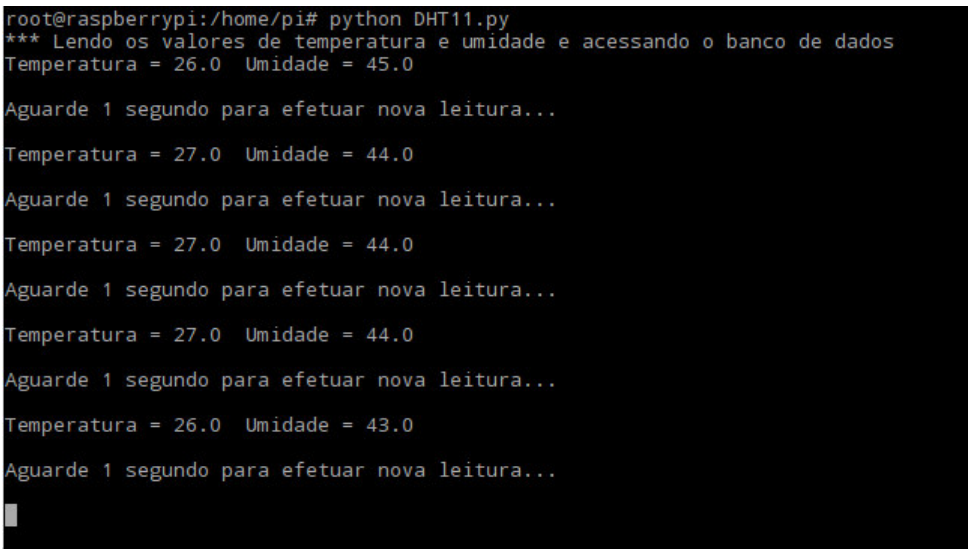
3 Captura de tela da instância do Framework ABIDS-WSN - Primeira rodada de análise dos pacotes



Fonte: O Autor (2017)

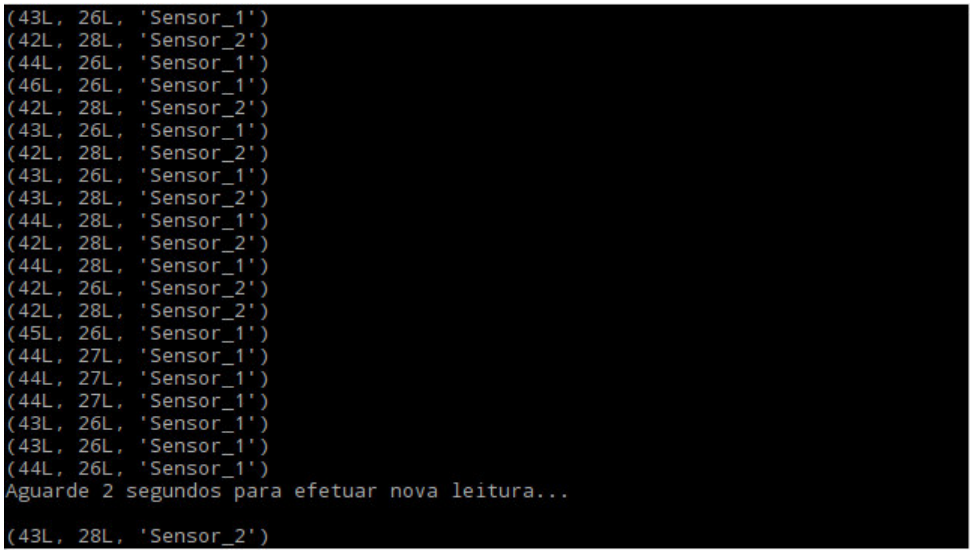
Anexo C – Capturas de tela dos *scripts* utilizados nos testes

1 Captura de tela do script DHT11.py



Fonte: O Autor (2017)

2 Captura de tela do script consultaBD.py



Fonte: O Autor (2017)