Universidade Federal do Maranhão
Centro de Ciências Exatas e Tecnologia
Programa de Pós-Graduação em Engenharia de Eletricidade

Uma arquitetura para a detecção de intrusos no ambiente wireless usando redes neurais artificiais

---

Ricardo Luis da Rocha Ataide

São Luís
2007

Universidade Federal do Maranhão
Centro de Ciências Exatas e Tecnologia
Programa de Pós-Graduação em Engenharia de Eletricidade

Uma arquitetura para a detecção de intrusos no ambiente wireless usando redes neurais artificiais

---

Ricardo Luis da Rocha Ataide

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia de Eletricidade da UFMA como parte dos requisitos necessários para obtenção do grau de Mestre em Engenharia de Eletricidade, Área de Concentração: Ciência da Computação.

São Luís
2007

Ataide, Ricardo Luis da Rocha.

Uma arquitetura para a detecção de intrusos no ambiente wireless usando redes neurais artificiais / Ricardo Luis da Rocha Ataide- 2007.

138f.

Impresso por computador (fotocópia).

Dissertação (Mestrado) - Universidade Federal do Maranhão, Programa de Pós-Graduação em Engenharia de Eletricidade, São Luís, 2007.

1.Segurança de redes (computadores). 2.Redes sem fio - segurança. 3.Redes neurais artificiais. I.Título.

CDU 004.056.53

Uma arquitetura para a detecção de intrusos no ambiente wireless usando redes neurais artificiais

Ricardo Luis da Rocha Ataide

Aprovado em 27/12/2007

BANCA EXAMINADORA

Prof. Zair Abdelouahab (UFMA)
Ph.D. em Ciência da Computação
Orientador

Prof. Omar Andres Carmona Cortes (CEFET-MA)
Dr. em Ciência da Computação e Matemática Computacional
Membro da Banca

Sofiane Labidi (UFMA)
Dr. em Ciência da Computação
Membro da Banca

Prof. Denivaldo Cicero Pavão Lopes (UFMA)
Dr. em Informática
Membro da Banca

"Porque, quando sou fraco, então,

é que sou forte."

2 Coríntios 12:10

“À memória da minha saudosa Vovó Rosa,
que partiu durante a realização deste trabalho.”

Agradecimentos

A Deus, por tudo.

À minha esposa, Tatiana Costa, pelo apoio constante e paciência durante tantas jornadas de trabalho isolado.

A todos os meus familiares, especialmente aos meus pais, Leví Ataide e Tânia Ataide, por terem lançado todas as bases necessárias para o meu crescimento pessoal e profissional.

Ao meu orientador, Prof. Ph.D. Zair Abdelouahab, pelo apoio e confiança em todos os momentos.

Ao Prof. Dr. Denivaldo Lopes, pelos direcionamentos tão necessários nos momentos de indecisão.

A todos os meus colegas e amigos do mestrado, especialmente Irlandino Almeida, Aline Lopes, Helaine Sousa, Johnneth e Adriano, que, em algum momento, contribuíram para a realização deste trabalho.

A todos os que contribuíram ou incentivaram a realização deste trabalho.

Resumo

A maioria dos sistemas de detecção de intrusos para redes *wireless* existentes identificam comportamentos intrusivos apenas tomando como base a exploração de vulnerabilidades conhecidas, comumente chamadas de assinaturas de ataques. Eles analisam a atividade do sistema, observando conjuntos de eventos que sejam semelhantes a um padrão pré-determinado que descreva uma intrusão conhecida. Com isso, apenas vulnerabilidades conhecidas são detectadas, trazendo a necessidade de que novas técnicas de intrusão sejam constantemente adicionadas ao sistema. Torna-se necessária a implementação de um WIDS (*Wireless Intrusion Detection System*) que possa identificar comportamentos intrusivos baseando-se também na observação de desvios do comportamento normal dos usuários, computadores pessoais ou conexões da rede. Esse comportamento normal deve se basear em dados históricos, coletados durante um longo período normal de operação. Este trabalho propõe uma arquitetura para um sistema de detecção de intrusos em redes *wireless* por anomalias, que tem como base a aplicação da tecnologia de redes neurais artificiais, tanto nos processos de detecção de intrusões quanto de tomada de contramedidas. O sistema pode se adaptar ao perfil de uma nova comunidade de usuários, bem como pode reconhecer ataques com características um pouco diferentes das já conhecidas pelo sistema, baseando-se apenas nos desvios de comportamento dessa nova comunidade. Um protótipo foi implementado e várias simulações e testes desse protótipo foram realizadas, para três ataques de negação de serviço. Os testes tiveram o objetivo de verificar a eficácia da aplicação de redes neurais na solução do problema da detecção de intrusos em redes wireless, concentrando seu foco no poder de generalização das redes neurais. Isto garante que o sistema detecte ataques ainda que estes apresentem características ligeiramente diferentes das já conhecidas.

Palavras-Chave: Redes *Wireless*, Detecção de Intrusos, Negação de Serviço, Redes Neurais Artificiais.

Abstract

Most of the existing software of wireless intrusion detection identify behaviors obtrusive only taking as a basis the exploitation of known vulnerabilities commonly called of attack signatures. They analyze the activity of the system, watching sets of events that are similar to a pre-determined pattern that describes an intrusion known. Thus, only known vulnerabilities are detected, leading to the need for new techniques for detecting intrusions be constantly added to the system. It is necessary to implement a wireless IDS that can identify intrusive behaviors also based on the observation of the deflection normal behaviour of the users, hosts or network connections. This normal behaviour should be based on historical data, collected over a long period of normal operation. This present work proposes an architecture for a system to intrusion detection in wireless networks by anomalies, which is based on the application of technology to artificial neural networks, both in the processes of intrusion detection, as making countermeasures. The system can be adapted to the profile of a new community of users, and can recognize attacks with characteristics somewhat different from the already known by the system, relying only on deviations in behaviour of this new community. A prototype has been implemented and various simulations and tests were performed on it, with three denial of service attacks. The tests were to verify the effectiveness of the application of neural networks in the solution of the problem of wireless network intrusion detection, and concentrated its focus on the power of generalization of neural networks. This ensures the system detects attacks though these features slightly different from those already known.

Keywords: Wireless Networks, Intrusion Detection, Denial of Service, Artificial Neural Networks.

Lista de Figuras

1.1 Atuação dos Tipos de IDSs nas Camadas do Modelo OSI 15

2.1 Modo *Ad-hoc*. 22
2.2 Modo Infra-estruturado. 23
2.3 ESS em uma empresa. 24
2.4 Taxonomia de Ataques. 25
2.5 Fluxo de Mensagens da *Shared Key Authentication*. 29
2.6 Visão Conceitual do Servidor de Autenticação na Rede. 36
2.7 Controle de Acesso Baseado em Portas do 802.1X. 36
2.8 Arquitetura do IDPS *Wireless*. 43

3.1 Principais Componentes da Arquitetura Proposta. 57
3.2 Arquitetura Geral do WIDS. 59
3.3 Modelo Relacional da Arquitetura. 60
3.4 Treinamento da RNA do Módulo de Detecção. 61
3.5 Treinamento da RNA do Módulo de Contramedidas. 62
3.6 Operação do Módulo Sensor. 63
3.7 Formato dos Quadros IEEE 802.11b. 64
3.8 Esquema de Agrupamento dos Frames. 65
3.9 Operação do Módulo de Detecção. 66
3.10 Operação do Módulo de Contramedidas. 67
3.11 Operação do Módulo Atuador. 68
3.12 Modelo em Camadas do NIDIA. 70
3.13 Integração do WIDS Proposto ao NIDIA. 73

4.1 Diagrama de Classes do Sensor. 77
4.2 Tela Inicial do Sensor. 79

4.3 Captura de Frames Brutos. 80
4.4 Captura de Frames com Análise de Campos. 81
4.5 Grandezas de Cada Estação. 82
4.6 Esquema Geral do Ambiente de Captura. 84
4.7 Trecho do Arquivo de Dados Normais - Parte I. 84
4.8 Trecho do Arquivo de Dados Normais - Parte II. 84
4.9 Trecho do Arquivo de Dados Normais - Parte III. 85
4.10 Mecanismo *Virtual Carrier Sense*. 86
4.11 Ataque *Virtual Carrier Sense*. 87
4.12 Ataque *De-authentication*. 90
4.13 Fluxograma do Programa de Simulação. 91
4.14 Arquitetura da Rede Neural Usada. 92

5.1 Interpolação e Extrapolação de Funções 96
5.2 Etapas de Realização dos Testes. 96
5.3 Faixas de Valores Para os Ataques. 97

A.1 Modelo de um Neurônio Artificial. 133
A.2 Arquitetura de um Perceptron Multicamada 135

Lista de Tabelas

1.1 Utilização do WEP nas Redes *Wireless* Encontradas. 14

2.1 Sumário das Tecnologias IEEE 802.11. 20
2.2 Comparação entre os IDSs *wireless*. 55

4.1 Grandezas em Cada Registro do tráfego. 83
4.2 Formação dos Registros do Ataque *Virtual Carrier Sense*. 87
4.3 Formação dos Registros do Ataque *Association Flood* - Tipo A. . 88
4.4 Formação dos Registros do Ataque *Association Flood* - Tipo B. . 89
4.5 Formação dos Registros do Ataque *De-authentication*. 90

5.1 Arquivo de Treino para o Ataque *Virtual Carrier Sense*. 98
5.2 Arquivo de Teste para o Ataque *Virtual Carrier Sense*. 99
5.3 Validação do Treinamento para o Ataque *Virtual Carrier Sense*. . 99
5.4 Resultados da Etapa de Interpolação para o Ataque *Virtual Carrier Sense*. 100
5.5 Resultados da Etapa de Extrapolação Inferior para o Ataque *Virtual Carrier Sense*. 101
5.6 Resultados da Etapa de Extrapolação Superior para o Ataque *Virtual Carrier Sense*. 101
5.7 Resultados da Etapa de Extrapolação Geral para o Ataque *Virtual Carrier Sense*. 102
5.8 Resultados da Generalização da Rede Neural para o Ataque *Virtual Carrier Sense*. 103
5.9 Arquivo de Treino para o Ataque *Association Flood*. 103
5.10 Arquivo de Teste para o Ataque *Association Flood*. 103
5.11 Validação do Treinamento para o Ataque *Association Flood*. . . . 104

5.12 Resultados da Etapa de Interpolação para o Ataque *Association Flood*. 105
5.13 Resultados da Etapa de Extrapolação Inferior para o Ataque *Association Flood*. 105
5.14 Resultados da Etapa de Extrapolação Superior para o Ataque *Association Flood*. 106
5.15 Resultados da Etapa de Extrapolação Geral para o Ataque *Association Flood*. 107
5.16 Resultados da Generalização da Rede Neural para o Ataque *Association Flood*. 107
5.17 Arquivo de Treino para o Ataque *De-authentication*. 108
5.18 Arquivo de Teste para o Ataque *De-authentication*. 108
5.19 Validação do Treinamento para o Ataque *De-authentication*. . . . 109
5.20 Resultados da Etapa de Interpolação para o Ataque *De-authentication*.109
5.21 Resultados da Etapa de Extrapolação Inferior para o Ataque *De-authentication*. 110
5.22 Resultados da Etapa de Extrapolação Superior para o Ataque *De-authentication*. 111
5.23 Resultados da Etapa de Extrapolação Geral para o Ataque *De-authentication*. 111
5.24 Resultados da Generalização da Rede Neural para o Ataque *De-authentication*. 112
5.25 Arquivo de Treino para os Ataques 01 e 02. 112
5.26 Arquivo de Teste para os Ataques 01 e 02. 113
5.27 Resultados da Generalização da Rede Neural para os Ataques 01 e 02. 113
5.28 Arquivo de Treino para os Ataques 01 e 03. 114
5.29 Arquivo de Teste para os ataques 01 e 03. 114
5.30 Resultados da Generalização da Rede Neural para os Ataques 01 e 03. 114
5.31 Arquivo de Treino para os Ataques 02 e 03. 115
5.32 Arquivo de Teste para os Ataques 02 e 03. 115

5.33 Resultados da Generalização da Rede Neural para os Ataques 02 e 03. 116
5.34 Arquivo de Treino para os Ataques 01, 02 e 03. 116
5.35 Arquivo de Teste para os Ataques 01, 02 e 03. 116
5.36 Resultados da Generalização da Rede Neural para os Ataques 01, 02 e 03. 117
5.37 Piores Casos em Relação ao Erro Máximo. 118
5.38 Piores Casos em Relação à Taxa de Falsos Positivos. 118
5.39 Piores Casos em Relação à Taxa de Falsos Negativos. 118

B.1 Configurações do AP. 136
B.2 Configurações do PALM. 136
B.3 Configurações do PC01. 137
B.4 Configurações do PC02. 137
B.5 Configurações do PC03. 137
B.6 Configurações do NB01. 138
B.7 Configurações do NB02. 138
B.8 Configurações do NB03. 138

Lista de Abreviaturas e Siglas

AES	Advanced Encryption Standard
AID	Associate Identification
AP	Access Point
API	Application Program Interface
AS	Authentication Server
BSS	Basic Service Set
CIDF	Common Intrusion Detection Framework
CRC-32	Redundância cíclica de 32 bits
CTS	Clear To Send
DDoS	Distributed Denial of Service
DER	Diagrama Entidade-Relacionamento
DoS	Denial of Service
EAP	Extensible Authentication Protocol
ESS	Extended Service Set
FTP	File Transfer Protocol
GHz	Gigahertz
GPL	GNU Public Licence
HMAC	Hash Message Authentication Code
IBSS	Independent Basic Service Set
ICV	Integrity Check Value
IDE	Integraded Development Environment
IDPS	Intrusion Detection and Prevention System
IDS	Intrusion Detection System
IEEE	Institute of Electrical and Eletronics Engineers
IETF	Internet Engineering Task Force
IIDB	Incidents of Intruders and Intrusions Database
IPS	Intrusion Prevention System
ISO	International Standards Organization
IV	Initialization Vector
LAN	Local Area Network
MAC	Medium Access Control

Mbps	Megabit por segundo
MCA	Main Controller Agent
MHz	Megahertz
MiTM	Man-in-the-middle attack
MLP	Multi-layer Perceptron
MMDS	Multi-level Monitoring and Detection System
MSE	Mean Square Error
NAV	Network Allocation Vector
NIC	Network Interface Card
NIDIA	Network Intrusion Detection System based on Intelligent Agents
OSA	Open Systems Authentication
OSI	Open Systems Interconnection
PDA	Personal Digital Assistant
PMK	Pairwise Master Key
PSK	Pré-shared Key
RABD	Reaction Database
RADIUS	Remote Authentication Dial In User Service
RNA	Rede Neural Artificial
RSN	Robust Security Network
RSNA	Robust Security Network Association
RTS	Request To Send
SCA	System Controller Agent
SEA	Security Evaluation Agent
SKA	Shared Key Authentication
SMA	System Monitoring Agent
SSID	Service Set Identifier
STBD	Strategy Database
SUA	System Updating Agent
VPN	Virtual Private Network
WEP	Wired Equivalent Privacy
WIDE	Wireless Intrusion Detection Extensions
WIDS	Wireless Intrusion Detection System
WLAN	Wireless Local Area Network
WPA	Wi-Fi Protected Access

Sumário

1 Introdução **11**

1.1 Cenário e Definição do Problema 12

1.2 Objetivo Geral e Específicos 15

1.3 Organização do Trabalho 16

2 Detecção e Prevenção de Intrusos em Redes *Wireless* **18**

2.1 Introdução 18

2.2 Visão Geral das Redes Wireless 18

2.2.1 Os Padrões IEEE 802.11 19

2.2.2 Componentes e Modelos de Arquiteturas das Redes IEEE 802.11 21

2.3 Visão Geral da Segurança do IEEE 802.11 23

2.3.1 Controle de Acesso e Autenticação 27

2.3.2 Criptografia 30

2.3.3 Integridade de Dados 32

2.3.4 Proteção à Repetição 33

2.3.5 Disponibilidade 33

2.3.6 A Segurança IEEE 802.11i 34

2.4 Sistemas de Detecção e Prevenção de Intrusos para Redes *Wireless* 37

2.4.1 Princípios da Detecção e Prevenção de Intrusos 37

2.4.2 IDPSs *Wireless* 40

2.4.3 Principais IDSs *Wireless* e Ferramentas Existentes 47

2.5 Conclusão 55

3 Arquitetura Proposta e Integração ao NIDIA **56**

3.1 Introdução 56

3.2 Visão Geral da Arquitetura 56
3.3 Funcionamento Geral da Arquitetura 60
3.3.1 Fase de Treinamento 60
3.3.2 Fase de Simulação 63
3.4 Aplicabilidade da Arquitetura ao NIDIA 68
3.4.1 O Projeto NIDIA 68
3.4.2 A Integração da Arquitetura ao NIDIA 72
3.5 Conclusão 74

4 Prototipagem da Solução **75**
4.1 Introdução 75
4.2 Configuração do Ambiente de Captura 76
4.3 Implementação do Sensor de Rádio 77
4.4 Geração do Arquivo de Registros Normais 80
4.5 Geração do Arquivo de Registros de Ataques 85
4.5.1 Ataque *Virtual Carrier Sense* 85
4.5.2 Ataque *Association Flood* 88
4.5.3 Ataque *De-Authentication* 89
4.6 Configuração do Ambiente de Simulação 90
4.7 Implementação do Programa Simulador 91
4.8 Conclusão 94

5 Simulações e Resultados **95**
5.1 Introdução 95
5.2 Divisão dos Testes em Etapas 95
5.3 Ataque 01 (*Virtual Carrier Sense*) 98
5.3.1 Validação do Treinamento 99
5.3.2 Interpolação 99
5.3.3 Extrapolação Inferior 100
5.3.4 Extrapolação Superior 100
5.3.5 Extrapolação Geral 101
5.3.6 Generalização 102
5.4 Ataque 02 (*Association Flood*) 102
5.4.1 Validação do Treinamento 103

5.4.2 Interpolação 104
5.4.3 Extrapolação Inferior 104
5.4.4 Extrapolação Superior 105
5.4.5 Extrapolação Geral 106
5.4.6 Generalização 106
5.5 Ataque 03 (*De-authentication*) 107
5.5.1 Validação do Treinamento 108
5.5.2 Interpolação 108
5.5.3 Extrapolação Inferior 109
5.5.4 Extrapolação Superior 110
5.5.5 Extrapolação Geral 110
5.5.6 Generalização 111
5.6 Ataque 01 com Ataque 02 112
5.7 Ataque 01 com Ataque 03 113
5.8 Ataque 02 com Ataque 03 115
5.9 Ataque 01 com Ataque 02 e Ataque 03 115
5.10 Análise dos Piores Casos 117
5.11 Conclusão 118

6 Conclusões e Sugestões para Trabalhos Futuros 120
6.1 Contribuições 120
6.2 Considerações Finais 122
6.3 Trabalhos Futuros 123

A Visão Geral das Redes Neurais Artificiais 132

B Dispositivos do Ambiente de Captura 136

Capítulo 1

Introdução

As redes sem fio IEEE 802.11, também conhecidas como redes Wi-Fi ou *wireless*, são um padrão de conectividade sem fio para redes locais. Uma combinação de fatores como expectro livre, eficiente codificação de canal e um *hardware* de interface relativamente barato tem tornado tais redes extremamente populares nos últimos anos (Bellardo and Savage, 2003). Os principais sistemas operacionais da atualidade já possuem suporte nativo às redes *wireless*. Por menos de duzentos reais, é possível comprar um ponto de acesso *wireless*, estendendo a conectividade da rede existente por um raio de aproximadamente cem metros. Como resultado disso, as redes *wireless* estão em quase todos os lugares, com seu uso bastante disseminado tanto no ambiente doméstico quanto empresarial.

Segundo relatórios de pesquisas da RSA[1] (RSA, 2007a), (RSA, 2007b), (RSA, 2007c), que tiveram como objetivo analisar o crescimento da utilização e a segurança de redes *wireless* em grandes centros financeiros ao redor do mundo, a cidade de Londres teve um aumento de 160% no número de pontos de acesso em 2007, muito acima dos 57% registrados no ano anterior. Este crescimento explosivo para 7.130 pontos de acesso coloca Londres bem à frente da cidade de Nova Iorque em número de pontos de acessos *wireless*. Nova Iorque mostrou uma taxa de crescimento de quase 49% em 2007, acima dos 20% registrados no ano anterior, enquanto que Paris mostrou um crescimento de 44%. Verificando puramente o número de pontos de acesso corporativos, Londres também lidera. A

[1]RSA Security Inc. (RSA, 2007d), a Divisão de Segurança da EMC Corporation (EMC, 2007).

cidade mostrou um crescimento exponencial de 180% em 2007, se comparado aos crescimentos percentuais de 57% e 45% em Nova Iorque e Paris, respectivamente.

De acordo com (Schiller, 2003), as principais vantagens das redes *wireless* são:

- Flexibilidade - as estações podem se comunicar sem qualquer restrição, dentro da cobertura de rádio. Uma vez que as ondas de rádio podem penetrar paredes, emissores e receptores podem estar localizados em qualquer lugar;
- Planejamento - somente as redes *wireless* ponto a ponto permitem a comunicação sem um planejamento prévio, ao contrário das redes cabeadas tradicionais. Basta que os dispositivos sigam o mesmo padrão, para que os mesmos possam se comunicar;
- Projeto - as redes *wireless* possibilitam projetos de dispositivos móveis pequenos e independentes, que podem ser colocados no bolso, já que estes dispositivos não necessitam de conexões permanentes a qualquer tipo de infra-estrutura para poderem se comunicar. Os cabos restringem não somente os usuários, mas também os projetistas de dispositivos como os PDAs (*Personal Digital Assistants*). Além disso, emissores e receptores podem se comunicar no interior de prédios históricos, ou seja, a tecnologia de rede pode ser introduzida em tais ambientes sem se tornar visível;
- Robustez - as redes *wireless* podem sobreviver a desastres, como terremotos ou enchentes. Se os dispositivos *wireless* se mantiverem íntegros, as pessoas ainda poderão se comunicar. Nessas situações, as redes que requerem uma infra-estrutura cabeada podem ser seriamente comprometidas.

1.1 Cenário e Definição do Problema

Apesar de suas grandes vantagens, o advento das redes *wireless* traz consigo uma série de novas ameaças de segurança, cujo tratamento não pode ser realizado através das contramedidas tradicionais, aplicáveis às redes cabeadas (Yang, Xie and Sun, 2004).

Ao contrário da instalação de uma rede cabeada, que requer um grau significante de conhecimento, tempo e dinheiro, um AP (*Acess Point*) para a rede

wireless é relativamente barato e pode ser instalado facilmente. Isto significa que pontos de acesso não autorizados podem ser instalados sem qualquer autorização por parte do administrador da rede local.

Devido à facilidade de implementação das redes *wireless*, muito frequentemente nenhuma análise de risco é realizada antes que qualquer equipamento *wireless* seja colocado em funcionamento. Na maioria dos casos, pontos de acesso são colocados em uso com suas configurações mínimas de segurança, como ferramentas de identificação e autenticação default, nenhuma criptografia, senha padrão do fabricante, etc.

Além disso, na maioria dos casos, os pontos de acesso *wireless* são conectados à rede cabeada em algum ponto atrás do *firewall* corporativo, quebrando assim a política de segurança da rede da empresa. Convém destacar que a maioria das empresas não têm ainda incluído as redes *wireless* em suas políticas de segurança de rede.

A natureza dos ambientes sem fio os torna bastante vulneráveis a ataques, devido às características físicas do enlace de rádio. Como a disseminação das ondas *wireless* não está restrita aos cabos, a existência de redes *wireless* é muito fácil de determinar e de se conectar. Com isso, indivíduos que meramente escutam as ondas de rádio sem autorização, os chamados *eavesdroppers*, podem localizar redes *wireless* fazendo uma varredura por um SSID (*Service Set Identifier*), bem como determinar se a criptografia está sendo usada ou não. Outra possibilidade é a criação de ataques de DoS (*Denial of Service*), visto que as redes podem simplesmente ser inundadas com ruídos estáticos que podem causar até mesmo a sua completa interrupção.

Outra vulnerabilidade que pode ocorrer é que duas estações equipadas com placas de redes *wireless* podem formar uma rede ponto a ponto, sem passar por qualquer ponto de acesso, abrindo assim uma nova brecha de segurança na rede corporativa.

Em (Cansian, Grégio, Sousa e Filho, 2005), um estudo de caso contendo uma análise da situação atual da segurança de redes *wireless* na cidade de São Paulo, Brasil, é apresentado. Neste estudo, os dados foram capturados nas regiões da Avenida Paulista, Avenida Luiz Carlos Berrini e suas imediações, englobando praticamente o centro financeiro do Brasil, bem como a maior concentração de

Tabela 1.1: Utilização do WEP nas Redes *Wireless* Encontradas.

Região	Redes com WEP		Redes sem WEP		TOTAL	
	Quant.	%	Quant.	%	Quant.	%
Av. Paulista e imediações:	26	8.23	89	28.16	115	36.39
Av. Berrini e imediações:	56	17.72	74	23.42	130	41.14
Eixo Juscelino / Brigadeiro:	22	6.96	49	15.51	71	22.47
TOTAL:	104	32.91	212	67.09	316	100

empresas de alta tecnologia da cidade de São Paulo.

A Tabela 1.1 mostra a distribuição das redes encontradas nas três grandes áreas pesquisadas, destacando a quantidade de redes que utilizam o protocolo criptográfico WEP e as que não o utilizam.

Do total geral de 316 redes sem fio encontradas, foi registrado que 212 delas não estavam usando o protocolo de criptografia WEP, representando assim 67.09% do total de redes apuradas nas áreas pesquisadas.

Outra informação relevante obtida é que considerando o total de redes encontradas, 32 delas estavam com a configuração padrão do fabricante, ou com somente uma pequena modificação de troca do canal de comunicação utilizado. Isto representa 10.13% do total de redes encontradas nas regiões pesquisadas.

Um IDS (*Intrusion Detection System*) é uma ferramenta efetiva para determinar se usuários não autorizados estão tentando acessar, já acessaram, ou até mesmo comprometeram a rede de computadores (NIST, 2002). Os IDSs convencionais concentram seu foco nas camadas mais altas da pilha de protocolos do modelo OSI (*Open Systems Interconnection*) da ISO (*International Standards Organization*). Esses sistemas podem, por exemplo, procurar anomalias de protocolos em pacotes FTP, identificar uma instância de um vírus ou *worm* particular, como os famosos *Code Red* e *Nimda*, ou até mesmo identificar varreduras de portas ou *flooding* de tráfego na rede. Já um IDS voltado para as redes *wireless* (WIDS) não tenta realizar essas tarefas, mas concentra seus esforços na identificação de problemas nas camadas 1 e 2 do modelo OSI, que são as camadas física e de enlace, como mostra a Figura 1.1 (NSA, 2005).

A maioria dos WIDSs existentes, apresentada no Capítulo 2, identifica comportamentos intrusivos apenas tomando como base a exploração de vulnerabilidades conhecidas, comumente chamadas de assinaturas de ataques. Eles analisam a ati-

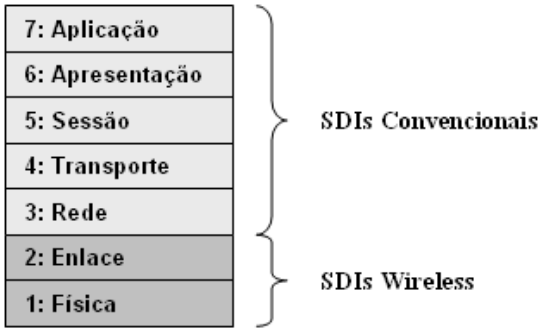

Figura 1.1: Atuação dos Tipos de IDSs nas Camadas do Modelo OSI

vidade do sistema, observando conjuntos de eventos que sejam semelhantes a um padrão pré-determinado que descreva uma intrusão conhecida. Com isso, apenas vulnerabilidades conhecidas são detectadas, trazendo a necessidade de que novas técnicas de intrusão sejam constantemente adicionadas ao sistema.

Nesse contexto, torna-se necessária a implementação de um WIDS (*Wireless Intrusion Detection System*) que possa identificar comportamentos intrusivos baseando-se também na observação de desvios do comportamento normal dos usuários, hosts ou conexões da rede. Esse comportamento normal deve se basear em dados históricos coletados durante um longo período de operação normal. Com isso, o sistema poderá se adaptar ao perfil de uma nova comunidade de usuários, bem como poderá reconhecer tipos de ataques que não foram previamente cadastrados, baseando-se apenas nos desvios de comportamento dessa nova comunidade.

1.2 Objetivo Geral e Específicos

O objetivo deste trabalho é propor um modelo para um sistema de detecção de intrusos em redes locais sem fio (padrão IEEE 802.11), utilizando o método da detecção por anomalias com a aplicação de redes neurais artificiais.

Uma rede neural artificial é um processador distribuído massivamente paralelo constituído de unidades simples de processamento, que têm uma tendência natural para armazenar conhecimentos baseados na experiência e torná-los disponíveis para o uso posterior (Haykin, 1998). Ela assemelha-se ao cérebro humano nos seguintes aspectos:

- O conhecimento é adquirido pela rede através do seu ambiente através de um processo de aprendizagem;
- Os pesos das conexões entre os neurônios, conhecidos como pesos sinápticos, são usados para armazenar o conhecimento adquirido.

Os objetivos específicos deste trabalho são:

- Propor um modelo de detecção de intrusos por anomalias, utilizando redes neurais artificiais. A aplicação de redes neurais artificiais se deve à sua eficiência e excelente capacidade de generalização;
- Integrar o modelo proposto ao modelo do NIDIA (*Network Intrusion Detection System based on Intelligent Agents*), que é um sistema de detecção de intrusos baseado em agentes inteligentes que está sendo desenvolvido na Universidade Federal do Maranhão (Oliveira, 2006). O NIDIA é capaz de gerar um índice de suspeita de ataque a partir da análise de dados de logs de máquinas e de datagramas IP capturados em redes Ethernet (padrão IEEE 802.3);
- Propor a arquitetura geral do sistema;
- Realizar a implementação de um protótipo para o modelo proposto;
- Realizar simulações e testes de modo a comprovar a eficácia da solução proposta.

1.3 Organização do Trabalho

Esta dissertação está organizada em seis capítulos.

No capítulo 1, uma visão geral das redes *wireless* no mundo atual, de suas vulnerabilidades e dos sistemas que buscam torná-las menos inseguras, é apresentada. Também são apresentados os objetivos da realização do presente trabalho.

No capítulo 2, o contexto de estudo desta dissertação é apresentado, suprindo uma visão geral do padrão IEEE 802.11 e alguns dos seus aspectos de segurança, como vulnerabilidades, ataques, ferramentas e os principais sistemas existentes para a detecção de intrusos no ambiente *wireless*.

No capítulo 3, a proposta de uma arquitetura para um sistema de detecção de intrusos em redes *wireless* é apresentada, que tem como base a aplicação da tecnologia de redes neurais artificiais, tanto nos processos de detecção de intrusões quanto de tomada de contramedidas.

No capítulo 4, a implementação de um protótipo para a arquitetura proposta é apresentada, na qual foram implementados o dispositivo sensor e o mecanismo de detecção de intrusões usando redes neurais.

No capítulo 5, as simulações e testes realizados com o protótipo são apresentados, bem como os resultados encontrados.

No capítulo 6, as considerações finais da dissertação são apresentadas, destacando as contribuições do trabalho realizado e sugestões para trabalhos futuros.

Capítulo 2

Detecção e Prevenção de Intrusos em Redes *Wireless*

2.1 Introdução

Este capítulo apresenta uma visão geral do padrão IEEE 802.11 e alguns dos seus aspectos de segurança, como vulnerabilidades, ataques, ferramentas e os principais sistemas existentes para a detecção de intrusos no ambiente *wireless*.

2.2 Visão Geral das Redes Wireless

As redes *wireless* possibilitam dispositivos com interfaces *wireless* usarem recursos computacionais sem estarem fisicamente conectados a uma rede. Os dispositivos simplesmente precisam estar dentro do raio de cobertura (conhecido como extensão) da infra-estrutura da rede *wireless*. Uma WLAN (*Wireless Local Area Network*) consiste de um grupo de nós *wireless* dentro de uma área geográfica limitada, que são capazes de estabelecer comunicação via rádio (NIST, 2007b). WLANs são tipicamente usadas por dispositivos dentro de uma extensão claramente delimitada, como no interior de um edifício, e são geralmente implementadas como extensões às LANs (*Local Area Networks*) existentes para prover uma mobilidade aprimorada aos usuários.

Desde o princípio das redes *wireless*, vários padrões e tecnologias têm sido desenvolvidos para WLANs. A principal organização de padronização que aborda

as redes *wireless* é o IEEE (*Institute of Electrical and Electronics Engineers*) (IEEE, 2007), sendo que esses padrões e a arquitetura das redes 802.11 são descritos nas próximas seções.

2.2.1 Os Padrões IEEE 802.11

As tecnologias de WLANs se tornaram disponíveis inicialmente em 1990, quando os fabricantes começaram a lançar no mercado produtos que operavam na banda de freqüência de 900 MHz (*Megahertz*). Estas soluções, que usavam projetos proprietários e não padronizados, permitiam taxas de transferência de aproximadamente 1 Mbps (*Megabit por segundo*). Isto era significantemente mais lento que os 10 Mbps de velocidade permitidos pela maioria das LANs daquela época. Em 1992, os fabricantes começaram a lançar produtos que usavam a banda de 2.4 GHz (*Gigahertz*). Embora tais produtos permitissem taxas de transferências de dados muito mais altas que os produtos da banda de 900 MHz, eles também usavam projetos proprietários. A necessidade de interoperabilidade entre diferentes marcas de produtos para WLANs levou várias organizações ao desenvolvimento de padrões para redes *wireless*.

Em 1997, o IEEE aprovou o padrão 802.11 para WLANs. O padrão IEEE 802.11 suporta três métodos de transmissão, incluindo a transmissão via rádio na banda 2.4 GHz. Em 1999, o IEEE aprovou duas alterações para o padrão IEEE 802.11, o 802.11a e o 802.11b, que definem os métodos de transmissão via rádio a serem usados. Com isso, os equipamentos para WLANs baseados no 802.11b se tornaram rapidamente a tecnologia *wireless* predominante.

O equipamento IEEE 802.11b transmite na banda 2.4 GHz, oferecendo taxas de dados de até 11 Mbps. O IEEE 802.11b foi pensado para prover performance, *throughput* [1] e ferramentas de segurança comparáveis às das redes locais cabeadas. Em 2003, o IEEE publicou a alteração IEEE 802.11g, que especifica um método de transmissão de rádio que usa a banda 2.4 GHz e pode suportar taxas de dados de até 54 Mbps. Convém destacar que os produtos IEEE 802.11g são perfeitamente compatíveis com os produtos 802.11b. A Tabela 2.1 compara as características básicas dos padrões IEEE 802.11, 802.11a, 802.11b e 802.11g.

[1] *Throughput* ou taxa de transferência é a quantidade de dados transferidos de um lugar a outro, ou a quantidade de dados processados em um determinado espaço de tempo.

Tabela 2.1: Sumário das Tecnologias IEEE 802.11.

Padrão IEEE	Taxa máxima de dados	Extensão típica	Banda de freqüência
802.11	2 Mbps	50 - 100 metros	2.4 GHz
802.11a	54 Mbps	50 - 100 metros	5 GHz
802.11b	11 Mbps	50 - 100 metros	2.4 GHz
802.11g	54 Mbps	50 - 100 metros	2.4 GHz

A Tabela 2.1 não inclui todas as correções mais atuais para o IEEE 802.11. Por exemplo, em novembro de 2005 o IEEE aprovou o IEEE 802.11e, que provê aprimoramentos de qualidade de serviço ao IEEE 802.11, melhorando a transferência de conteúdo multimídia. O projeto IEEE 802.11n especifica melhorias para o 802.11 que possibilitarão a taxa de dados de 108 Mbps.

Todas as variantes do Padrão IEEE 802.11 listadas na Tabela 2.1 incluem ferramentas de segurança conhecidas como WEP (*Wired Equivalent Privacy*), que são idealizadas para prover um nível de segurança comparável ao das redes locais cabeadas. No entanto, todas as configurações IEEE 802.11 que confiam no WEP apresentam vários problemas de segurança, que são amplamente conhecidos e documentados. O IEEE admitiu o escopo de tais problemas e desenvolveu estratégias de curto e longo prazo para corrigir a situação. Em junho de 2004, o IEEE finalizou a alteração 802.11i, que é justamente projetado para superar as deficiências do WEP. O IEEE 802.11i especifica componentes de segurança que trabalham em conjunto com todos os padrões de rádio 802.11, tais como 802.11a, 802.11b e 802.11g.

Enquanto o IEEE estava examinando as deficiências da segurança do IEEE 802.11 e iniciando o desenvolvimento da correção 802.11i, um consórcio industrial sem fins lucrativos constituído de fabricantes de software e equipamentos para WLANs, denominado Wi-Fi Alliance (WI-FI Alliance, 2007), desenvolveu um programa de certificação de interoperabilidade para produtos para WLANs. A Wi-Fi Alliance percebeu que era necessário criar uma solução temporária que pudesse ser empregada usando o *hardware* IEEE 802.11 existente, enquanto o IEEE trabalhava na finalização da alteração 802.11i. Com isso, a Alliance criou o WPA (*Wi-Fi Protected Access*), que foi publicado em outubro de 2002, e era essencialmente um subconjunto dos requisitos do projeto IEEE 802.11i disponível naquele momento.

A diferença mais significante entre o WPA e o projeto IEEE 802.11i é que o WPA não requer suporte para o AES (*Advanced Encryption Standard*), que é um algoritmo de criptografia forte, já que vários componentes de *hardware* 802.11 existentes não podem suportar o processamento de algoritmos de criptografia intensiva, sem o uso de componentes de hardware adicionais.

Juntamente com a aprovação da alteração IEEE 802.11i, a Wi-Fi Alliance introduziu o WPA2, que é o seu termo para equipamentos interoperáveis que são capazes de suportar os requisitos do padrão 802.11i. A Wi-Fi Alliance começou a testar os produtos IEEE 802.11i para a certificação WPA2 logo que a correção IEEE 802.11i foi finalizada.

2.2.2 Componentes e Modelos de Arquiteturas das Redes IEEE 802.11

A arquitetura do IEEE 802.11 tem dois componentes fundamentais:

- STA (*Station*) - É um dispositivo *wireless* que fica no ponto extremo da arquitetura. Típicos exemplos de STAs são *laptops*, PDAs (*Personal Digital Assistants*), telefones móveis e outros dispositivos eletrônicos com interfaces IEEE 802.11;
- AP (*Access Point*) - Conecta logicamente as STAs com um sistema de distribuição, o qual é normalmente a infra-estrutura cabeada de uma organização. Os APs podem também conectar STAs *wireless* entre si sem acessar um sistema de distribuição.

O padrão IEEE 802.11 também define duas estruturas de projeto ou configurações para WLANs:

- Modo *ad-hoc* - Não usa APs. Apenas a comunicação ponto-a-ponto é realizada entre as STAs;
- Modo infra-estruturado - Um AP conecta STAs entre si ou para um sistema de distribuição. É o modo mais comumente usado para WLANs.

A topologia do modo *ad-hoc* está representada conceitualmente na Figura 2.1. Este modo de operação, também conhecido como peer-to-peer (*ponto-a-ponto*),

é possível quando duas ou mais STAs são capazes de se comunicar diretamente. A Figura 2.1 mostra três dispositivos comunicando-se diretamente, sem qualquer infra-estrutura. Um conjunto de STAs configuradas no modo *ad-hoc* é conhecido como um IBSS (*Independent Basic Service Set*). O círculo na Figura 2.1 representa o IBSS. É importante considerá-lo como a área de cobertura de rádio dentro da qual as estações podem permanecer em comunicação. Uma propriedade fundamental da IBSS é que ela não define roteamento ou encaminhamento de mensagens, requerendo que cada dispositivo esteja dentro da cobertura de rádio de todos os demais.

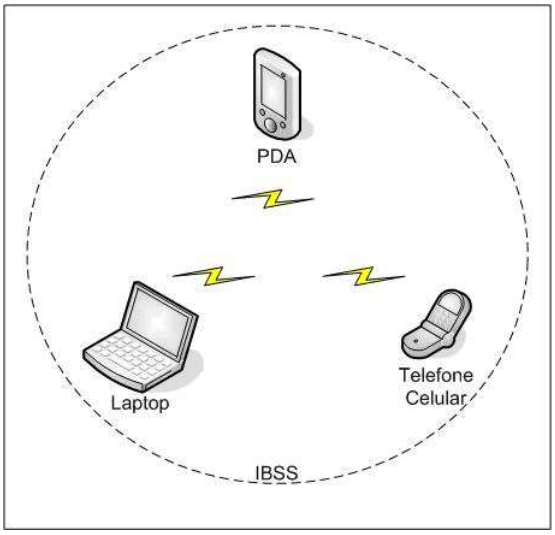

Figura 2.1: Modo *Ad-hoc*.

A principal vantagem de uma WLAN *ad-hoc* é que teoricamente ela pode ser criada a qualquer momento e em qualquer lugar, permitindo que os usuários criem conexões *wireless* de forma barata, rápida e fácil, com uma necessidade mínima de *hardware* de manutenção. Uma rede *ad-hoc* pode ser criada por vários motivos, tais como permitir o compartilhamento de arquivos ou a troca rápida de *e-mails*. Entretanto, uma WLAN *ad-hoc* não pode comunicar-se com redes externas. Outra complicação é que ela pode interferir na operação de um AP de uma rede infra-estruturada, que porventura exista dentro do mesmo espaço *wireless*.

No modo infra-estruturado, uma WLAN IEEE 802.11 compreende um ou mais BSSs (*Basic Service Sets*), que são os blocos básicos de construção de uma WLAN.

Um BSS inclui um AP e uma ou mais STAs. O AP em um BSS conecta as STAs ao sistema de distribuição, que é a maneira pela qual as STAs podem se comunicar com a rede cabeada da organização e com redes externas, como a Internet. O modo infra-estruturado IEEE 802.11 é representado na Figura 2.2.

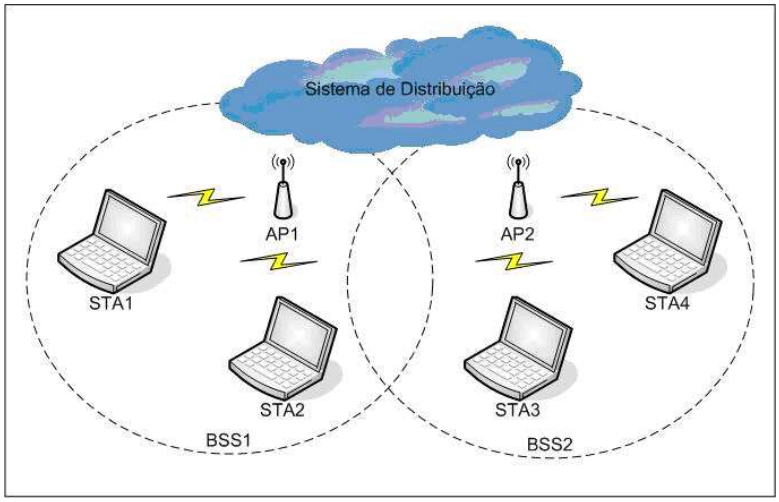

Figura 2.2: Modo Infra-estruturado.

O sistema de distribuição e o uso de múltiplos BSSs e seus APs associados permitem a criação de redes *wireless* de tamanho e complexidade arbitrários. Na especificação IEEE 802.11, este tipo de rede *multi-BSS* é referido como um ESS (*Extended Service Set*). A Figura 2.3 representa conceitualmente uma rede com capacidades tanto cabeadas quanto *wireless*. Ela mostra três APs com seus BSSs correspondentes, o que compreende um ESS. O ESS é ligado à infra-estrutura cabeada. Por sua vez, a infra-estrutura cabeada é conectada através de um *firewall* à Internet. Esta arquitetura pode permitir várias STAs, tais como *laptops* e PDAs, provendo conectividade à Internet aos seus usuários.

2.3 Visão Geral da Segurança do IEEE 802.11

As WLANs precisam suportar vários objetivos de segurança. A intenção é que isso seja alcançado através de uma combinação de ferramentas de segurança embutidas no próprio padrão de redes *wireless*. Os objetivos mais comuns são:

- Confidencialidade - garante que os dados da comunicação não possam ser

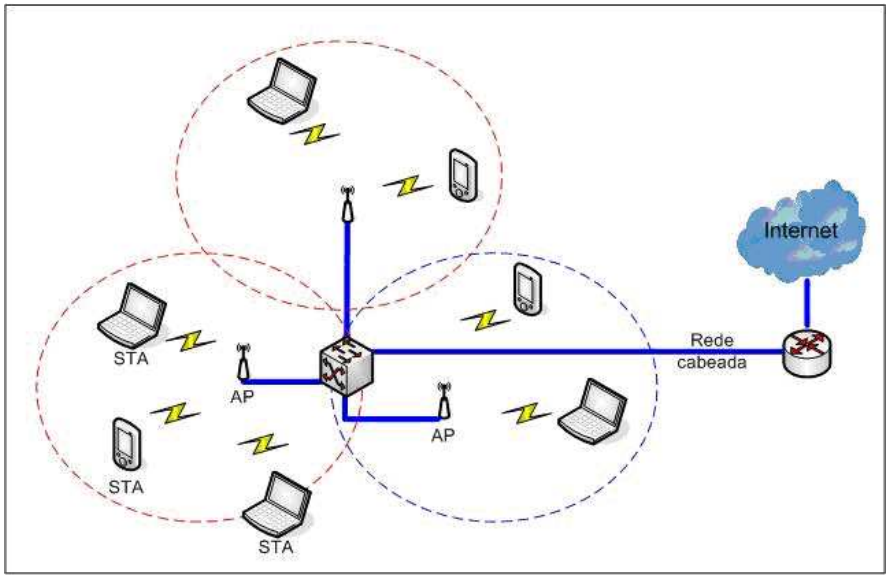

Figura 2.3: ESS em uma empresa.

lidos por indivíduos não autorizados;

- Integridade - detecta qualquer mudança, intencional ou não, que possa ocorrer nos dados durante a transmissão;
- Disponibilidade - garante que dispositivos ou indivíduos possam acessar uma rede e seus recursos sempre que houver necessidade;
- Controle de acesso - restringe o direito de dispositivos ou indivíduos em acessar uma rede ou recursos dentro de uma rede.

Os objetivos de segurança para redes *wireless* e redes cabeadas são os mesmos, assim como as maiores categorias de ameaças e ataques que elas enfrentam. A Figura 2.4 provê uma taxonomia geral dos ataques em redes *wireless*, dividindo-os basicamente entre ataques ativos e ataques passivos (NIST, 2002).

Um ataque passivo é aquele no qual um indivíduo não autorizado obtém acesso à informação, mas não modifica seu conteúdo. Ataques passivos podem ser: escuta ou análise de tráfego, algumas vezes chamado de análise de fluxo de tráfego.

No ataque de escuta, o atacante fica monitorando passivamente as transmissões na rede, em busca do conteúdo das mensagens, incluindo credenciais de autenticação.

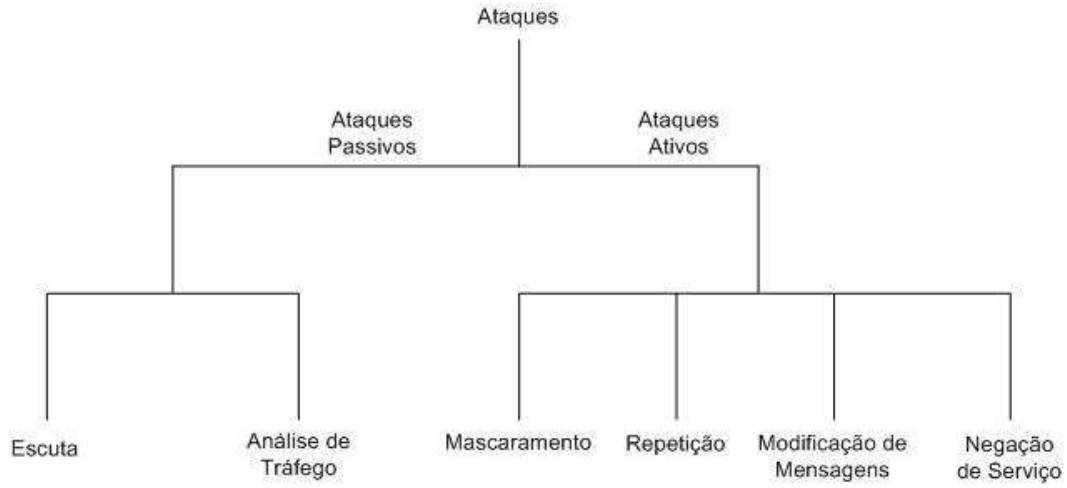

Figura 2.4: Taxonomia de Ataques.

Na análise de tráfego, o atacante obtém informações valiosas sobre a rede, monitorando passivamente as transmissões em busca de padrões de comunicação. Isso é possível porque uma quantidade considerável de informações sobre o sistema é contida nas próprias mensagens que trafegam na rede.

Um ataque ativo é aquele através do qual um indivíduo não autorizado realiza modificações em mensagens, fluxos de informações ou arquivos. É possível detectar esse tipo de ataque, mas pode não ser possível evitá-lo. Os ataques ativos podem tomar a forma de um de quatro ataques (ou uma combinação deles): mascaramento, repetição, modificação de mensagens, negação de serviço.

No ataque de mascaramento, o atacante personifica um usuário autorizado para obter privilégios para os quais não possui autorização.

No ataque de repetição, o atacante fica monitorando passivamente as transmissões e retransmitindo mensagens, como se fosse o usuário legítimo.

No ataque de modificação de mensagens, o atacante altera mensagens legítimas excluindo-as, acrescentando conteúdo, modificando-as ou reordenando-as.

Nos ataques de DoS (*Denial of Service*), o atacante normalmente não rouba informações. Ele simplesmente impede que os usuários acessem os serviços de rede, fazendo com que esses serviços sejam interrompidos ou atrasados. Suas conseqüências podem se estender desde uma redução moderada na performance, até a falha total do sistema. Existe uma variedade de tipos de ataques de DoS

possíveis. Alguns dos mais importantes são: rogue APs, MiTM e roubo de sessão.

Um tipo particular de ataque de DoS são os APs não autorizados (*rogue APs*), que são APs instalados sem o conhecimento ou autorização por parte do administrador da rede. Esse ataque pode ser de dois tipos: interno e externo. No tipo interno, algum usuário da rede instala um AP sem habilitar qualquer mecanismo de segurança, permitindo que qualquer indivíduo com um dispositivo com interface de rede 802.11 possa se conectar na rede corporativa. Isso faz com que a rede fique totalmente aberta e disponível para eventuais intrusos. No tipo externo, um atacante instala na área externa à organização um AP conectado a uma rede falsa, porém com as mesmas configurações de um AP autêntico e da rede interna. Desse modo, os atacantes fazem com que os usuários se conectem ao falso AP, na ilusão de estar usando os serviços da rede verdadeira.

Outro tipo de ataque de DoS é o ataque MiTM (*man-in-the-middle*), no qual o atacante é capaz de ler, inserir e modificar mensagens no caminho da comunicação entre duas partes legítimas da rede, sem que nenhuma delas saiba que o *link* entre elas está comprometido. No contexto de uma WLAN, o ataque MiTM pode ser realizado através de um *rogue* AP, que se parece com um AP autorizado para legitimar usuários.

Outro tipo de ataque de DoS é o roubo de sessão (*session hijacking*), no qual o atacante espera até que um cliente tenha se autenticado na rede, para então enviar-lhe uma mensagem de desautenticação usando o endereço MAC do AP verdadeiro, como se o AP verdadeiro estivesse desautenticando o cliente. Feito isto, o atacante pode começar a transmitir quadros na rede, usando o endereço MAC do cliente que foi desconectado, roubando efetivamente a sua sessão. Na próxima reautenticação, o atacante não poderá ser reautenticado e será banido, necessitando roubar outra sessão válida.

A maioria dos ataques contra WLANs envolve um atacante com acesso ao enlace de rádio entre uma STA e um AP ou entre duas STAs. Vários dos ataques da Figura 2.4 contam com a habilidade do atacante em interceptar e injetar tráfego na rede. Isto evidencia a diferença mais significante entre proteger redes *wireless* e proteger redes cabeadas: a relativa facilidade de interceptar o tráfego da rede e inserir novo tráfego a partir do que pode ser presumido somente como a fonte autêntica. Em uma rede cabeada, o atacante precisa obter acesso físico à rede

ou precisa estar conectado ao sistema remotamente. Em uma rede *wireless*, o atacante simplesmente precisa estar dentro da extensão da infra-estrutura *wireless*. Além disso, ele pode usar antenas direcionais altamente sensíveis, as quais aumentam significativamente a extensão efetiva da rede local *wireless*, para uma cobertura muito além do padrão.

Antes da alteração IEEE 802.11i e seu *framework* para RSNs (*Robust Security Networks*), o IEEE 802.11 apresentava algumas vulnerabilidades de segurança muito sérias. Vários fabricantes adicionaram ferramentas proprietárias às suas implementações para compensar as falhas de segurança do IEEE 802.11, mas como se sabe, ferramentas proprietárias geralmente impossibilitam a interoperabilidade. A seguir, serão abordados os principais aspectos da segurança do padrão 802.11, que são: controle de acesso e autenticação, criptografia, integridade de dados, proteção contra a repetição e disponibilidade.

2.3.1 Controle de Acesso e Autenticação

A especificação original do IEEE 802.11 define duas formas de validar as identidades de dispositivos *wireless* que tentam obter acesso a uma WLAN: OSA (*Open System Authentication*) e SKA (*Shared Key Authentication*). Nenhuma dessas alternativas é segura. Todas as implementações do IEEE 802.11 precisam suportar a OSA, enquanto que o suporte à SKA é opcional.

A OSA é efetivamente um mecanismo de autenticação nula que não provê uma verificação correta de indentidade. Na prática, uma STA é autenticada em um AP simplesmente por prover a seguinte informação:

- SSID (*Service Set Identifier*) para o AP. O SSID é um nome atribuído para uma WLAN. Ele possibilita às STAs distinguirem uma WLAN de outra. SSIDs são disseminados em texto claro (sem criptografia) no meio *wireless*, possibilitando a um *eavesdropper* [2] descobrir facilmente o SSID para uma WLAN.

- MAC (*Media Access Control*) para a STA. O endereço MAC é um valor único de 48 bits atribuído permanentemente a uma interface de rede *wireless*. Várias implementações do IEEE 802.11 permitem ao administrador

[2] *Eavesdropper* é o indivíduo que realiza a escuta não autorizada de mensagens.

especificar uma lista de endereços MAC autorizados. O AP vai fazer com que somente os dispositivos com aqueles endereços MAC listados possam usar a WLAN. Isto é conhecido como filtragem de endereços MAC. Entretanto, como os endereços MAC não são criptografados, é muito simples interceptar o tráfego e identificar os endereços MAC que têm passagem permitida pelo filtro MAC. Infelizmente, quase todos os adaptadores para interfaces de redes *wireless* permitem que aplicações configurem o endereço MAC, o que significa que os atacantes podem obter acesso não autorizado muito facilmente.

Além disso, o AP não é autenticado junto à STA pela OSA. Assim, a STA não tem certeza se está se comunicando com o AP verdadeiro ou com um falso AP, que está usando o mesmo SSID. Por esses motivos, a OSA não provê uma segurança razoável para nenhuma das identidades, podendo ser abusada facilmente para obter acesso não autorizado ou para enganar usuários para que os mesmos se conectem a uma falsa WLAN.

A SKA era supostamente mais segura que a OSA. De fato, ela é igualmente insegura. Como o próprio nome implica, a SKA é baseada em uma chave criptográfica secreta conhecida como chave WEP (*Wired Equivalent Privacy*). Esta chave é compartilhada por STAs e APs legítimos. A SKA usa um esquema simples de desafio-resposta, que verifica se a STA buscando o acesso à WLAN conhece a chave WEP.

Como mostrado na Figura 2.5, a STA inicia uma requisição de autenticação (*Authentication Request*) com o AP, e o AP gera um valor aleatório de 128 bits como desafio e o envia para a STA. Usando a chave WEP, a STA criptografa o desafio e retorna o resultado para o AP. O AP descriptografa o resultado usando a mesma chave WEP e permite o acesso da STA somente se o valor descriptografado corresponde ao desafio lançado inicialmente. Os cálculos criptográficos são realizados usando uma cifra de fluxo baseada no algoritmo RC4, que gera um fluxo de chaves que sofre uma operação XOR com um texto simples para formar o texto criptografado.

A SKA também é fraca porque o AP não é autenticado junto à STA, assim não existe garantia de que a STA está se comunicando com um AP legítimo. Além disso, simples esquemas unilaterais de desafio-resposta são bastante conhecidos

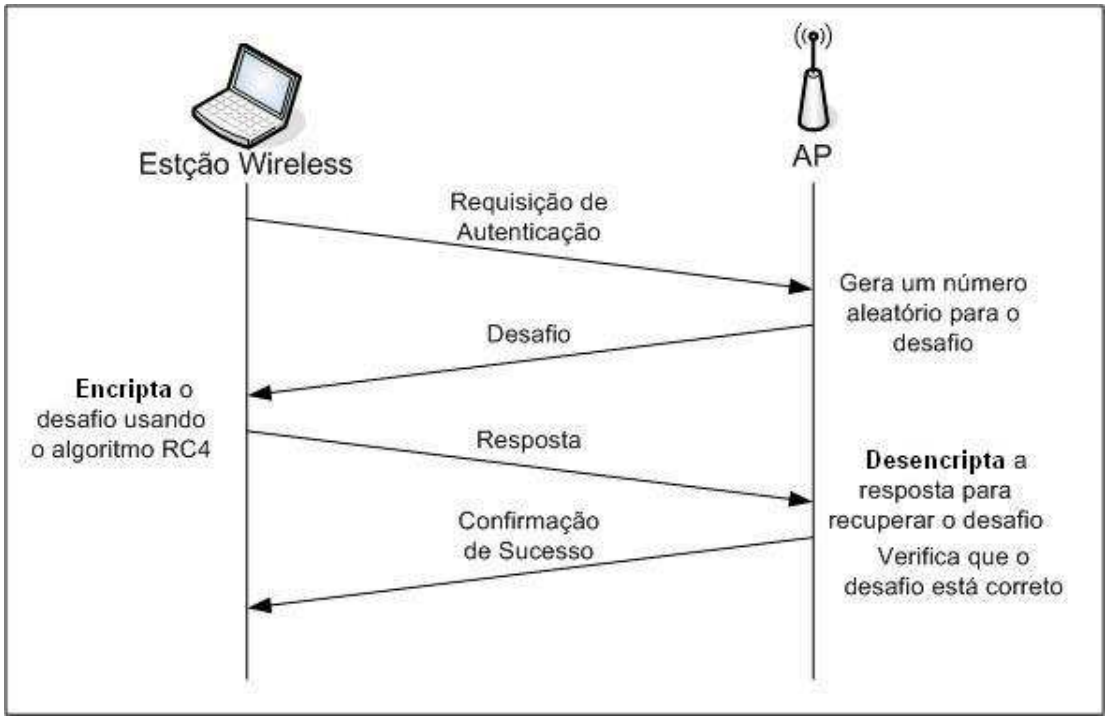

Figura 2.5: Fluxo de Mensagens da *Shared Key Authentication*.

por serem fracos, a não ser que sejam cuidadosamente projetados, com suficiente entropia, chaves de um tamanho apropriado, uma forte função de *hash* e projeto de protocolo seguro. Embora as mensagens de desafio-resposta usadas para SKA possam prevenir a repetição de tráfego de autenticação, o processo de desafio-resposta pode ser comprometido por métodos tais como o ataque MiTM e ataques de dicionário ou força bruta.

Vulnerabilidades adicionais na SKA do IEEE 802.11 são amplamente conhecidas e documentadas. Por exemplo, um atacante pode escutar, capturar e ver o valor do desafio em texto claro (descriptografado) e a respectiva resposta criptografada. O atacante pode então analizar as duas peças de informação para determinar a chave WEP. Algumas organizações preferem usar a OSA porque a SKA provê muito mais informações a *eavesdroppers* acerca da chave WEP, colocando em risco a confidencialidade e integridade que deveria ser garantida através da chave WEP. Outra limitação significante da SKA é que ela autentica a identidade de dispositivos, mas não autentica a identidade de usuários. Se um atacante obtém acesso a uma STA contendo uma chave WEP, ele pode usar aquela mesma

chave em qualquer outro dispositivo com capacidades WEP para se autenticar e obter acesso à WLAN.

Outro dos maiores problemas com SKA é que o IEEE 802.11 requer que todos os dispositivos em uma WLAN usem a mesma chave WEP, ou o mesmo conjunto pequeno de chaves. Se a chave WEP é comprometida, ela precisa ser substituída logo que possível para prevenir novas ações maliciosas, porque as chaves WEP são usadas não somente para o controle de acesso, mas também para proteger a confidencialidade e integridade. Infelizmente, O IEEE 802.11 não especifica qualquer suporte ao gerenciamento de chaves. Quando uma chave WEP precisa ser trocada, os administradores da WLAN têm que implementar seus próprios métodos para gerar e distribuir uma nova chave para todas as STAs e APs em de forma totalmente manual. Os problemas de gerenciamento de chaves frequentemente limitam a escalabilidade de WLANs IEEE 802.11.

Em alguns casos, a SKA é enfraquecida por implementações que usam chaves WEP pobres. Por exemplo, algumas implementações usam chaves WEP padrão que vêm pré-configuradas nos dispositivos, ou configuram uma chave trivial, como todos os dígitos '0' ou todos os dígitos '1'. A chave deve ser gerada aleatoriamente de modo que não seja facilmente adivinhada. Isto vai atrasar atacantes que capturam o tráfego da rede e disparam ataques de dicionário contra ela, tentando encontrar a chave que descriptografa o tráfego com sucesso. As chaves WEP devem ser trocadas frequentemente para reduzir as possibilidades e o impacto de qualquer compromisso da chave.

2.3.2 Criptografia

O protocolo WEP, parte do padrão IEEE 802.11, usa uma cifra de fluxo baseada no algoritmo RC4 para criptografar as comunicações *wireless*, o que protege seus conteúdos de serem descobertos por *eavesdroppers*. O padrão para o WEP especifica o suporte para uma chave WEP de 40 bits, apenas. Entretanto, vários fabricantes oferecem extensões não-padrão ao WEP que suportam tamanhos de chave de até 128 ou até 256 bits. O WEP também usa um valor de 24 bits conhecido como IV (*Initialization Vector*) como um valor semente para inicializar o fluxo de chaves criptográficas. Por exemplo, uma chave WEP de 104 bits com um IV de 24 bits torna-se uma chave RC4 de 128 bits. Idealmente, tamanhos maiores

de chaves se traduzem em uma proteção mais forte, mas a técnica de criptografia usada pelo WEP tem conhecidas falhas que não são mitigadas pelo uso de chaves mais longas.

A maioria dos ataques contra a criptografia WEP têm se baseado nas vulnerabilidades relacionadas ao IV. Por exemplo, a porção IV da chave RC4 é enviada em texto claro, o que permite a um *eavesdropper* que monitora e analisa uma quantidade relativamente pequena de tráfego de rede recuperar a chave, levando vantagem do conhecimento do valor do IV, do espaço de chave IV de 24 bits que é relativamente pequeno, e de uma fraqueza na forma que o WEP implementa o algoritmo RC4.

Além disso, o WEP não especifica precisamente como os IVs devem ser atribuídos ou mudados. Alguns produtos usam valores IV estáticos e bem conhecidos, outros atribuem zero. Se duas mensagens têm o mesmo IV e o texto claro da mensagem é conhecido, é relativamente trivial para um atacante determinar o texto claro da segunda mensagem. Em particular, devido ao fato de várias mensagens conterem cabeçalhos de protocolos comuns ou outros conteúdos de fácil adivinhação, é frequentemente possível descobrir o conteúdo do texto claro original com um mínimo esforço. Mesmo o tráfego proveniente de produtos que usam valores de IV que variam sequencialmente é ainda suscetível a ataques. Existem menos de 17 milhões de valores possíveis de IVs. Em uma WLAN com muito tráfego, o espaço inteiro de IVs pode ser esgotado em poucas horas. Quando um IV é escolhido aleatoriamente, o que representa o melhor algoritmo de seleção de IV genérica possível, pelo *paradoxo de aniversário* dois IVs já apresentam uma chance de colidirem após cerca de 212 frames.

Outra ameaça possível contra a confidencialidade é a análise de tráfego da rede. *Eavesdroppers* podem ser capazes de obter informações monitorando que partes se comunicam em dado momento. Além disso, analisando padrões de tráfego pode ajudar na determinação do conteúdo da comunicação. Por exemplo, curtos períodos de tráfego intenso podem ser causados por uma emulação de terminal ou mensagens instantâneas, enquanto que fluxos fixos de tráfego podem ser gerados por uma vídeo-conferência. Uma análise mais sofisticada pode ser capaz de determinar os sistemas operacionais em uso baseado no comprimento de determinados frames. A não ser em comunicações criptografadas, o IEEE 802.11, como a

maioria dos outros protocolos de rede, não oferece qualquer ferramenta que possa frustrar a análise de tráfego de rede, como a adição de tamanhos aleatórios de conteúdo vazio nas mensagens ou o envio de mensagens adicionais com dados gerados aleatoriamente.

Desse modo, problemas como: tamanho muito pequeno da chave WEP, falhas da técnica de criptografia usada pelo WEP, vulnerabilidades relacionadas ao IV e ausência de mecanismos para frustrar a análise do tráfego da rede tornam o padrão 802.11 totalmente ineficaz com respeito à criptografia das comunicações na rede *wireless*.

2.3.3 Integridade de Dados

O WEP realiza a avaliação da integridade de dados para mensagens transmitidas entre STAs e APs. O WEP é projetado para rejeitar quaisquer mensagens que tenham sido alteradas em trânsito, como por um ataque MiTM. A integridade de dados do WEP é baseada num total de verificação - um código de redundância cíclica de 32 bits (*CRC-32*) que é calculado sobre cada carga útil antes da transmissão. A carga útil e o total de verificação são criptografados usando o fluxo de chave RC4 e são transmitidos. O receptor descriptografa a mensagem, recalcula o total de verificação e o compara com o total de verificação recebido. Se os totais de verificação não forem os mesmos, significa que o quadro foi alterado em trânsito, sendo imediatamente descartado.

O CRC-32 é sujeito a ataques de inversão de bits, o que significa que o atacante sabe que bits vão mudar quando os bits da mensagem forem alterados. O WEP tenta conter esse problema criptografando o CRC-32 para produzir um ICV (*Integrity Check Value*). Os criadores do WEP acreditaram que um CRC-32 cifrado pudesse ser menos sujeito a quebra. Entretanto, eles não perceberam que, em cifras como o RC4 do WEP, a inversão de bits ocorre caso a criptografia seja ou não usada. Assim, o ICV WEP não oferece nenhuma proteção adicional contra a inversão de bits.

A integridade deve ser providenciada por um total de verificação criptografado, ao invés de um CRC. Também conhecidos como *Keyed Hashes* ou MACs (*Message Authentication Codes*), totais de verificação criptografados previnem o ataque de inversão de bits porque são projetados de modo que qualquer mudança

na mensagem original resulte em mudanças significantes e imprevisíveis no total de verificação resultante. CRCs são geralmente mais eficientes computacionalmente do que totais de verificação criptográficos, mas são projetados somente para proteger contra erros aleatórios de bits, não falsificações intensionais, de modo que não provêem o mesmo nível de proteção de integridade.

2.3.4 Proteção à Repetição

A implementação criptográfica não provê nenhuma proteção contra ataques de repetição visto que não inclui ferramentas tais como um contador incremental, *timestamp*, ou outro dado temporal que possa tornar o tráfego repetido facilmente detectável.

2.3.5 Disponibilidade

Indivíduos que não têm acesso físico à infra-estrutura da WLAN podem ainda assim causar uma negação de serviço para a WLAN. Uma das ameaças é conhecida como *jamming*, que envolve um dispositivo emitindo energia eletromagnética nas mesmas freqüências da WLAN. A energia torna as freqüências inutilizáveis pela WLAN, causando uma negação de serviço. O *jamming* pode ser realizado intensionalmente por um atacante ou despropositalmente por um dispositivo que não faz parte da WLAN e que transmite na mesma freqüência. Outra ameaça contra a disponibilidade é o *flooding*, que envolve um atacante enviando um grande número de mensagens para um AP a uma alta taxa que o AP não pode processá-las. Isso faz com que outras STAs não possam acessar o canal, causando uma negação de serviço parcial ou total.

Essas ameaças são difíceis de conter em comunicações via rádio. Infelizmente, o padrão IEEE 802.11 não provê qualquer defesa contra o *jamming* ou o *flooding*. Além disso, os atacantes também podem estabelecer falsos APs. Se as estações se associarem por engano a um falso AP, ao invés de um AP legítimo, isto pode tornar a WLAN efetivamente indisponível aos usuários. Embora o 802.11i proteja os quadros de dados, ele não oferece proteção para quadros de controle ou gerenciamento. Com isso, um atacante pode explorar o fato de que os quadros de gerenciamento não são autenticados, para desautenticar um cliente ou desas-

sociá-lo da rede.

2.3.6 A Segurança IEEE 802.11i

O padrão IEEE 802.11i é a sexta correção aos padrões da base IEEE 802.11. Ele inclui várias melhorias de segurança, que usam tecnologias de segurança maduras e comprovadas. Por exemplo, o IEEE 802.11i faz referência ao padrão EAP (*Extensible Authentication Protocol*), que é uma maneira de prover autenticação mútua entre STAs e a infra-estrutura da WLAN, bem como realizar distribuição automática de chaves criptográficas. O EAP é um padrão desenvolvido pelo IETF (*Internet Engineering Task Force*) (IETF, 2007). O IEEE 802.11i emprega práticas criptográficas amplamente aceitas, tais como a geração de totais de verificação criptográficos através de HMAC (*Hash Message Authentication Codes*).

A especificação IEEE 802.11i introduz o conceito de RSN (*Robust Security Network*), que é definida como uma rede de segurança que permite somente a criação de RSNAs (*Robust Security Network Associations*). RSNA é uma conexão lógica entre entidades IEEE 802.11 comunicantes, estabelecida através do esquema de gerenciamento de chaves do IEEE 802.11i, chamado de *handshake* de 4 vias. O *handshake* de 4 vias é um protocolo que valida ambas as entidades que compartilharm uma PMK (*Pairwise Master Key*), sincroniza a instalação de chaves temporárias e confirma a seleção e configuração de protocolos de confidencialidade de dados e integridade. As entidades obtêm o PMK em uma de duas formas - ou o PMK já está configurado em cada dispositivo, sendo nesse caso chamado de PSK (*Pré-shared Key*), ou é distribuído como resultado de uma instância de autenticação EAP com sucesso, que é um componente do controle de acesso baseado em portas do IEEE 802.1X. O PMK serve como base para os protocolos de confidencialidade e integridade de dados que provêem segurança melhorada sobre o falho WEP. Na maioria dos casos de emprego da tecnologia RSN em grandes empresas é utilizado o 802.1X e o EAP ao invés de PSKs devido à dificuldade de gerenciar PSKs em um grande número de dispositivos. Conexões WLAN empregando o modo *ad-hoc*, que tipicamente envolvem apenas poucas estações, são mais apropriadas para usar o PSK.

Dois componentes definidos no IEEE 802.1X contam com o estabelecimento de

RSNs: servidores de autenticação e o controle de acesso do IEEE 802.1X baseado em portas. O padrão IEEE 802.1X provê um *framework* para o controle de acesso que emprega o EAP para prover uma autenticação mútua centralizada. O IEEE 802.1X foi originalmente desenvolvido para redes locais cabeadas para prevenir o uso não autorizado em ambientes abertos, tais como campus universitários, mas tem sido usado pelo IEEE 802.11i para WLANs também. O *framework* do IEEE 802.1X provê a maneira de bloquear o acesso do usuário até que a autenticação tenha sucesso, desse modo controlando o acesso aos recursos da WLAN.

O padrão IEEE 802.1X define vários termos relacionados à autenticação. O autenticador é uma entidade na ponta de um segmento de rede local que facilita a autenticação da entidade que está na outra ponta do enlace. Por exemplo, o AP na Figura 2.6 serve como um autenticador. O suplicante é a entidade sendo autenticada. A STA pode ser vista como um suplicante. O AS (*Authentication Server*) é uma entidade que provê um serviço de autenticação para o autenticador. Este serviço determina através das credenciais providas pelo suplicante se o mesmo é autorizado a acessar os serviços disponibilizados pelo autenticador. O AS provê esses serviços de autenticação e entrega chaves de sessão para cada AP na rede *wireless*. Cada STA ou recebe as chaves de sessão do AS ou deriva as chaves de sessão por sua própria conta. O AS autentica a STA e o AP, ou provê informações para que os mesmos possam se autenticar mutuamente.

O AS tipicamente fica no sistema de distribuição, como está representado na Figura 2.6. Quando empregando uma solução baseada no padrão IEEE 802.11i, o AS mais frequentemente usado para autenticação é um servidor AAA (*Authentication, Authorization e Accounting*) que usa RADIUS (*Remote Authentication Dial In User Service*) ou protocolo *Diameter* para transportar o tráfego relacionado à autenticação. O modelo suplicante/autenticador é intrinsecamente um modelo unilateral, ao invés de um modelo de autenticação mútua: o suplicante se autentica para acessar a rede. O IEEE 802.11i combate esta propensão requerendo que o método EAP usado providencie a autenticação mútua.

A Figura 2.7 provê uma visão conceitual simples do IEEE 802.1X, que representa todos os componentes fundamentais do IEEE 802.11i: STAs, um AP e um AS. No exemplo da figura considerada, as STAs são os suplicantes e o AP é o autenticador. Até que a autenticação ocorra com sucesso entre a STA e o AS,

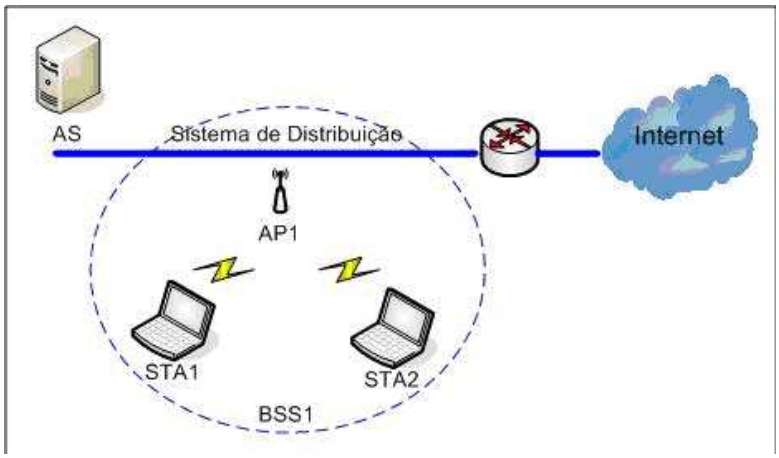

Figura 2.6: Visão Conceitual do Servidor de Autenticação na Rede.

as comunicações da STA são bloqueadas pelo AP. Uma vez que o AP se situa na fronteira entre a rede *wireless* e a rede cabeada, isto evita que uma STA não autenticada alcance a rede cabeada. A técnica usada para bloquear a comunicação é conhecida como controle de acesso baseado em porta. O IEEE 802.1X pode controlar fluxos de dados através da distinção entre quadros EAP e não-EAP, passando quadros EAP através de uma porta não controlada e quadros não-EAP através de uma porta controlada, que pode bloquear o acesso. O IEEE 802.11i estende isto para bloquear a comunicação do AP até que as chaves estejam no lugar também.

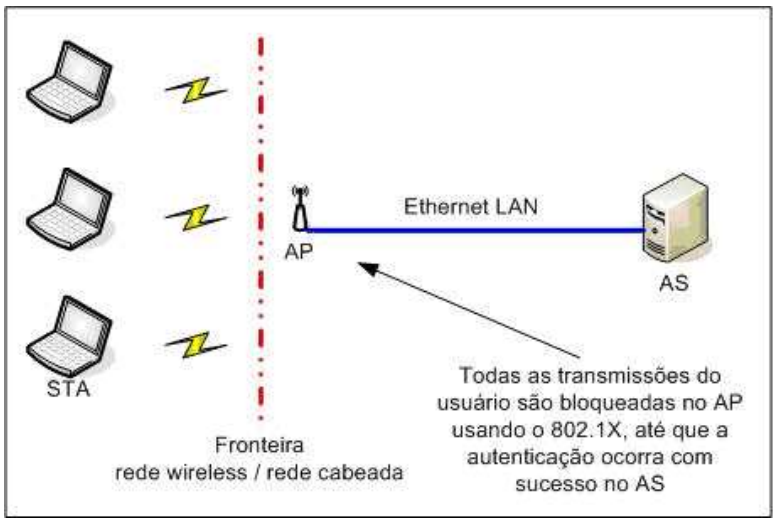

Figura 2.7: Controle de Acesso Baseado em Portas do 802.1X.

2.4 Sistemas de Detecção e Prevenção de Intrusos para Redes *Wireless*

2.4.1 Princípios da Detecção e Prevenção de Intrusos

A detecção de intrusos é o processo de monitoramento de eventos ocorrendo em um sistema computacional ou rede e a análise de tais eventos em busca de sinais de possíveis incidentes, que são violações ou ameaças iminentes de violações de políticas de segurança computacional, políticas de uso aceitável, ou práticas padronizadas de segurança (NIST, 2007a). Os incidentes podem ter várias causas, tais como códigos maliciosos, atacantes tentando obter o acesso não autorizado aos sistemas a partir da Internet, e usuários autorizados de sistemas que se aproveitam dos seus privilégios ou tentam obter privilégios adicionais para os quais não estão autorizados. A prevenção de intrusos, por sua vez, é o processo de realizar a detecção de intrusos e tentar parar possíveis incidentes detectados.

Um IDS (*Intrusion Detection System*) é um software que automatiza o processo de detecção de intrusos. Já um IPS (*Intrusion Prevention System*) é um software que tem todas as capacidades de um sistema de detecção de intrusos e pode também tentar parar possíveis incidentes. Tecnologias IDS e IPS oferecem várias das mesmas capacidades, e o administrador pode eventualmente desabilitar ferramentas de prevenção em produtos IPS, fazendo com que os mesmos funcionem simplesmente como IDSs. Consequentemente, por brevidade o termo sistemas de detecção e prevenção de intrusos, IDPS (*Intrusion Detection and Prevention Systems*), será usado por todo o restante deste trabalho para referir-se a ambas as tecnologias.

As tecnologias IDPS usam várias metodologias para detectar incidentes: baseadas em assinaturas, baseadas em anomalias e análise de estado dos protocolos. A maioria das tecnologias IDPS usa múltiplas tecnologias de detecção, ou separadas ou integradas, para prover uma detecção mais ampla e apurada.

Uma assinatura é um padrão que corresponde a uma ameaça conhecida. A detecção baseada em assinaturas é o processo de comparar assinaturas contra eventos observados para identificar possíveis incidentes. Um exemplo de assinatura é um e-mail com o assunto *"Fotos grátis!"* e um arquivo anexo com o nome

"freepics.exe", que constituem características de uma conhecida forma de vírus ou cavalo de tróia.

A **detecção baseada em assinaturas** é muito efetiva na detecção de ameaças conhecidas, mas é ineficaz na detecção de ameaças desconhecidas. Por exemplo, se um atacante modificou o vírus no exemplo anterior para usar o nome de arquivo *"freepics2.exe"*, uma assinatura buscando *"freepics.exe"* não deverá detectá-lo.

A **detecção baseada em anomalias** é o processo de comparar definições de que atividade é considerada normal contra eventos observados para identificar desvios significantes. Um IDPS usando detecção baseada em anomalias possui perfis que representam o comportamento normal de vários elementos, como usuários, *hosts*, conexões de rede ou aplicações. Os perfis são desenvolvidos pelo monitoramento das características da atividade típica em um dado período de tempo.

O maior benefício de métodos de detecção baseados em anomalias é que eles podem ser muito efetivos na detecção de ameaças previamente desconhecidas. Por exemplo, suponhamos que um computador se torne infectado por um vírus. Esse vírus pode consumir recursos de processamento do computador, enviar um grande número de e-mails, iniciar um grande número de conexões de rede e realizar outro comportamento que seja significantemente diferente dos perfis estabelecidos para o computador.

Produtos IDPS baseados em anomalias frequentemente produzem muitos falsos positivos por causa da atividade benigna que se desvia significante dos perfis, especialmente nos ambientes mais diversificados e dinâmicos. Outro problema notável com o uso de técnicas de detecção baseadas em anomalias, é que frequentemente é difícil para os analistas determinarem porque um alerta em particular foi gerado e se o alerta procede e não é um falso positivo, devido à complexidade e quantidade de eventos que podem ter causado a geração do alerta.

A **análise de estado dos protocolos** é o processo de comparar perfis pré-determinados da atividade benigna de cada estado dos protocolos, contra eventos observados no sistema, com o objetivo de identificar desvios. Diferentemente da detecção baseada em anomalias, que usa perfis de host ou de rede, a análise de estado dos protocolos leva em conta os perfis universais desenvolvidos pelo fabricante, que especificam como os protocolos em particular devem ou não ser usados. Desse modo, o IDPS deve ser capaz de entender e rastrear o estado dos protocolos

das camadas de rede, transporte e aplicação que possuem a noção de estado. A análise de estado dos protocolos pode identificar seqüências inesperadas de comandos, tais como a execução do mesmo comando repetidamente ou a execução de um comando sem primeiro executar um comando do qual ele é dependente.

A primeira desvantagem dos métodos de análise de estado dos protocolos é que eles fazem um consumo intensivo de recursos. Isso se deve à complexidade da análise e o *overhead* envolvido na realização do rastreamento de estados para várias sessões simultâneas. Outro problema sério é que os métodos de análise de estado dos protocolos não detectam ataques que não violem as características do comportamento geralmente aceito para os protocolos, como executar várias ações benignas em um curto intervalo de tempo, podendo causar uma negação de serviço. Outro problema é que o modelo do protocolo usado por um IDPS pode entrar em conflito com a forma que o protocolo é implementado em versões particulares de aplicações específicas ou sistemas operacionais, ou mesmo como as diferentes implementações do protocolo interagem.

Baseado no tipo de eventos que podem monitorar e na forma como são desenvolvidas, as tecnologias IDPS podem ser classificadas em:

- Baseada em rede - Monitora o tráfego da rede, para segmentos de rede particulares ou dispositivos, e analisa as atividades dos protocolos das camadas de rede e aplicação para identificar atividade suspeita. Pode identificar vários tipos diferentes de eventos de interesse. É mais comumente empregada na fronteira entre redes, como na proximidade de *firewalls* de borda ou roteadores, servidores de VPNs (*Virtual Private Networks*), servidores de acesso remoto e redes *wireless*;

- *Wireless* - Monitora o tráfego de redes *wireless* e analisa seus protocolos para identificar atividades suspeitas. Não pode identificar atividade suspeita nos protocolos das camadas de rede, transporte ou aplicação em que o tráfego da rede *wireless* está transferindo. É mais comumente empregada dentro da extensão da rede *wireless* de uma organização para monitorá-la, mas pode também ser empregada em locais onde o acesso à rede *wireless* não autorizado possa estar ocorrendo;

- Análise do comportamento da rede - Examina o tráfego da rede para identi-

ficar ameaças que gerem fluxos de tráfego não usuais, tais como um ataque DDoS (*Distributed Denial of Service*), certas formas de códigos maliciosos e violações de política. Sistemas de análise do comportamento da rede são mais frequentemente empregados para monitorar fluxos entre as redes internas de uma organização, mas podem também serem empregados para monitorar fluxos entre as redes de uma organização e redes externas, como a Internet;

- Baseada em *host* - Monitora as características de um único *host* e os eventos ocorrendo naquele *host* em busca de atividades suspeitas. Exemplos de características que um IDPS baseado em *host* pode monitorar são: o tráfego da rede (somente para esse *host*), *logs* do sistema, processos em execução, atividades das aplicações, acesso e modificação de arquivos e mudanças na configuração do sistema e das aplicações. Os IDPSs baseados em *host* são mais comumente empregados em *hosts* críticos, como servidores de acesso público e servidores contendo informações sensíveis.

2.4.2 IDPSs *Wireless*

Os componentes típicos em um IDPS *wireless* são os mesmos que nos IDPSs baseados em rede: consoles, servidores de bancos de dados (opcionais), servidores de gerenciamento e sensores. Todos os componentes, exceto os sensores, têm essencialmente a mesma funcionalidade para ambos os tipos de IDPS. Sensores *wireless* realizam o mesmo papel básico que os sensores de IDPSs baseados em rede, mas funcionam de uma forma muito diferente devido às complexidades do monitoramento das redes *wireless*.

Diferentemente de um IDPS baseado em rede, o qual pode ver todos os pacotes na rede que monitora, um IDPS *wireless* trabalha por tráfego de amostra. Existem duas bandas de freqüência a monitorar (2.4 GHz e 5 GHz), e cada banda é separada em vários canais. Atualmente, não é possível para um sensor monitorar todo o tráfego em uma banda simultaneamente. Um sensor precisa monitorar apenas um canal por vez. Quando o sensor está pronto para monitorar um canal diferente, ele deve desligar sua interface de rádio, mudar de canal e então tornar a ligar sua interface de rádio.

Por quanto mais tempo um único canal for monitorado, torna-se mais provável que o sensor perca atividades maliciosas ocorrendo em outros canais. Para evitar isso, os sensores normalmente trocam de canal frequentemente, o que é conhecido como varredura de canais, de modo que ele possa monitorar cada canal em uma fração de tempo a cada segundo. Para reduzir ou eliminar a varredura de canais, estão disponíveis no mercado sensores especializados que usam várias antenas de rádio de alta potência, com cada par rádio/antena monitorando um canal diferente. Devido a sua altíssima sensitividade, as antenas de alta potência têm também uma extensão de monitoramento bem maior que as antenas regulares. Algumas implementações coordenam padrões de varredura entre os sensores com extensões alternadas, de modo que cada sensor precise monitorar apenas alguns canais.

Sensores *wireless* estão disponíveis em várias formas:

- Dedicados - Um sensor dedicado é um dispositivo que realiza as funções de IDPS *wireless* mas não passa o tráfego de rede da fonte para o destino. Sensores dedicados frequentemente são completamente passivos, funcionando em um modo de monitoramento de RF (*Radio Frequency*) para capturar o tráfego da rede. Alguns sensores dedicados realizam a análise do tráfego que eles monitoram, enquanto que outros sensores redirecionam o tráfego de rede para um servidor de gerenciamento para análise. O sensor é tipicamente conectado à rede cabeada. Sensores dedicados são normalmente projetados para um de dois usos:

  - Fixo - O sensor é empregado em uma localização particular. Tais sensores são tipicamente dependentes da infra-estrutura da organização.
  - Móvel - O sensor é projetado para ser usado em movimento. Por exemplo, o administrador de segurança pode usar um sensor móvel enquanto caminha pelo prédio de uma empresa ou campus para encontrar APs falsos.

- Embutido em um AP - Vários fabricantes têm adicionado capacidades de IDPS aos seus APs. Um AP embutido tipicamente provê uma capacidade de detecção menos rigorosa que um sensor dedicado porque o AP precisa dividir

seu tempo entre prover o acesso à rede e o monitoramento de vários canais ou bandas em busca de atividades maliciosas. Se o IDPS precisa apenas monitorar uma única banda e canal por vez, uma solução embutida pode prover uma disponibilidade de rede e segurança razoáveis. Se o IDPS precisa monitorar várias bandas ou canais, o sensor precisa realizar a varredura de canais, o que pode interromper as funções de AP do sensor tornando-o temporariamente indisponível em sua banda e canal primários.

- Embutido em um switch *wireless* - Switches *wireless* são idealizados para auxiliar o administrador com o gerenciamento e monitoramento de dispositivos *wireless*. Os switches *wireless* tipicamente não oferecem capacidades de detecção tão fortes quanto os APs embutidos ou sensores dedicados.

Visto que os sensores dedicados podem focar apenas na detecção e não precisam transmitir o tráfego *wireless*, eles tipicamente oferecem capacidades de detecção mais robustas do que os sensores *wireless* embutidos em APs ou sensores embutidos em switches *wireless*. Entretanto, sensores dedicados são frequentemente mais caros que sensores embutidos, pois os sensores embutidos podem ser instalados sobre o *hardware* existente, enquanto que os sensores dedicados envolvem tanto *hardware* quanto *software* adicionais.

Alguns fabricantes também disponibilizam *software* para sensores IDPSs *wireless* que podem ser instalados em STAs, como *laptops*. O *software* sensor detecta ataques dentro da extensão das STAs e como más-configurações das STAs, e relata esta informação aos servidores de gerenciamento.

Os componentes de IDPSs *wireless* são tipicamente inter-conectados através de uma rede cabeada, como mostrado na Figura 2.8. Tal como em um IDPS baseado em rede, uma rede separada de gerenciamento ou mesmo a rede padrão da organização pode ser usada para as comunicações entre os componentes do IDPS *wireless*.

A escolha da localização dos sensores para um IDPS *wireless* é um problema fundamentalmente diferente da escolha da localização para qualquer outro tipo de sensor IDPS. Se a organização usa WLANs, os sensores *wireless* devem ser empregados de modo que eles monitorem toda a extensão de RF da WLAN da organização (tanto APs quanto STAs), o que freqüentemente inclui componentes

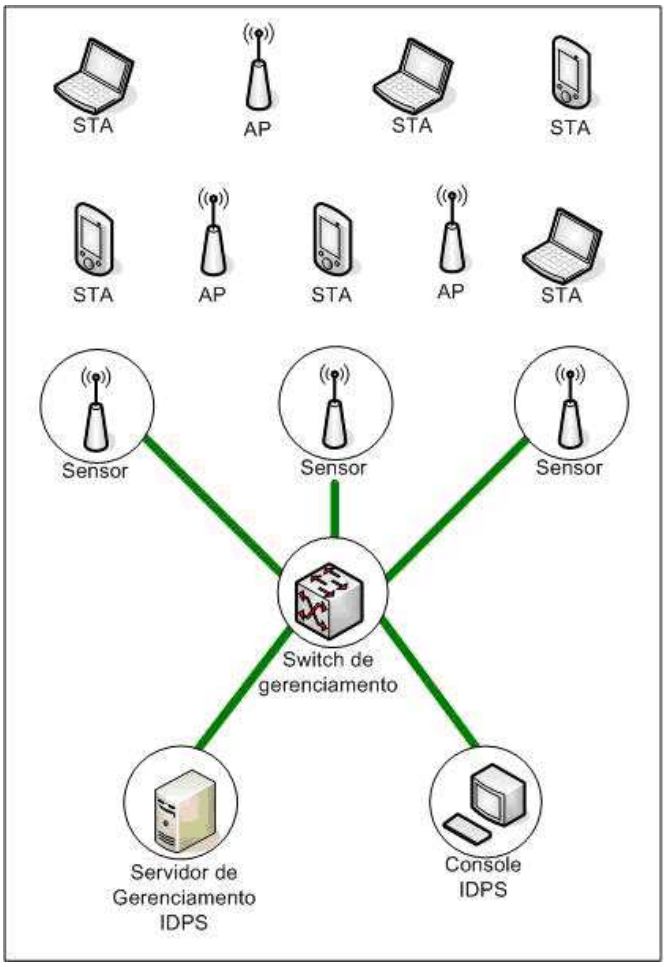

Figura 2.8: Arquitetura do IDPS *Wireless*.

móveis como *laptops* e PDAs. Várias organizações também optam por empregar sensores para monitorar partes de suas instalações onde não pode haver atividade de WLAN, bem como canais ou bandas que a WLAN da organização não deve usar.

IDPSs *wireless* provêem vários tipos de capacidades de segurança. Devido ao fato de que os IDPSs *wireless* são uma forma relativamente nova de IDPS, tais capacidades ainda variam muito entre os produtos. Com o passar do tempo, a tendência é que essas capacidades se tornem mais consistentes. As capacidades de segurança mais comuns são: coleta de informações, registro, detecção e prevenção.

A maioria dos IDPSs *wireless* podem coletar informações em dispositivos *wireless*. Exemplos destas capacidades de coleta de informações são:

- Identificação de dispositivos de WLANs - A maioria dos sensores pode criar e manter um inventário de dispositivos WLANs observados, incluindo APs, clientes de WLANs e clientes de redes *ad-hoc*. O inventário normalmente é baseado no SSID e nos endereços MAC das placas de redes *wireless* dos dispositivos. A primeira porção de cada endereço MAC identifica o fabricante da placa. Alguns sensores podem também usar técnicas de impressões digitais (*fingerprinting*) no tráfego observado para verificar o fabricante, ao invés de confiar na informação do endereço MAC, a qual pode ser forjada. O inventário pode ser usado para identificar novos dispositivos WLAN e também para a remoção de dispositivos existentes;

- Identificação de WLANs - A maioria dos sensores dos IDPSs mantém o rastreamento de WLANs observadas, identificando-as pelo seu SSID. Os administradores podem então marcar cada entrada como sendo uma WLAN autorizada, uma WLAN vizinha benigna, como uma outra organização no mesmo prédio, ou uma falsa WLAN. Esta informação pode ser usada para identificar novas WLANs, bem como para priorizar respostas aos eventos identificados.

Os IDPSs *wireless* tipicamente realizam registros extensivos de dados relacionados aos incidentes detectados. Estes dados podem ser utilizados para confirmar a validade dos alertas, para investigar incidentes, e para correlacionar eventos entre os IDPSs e outras fontes de registro. Campos de dados comumente registrados pelos IDPSs incluem:

- *Timestamp*;
- Tipo de evento ou alerta;
- Índice de prioridade ou severidade;
- Endereço MAC da fonte (o fabricante é frequentemente identificado a partir deste endereço);
- Número do canal;
- ID do sensor que observou o evento;

- Ação de prevenção realizada (se houve).

Os IDPSs *wireless* podem detectar ataques, configurações erradas e violações da política de segurança no nível de protocolo WLAN, primariamente examinando a comunicação nos protocolos IEEE 802.11a, b, g e i. Os IDPSs *wireless* não examinam as comunicações em níveis mais altos, como endereços IP ou carga útil de aplicações.

Alguns produtos realizam somente uma simples detecção baseada em assinaturas, enquanto que outros usam uma combinação de detecção baseada em assinaturas, detecção baseada em anomalias e técnicas de análise de estado dos protocolos. Esta última configuração é a ideal para se alcançar uma detecção mais ampla e apurada.

Os tipos de eventos mais comumente detectados pelos sensores dos WIDSs incluem:

- WLANs ou dispositivos de WLANs não autorizados - Através de suas capacidades de obtenção de informações, a maioria dos sensores de IDPSs *wireless* pode detectar APs falsos, STAs não autorizadas e WLANs não autorizadas, tanto no modo infra-estruturado quanto no modo *ad-hoc*;

- Dispositivos de WLANs com pouca segurança - A maioria dos sensores de IDPSs *wireless* pode indentificar APs e STAs que não estão usando os controles de segurança apropriados. Isto inclui a detecção de configurações erradas e o uso de implementações fracas de protocolos WLANs. Por exemplo, um sensor pode detectar que uma STA está usando o WEP ao invés de WPA2 ou IEEE 802.11i. Os maiores tipos de eventos que podem ser detectados por IDPSs *wireless* caem nesta categoria de detecção;

- Padrões de uso não usuais - Alguns sensores podem usar métodos de detecção baseados em anomalias para detectar padrões de uso de WLANs não usuais. Por exemplo, os sensores podem alertar na ocorrência de várias tentativas de conexão sem sucesso em um curto período de tempo, o que pode indicar uma tentativa de obtenção de acesso não autorizado para a WLAN;

- O uso de *scanners* de redes *wireless* - Tais *scanners* são usados para identificar WLANs inseguras ou com pouca segurança. Sensores de IDPSs *wireless*

podem detectar somente o uso de *scanners* ativos, que são os que geram tráfego na rede *wireless*. Eles não podem detectar o uso de *scanners* passivos que simplesmente monitoram e analizam o tráfego observado;

- Ataques e condições de negação de serviço - Ataques de DoS incluem ataques lógicos, como o *flooding*, que envolve o envio de um grande número de mensagens para um AP a uma alta taxa; e ataques físicos, como o *jamming*, que envolve a emissão de energia eletromagnética nas freqüências da WLAN para tornar tais freqüências inutilizáveis pela WLAN. Ataques de DoS podem frequentemente ser detectados através de análise de estado do protocolo ou métodos de detecção de anomalias;

- Disfarce e ataques MiTM - Alguns sensors de IDPSs *wireless* podem detectar quando um dispositivo está tentando assumir a identidade de outro dispositivo. Isso pode ser feito identificando diferenças nas características da atividade de cada um dos dispositivos, tais como certos valores nos quadros.

A maioria dos IDPSs *wireless* pode indentificar a localização física de uma ameaça detectada através do uso da triangulação - estimando a distância aproximada da ameaça a partir de múltiplos sensores, através da intensidade do sinal da ameaça recebido em cada sensor, e então calculando a localização física da ameaça. Isto permite a uma organização enviar o pessoal de segurança para o local para tratar a ameaça mais efetivamente. Sensores de IDPSs em *handhelds* podem também ser usados para identificar a localização da ameaça, particularmente se sensores fixos não oferecerem capacidades de triangulação ou se a ameaça estiver se movendo.

IDPSs *wireless* oferecem algumas ferramentas de personalização. A maioria possui margens de tolerância que podem ser usados para a detecção baseada em anomalias. "Listas negras"e "listas brancas"podem ser usadas para carregar listas de dispositivos maliciosos conhecidos ou dispositivos benignos da WLAN, respectivamente. As listas podem também ser usadas para gravar NICs (*Network Interface Cards*) de fabricantes não autorizados, e alertas podem ser gerados quando qualquer NIC que não estiver na lista autorizada for usado para APs ou STAs. Alertas individuais podem ser customizados, como pode ser feito para IDPSs baseados em redes. A edição de código não é disponível para a maioria dos produtos,

embora alguns fabricantes permitam aos administradores entrar expressões lógicas complexas para afinar certas capacidades de detecção.

Sensores de IDPSs *wireless* oferecem dois tipos de capacidades de prevenção de intrusos:

- *Wireless* - Alguns sensores podem encerrar conexões entre uma STA maliciosa ou mal configurada e um AP autorizado, ou ainda entre uma STA autorizada e um AP malicioso ou mal configurado. Isto é feito tipicamente pelo envio de mensagens para as extremidades da conexão, notificando os mesmos a se desassociarem da sessão corrente. A partir daí, o sensor passa a rejeitar qualquer tentativa de estabelecimento de uma nova conexão entre esses dispositivos;

- Cabeada - Alguns sensores podem instruir um *switch* na rede cabeada a bloquear toda a atividade de rede envolvendo uma STA particular ou AP, baseando-se no endereço MAC do dispositivo ou porta do *switch*. Por exemplo, se uma STA está disparando um ataque contra um servidor na rede cabeada, um sensor pode direcionar o *switch* para bloquear toda a atividade relacionada à STA. Convém destacar que, apesar de ser efetiva para bloquear as comunicações das STAs ou APs maliciosos na rede cabeada, esta técnica não vai impedir a STA ou o AP de continuar realizando ações maliciosas através da rede *wireless* e seus protocolos.

Uma consideração importante é o efeito que a execução das ações de prevenção podem ter no monitoramento dos sensores. Por exemplo, se um sensor está transmitindo sinais para encerrar conexões, ele pode não ser capaz de realizar a varredura de canais para monitorar outros dispositivos até que ele complete a ação de prevenção em andamento. Para atenuar isso, alguns sensores possuem duas interfaces de rádio - uma para monitoramento e detecção e outra para a realização de ações de prevenção.

2.4.3 Principais IDSs *Wireless* e Ferramentas Existentes

Nesta seção, as principais arquiteturas propostas na área de pesquisa em IDSs *wireless* serão apresentadas, bem como os principais IDSs *wireless* já disponíveis

para uso, além de uma variedade de ferramentas que podem ser empregadas tanto na realização de ataques contra as redes 802.11 quanto na tomada de contramedidas por parte da equipe de segurança da organização.

Em (Pleskonjic, 2003) é proposta uma arquitetura para um WIDS (*Wireless Intrusion Detection System*) que consiste nos seguintes componentes: agente, sensor, console de gerenciamento e ferramentas de relatório. Essa arquitetura é baseada em agentes inteligentes e algumas de suas capacidades, como: auto-aprendizagem, cooperação, autonomia e poder de decisão. Esses agentes são integrados aos clientes da rede, onde realizam a coleta e filtragem local de dados e cooperam com os agentes vizinhos, constituindo assim um módulo de detecção cooperativa. Desse modo, as respostas aos ataques podem ser locais ou globais. A arquitetura também pode ser associada aos sistemas de autenticação e criptografia propostos pelo IEEE 802.11i e 802.1X, para uma maior garantia de segurança. Além disso, novos tipos de ataques podem ser detectados, graças ao poder de auto-aprendizagem da arquitetura, que utiliza técnicas de Inteligência Artificial como Redes Neurais e Lógica Fuzzy.

Em (Yang, Xie and Sun, 2004) é apresentada uma arquitetura distribuída e colaborativa para um sistema de detecção de intrusos *wireless*. Nessa arquitetura, cada nó móvel possui um agente IDS que monitora as atividades locais, incluindo atividades do usuário, do sistema e atividades de comunicação. Esse agente participa ativamente na detecção e resposta a intrusões, sendo responsável por detectar sinais de intrusões localmente e independentemente, colaborando também com seus nós vizinhos, para detectar intrusões em uma extensão maior. O modelo conceitual de cada agente IDS é constituído de um sensor e quatro módulos. Cada módulo representa um agente móvel leve com certas funcionalidades, sendo que alguns desses módulos estão presentes em todos os *hosts* móveis, enquanto que outros estão distribuídos em apenas um grupo selecionado de *hosts* móveis.

Em (Dasgupta, Gómez, González, Kaniganti, Yallapu and Yarramsetti, 2003) é apresentado um sistema multi-agente denominado MMDS (*Multi-level Monitoring and Detection System*) que realiza em tempo real o monitoramento, análise, detecção e geração de respostas a tentativas de intrusão. Esse sistema usa um módulo Fuzzy de suporte a decisões, que utiliza regras para diferentes ataques e

tem como foco a detecção por anomalias tanto em redes *wireless ad-hoc* quanto infra-estruturadas. A modelagem do comportamento é elástica, ou seja, ela se adapta às flutuações normais de uso em função do tempo. O sistema provê um *framework* hierárquico de agentes de segurança, onde cada nó de segurança consiste de quatro agentes: agente de gerenciamento, agente monitor, agente de decisão e agente de ação. As atividades desses agentes são coordenadas pelo agente de gereciamento durante os processos de percepção, comunicação e geração de respostas.

Em (Lim, Schmoyer, Levine and Owen, 2003) é apresentada a implementação de um protótipo para um sistema de detecção e respostas ativas a intrusos *wireless*. A arquitetura desse sistema é constituída de diversos dispositivos espalhados por toda a rede *wireless*, os quais são conectados a um servidor central através da rede cabeada da organização. Visto que esses dispositivos são todos gerenciados pelo servidor central, é possível determinar através de um processo de triangulação a posição aproximada do atacante ou do *rogue* AP, dada a intensidade do sinal recebida em cada dispositivo. O servidor central pode também correlacionar a autenticação *wireless* com a autenticação em outros sistemas de segurança, como a autenticação RADIUS (*Remote Authentication Dial-In User Service*). Um protótipo foi implementado tomando como base a modificação de um ponto de acesso *USRobotics USR2450*, instalando nele um novo sistema operacional Linux com funcionalidades extras de ponto de acesso. O sistema responde ativamente às tentativas de intrusões realizando ataques de DoS contra o intruso, utilizando quadros mal-formados direcionados ao intruso ou confundindo o intruso através do uso de armadilhas (*decoys*).

Em (Schmoyer, Lim and Owen, 2004) é apresentada uma arquitetura para um sistema de detecção e respostas a intrusões *wireless* que utiliza estratégias de respostas adaptativas baseadas em confidência de alarmes, freqüência de ataques, avaliação de riscos e custos estimados de resposta. Nessa arquitetura, cada nó usa um agente IDS para monitorar a atividade local e responder a intrusões. Visto que a atividade local nem sempre provê dados suficientes para detectar ou determinar o tipo de um ataque, os agentes locais devem ser capazes de se comunicar de forma segura e agir coletivamente quando uma intrusão estiver sob suspeita. Um protótipo foi desenvolvido através da criação de uma ferramenta

para detectar ataques e enviar frames de resposta 802.11. O conhecido ataque MiTM foi utilizado como estudo de caso.

Em (Lackey, Roths and Goddard, 2003) é descrita uma arquitetura para o monitoramento de redes *wireless*. Visto que grande parte desta arquitetura é bastante similar a topologias de avaliação de vulnerabilidades e sistemas de detecção de intrusos tradicionais, ela é chamada de WIDE (*Wireless Intrusion Detection Extensions*). A WIDE consiste de três componentes principais: o sensor, o analisador mestre e o adaptador de alertas. Cada sensor é configurado para enviar dados de forma segura para o mestre, para análise. Esse mestre pode residir no próprio sensor, para conservar largura de banda, ou pode residir em uma localização central de modo que a correlação entre múltiplos sensores seja possível. Em ambos os casos, o mestre é configurado com um certo número de módulos de ataques, que são programas independentes que podem ser carregados individualmente no espaço de execução do mestre, para processar dados e gerar alertas. Por exemplo, o módulo de detecção do ataque de DoS usa métodos estatísticos sobre a intensidade do sinal e os níveis de ruído para determinar quando potenciais ataques de DoS estão ocorrendo.

O AirDefense (AirDefense, 2007) é um sistema de detecção e prevenção de intrusos *wireless*, que consiste de sensores dispostos por toda a rede, os quais são interfaceados com uma ferramenta de gerenciamento e administrados por um console de gerenciamento. Ele detecta APs não autorizados e ataques, além de diagnosticar vulnerabilidades potenciais, como más configurações. Além disso, o AirDefense oferece outras funções de gerenciamento tais como rastreamento de falhas e auditoria.

O AirMagnet (AirMagnet, 2007) é uma ferramenta comercial de monitoramento e diagnóstico de rede para Windows e Pocket PC, que roda em *laptops* e *handhelds*. Assim como AirDefense, ele incorpora a detecção de vulnerabilidades e intrusões. Para intrusões, o AirMagnet detecta pontos de acesso e clientes não-autorizados e ataques de DoS por *flooding*. Esse software requer que um técnico se mova ao redor da rede para detectar possíveis ameaças de segurança. Pode ser usado também por um intruso, mas esse uso é pouco provável devido ao seu alto custo.

O Surveyor Wireless (Surveyor, 2007) é uma ferramenta de monitoramento

e análise de redes 802.11, bastante similar ao AirMagnet, que roda em Windows 2000 e XP.

O AirSnare (AirSnare, 2007) é um programa para Windows que funciona como sistema de detecção de intrusos, detectando requisições de DHCP ou endereços MAC não-autorizados tentando se conectar com um ponto de acesso. A resposta à intrusão consiste de uma mensagem de alerta por e-mail para o administrador, a gravação da sessão inteira e, opcionalmente, uma mensagem para o intruso informando que o mesmo está sendo monitorado. Ele é compatível com o *Ethereal*, com o objetivo de oferecer recursos adicionais de análise e rastreamento.

O Snort-Wireless (Snort-Wireless, 2007) é um WIDS *open-source* projetado para se integrar ao ambiente do Snort 2.x (Snort, 2007), que é um IDS para redes cabeadas. Ele permite a criação de regras customizadas, baseadas na estrutura dos pacotes *wireless*, para a detecção de APs não autorizados, *wardrivers* e redes *ad-hoc*.

A Red-M (Red-M, 2007) desenvolveu um IDS *wireless* que monitora todos os serviços baseados em Wi-Fi e *Bluetooth*, identificando falhas de segurança ou fraquezas em sistemas que podem torná-los vulneráveis a ataques. Equipamentos de sonda são instalados em toda a área de cobertura para monitorar e prevenir o acesso não autorizado de intrusos ou atividade de rogue APs. Esses equipamentos são controlados de forma centralizada, e enviam toda a informação e alertas para um servidor de detecção de intrusos. Além disso, o módulo de contramedidas pode interromper e isolar dispositivos intrusos tentando se infiltrar na rede.

O Kismet (Kismet, 2007) é um detector de redes *wireless* 802.11, *sniffer* e sistema de detecção de intrusos, baseado em Linux. Ele trabalha com qualquer placa de interface de rede *wireless* que suporte o monitoramento em modo promíscuo, podendo assim capturar o tráfego em redes 802.11a, 802.11b e 802.11g. O Kismet monitora passivamente o tráfego *wireless*, obtendo dados para identificar SSIDs, endereços MAC, canais e velocidades de conexões, até mesmo de WLANs que não disseminam sinais de *beacon*. Também identifica dados com IVs fracos, os quais podem ser usados por ataques contra a criptografia WEP. O Kismet pode ser usado com agentes remotos capturando dados e enviando-os para um servidor central para correlação e relatórios. Esta é uma arquitetura muito comum para IDSs em geral, de modo que muitas organizações têm empregado o Kismet com

um IDS *wireless* bastante efetivo e extensível.

O NetStumbler (NetStumbler, 2007), também conhecido como Network Stumbler, é um *sniffer* para Windows que possibilita a detecção de WLANs que usam os padrões 802.11a, 802.11b ou 802.11g. Ele envia um quadro de *probe request* para endereço do *broadcast* da rede 802.11, que faz com que todos os APs na área respondam com um quadro de *probe response*, contendo informações de sua configuração de rede, como seu SSID, status WEP, endereço MAC, nome, canal em que está transmitindo, fabricante, tipo e outras informações. Geralmente, é utilizado para comprovar a integridade e o correto funcionamento da WLAN, localizar zonas onde há fraca cobertura, detectar outras redes que possam estar interferindo com WLAN e até mesmo pontos de acesso não autorizados.

Uma versão do NetStumbler está disponível para Windows CE, sendo denominada MiniStumbler (NetStumbler, 2007). Existe também uma ferramenta bastante similar ao NetStumbler, que roda em sistemas BSD, sendo denominada dStumbler (Bsd-airtools, 2007). Ela faz parte do pacote *BSD Air Tools*, que provê um conjunto completo de ferramentas para auditoria de redes *wireless* 802.11b. O MacOS também conta com uma ferramenta bastante similar, denominada MacStumbler (MacStumbler, 2007).

Tanto o NetStumbler quanto o Kismet têm a habilidade de trabalhar em conjunto com sistemas GPS (*Global Positioning System*), para mapear a exata localização de WLANs identificadas. Informações GPS para WLANs podem ser obtidas em (WIGLE, 2007). O Kismet possui uma ferramenta denominada GPSMap, que pode indicar através de mapas a localização física de dispositivos, para que os mesmos possam ser examinados e eventualmente desligados.

O OmniPeek (Omnipeek, 2007) é um analisador de redes *wireless* e cabeadas, com suporte completo aos protocolos 802.11 e a várias placas de interface de redes *wireless*. Ele possui um conjunto poderoso de ferramentas de diagnóstico e resolução de problemas em WLANs, que exibe a taxa de dados, o canal e a intensidade do sinal para cada pacote detectado. O Omnipeek possui também a *Peer Map*, que é uma visão gráfica atualizada continuamente do tráfego entre pares de nós da rede, mostrando volume, protocolo, endereço e tipo de cada nó.

O ISS (*Internet Security Systems*) é um fabricante muito conhecido de uma vasta linha de produtos IDSs para redes cabeadas (ISS, 2007). Sua ferramenta

wireless constitui-se de um *scanner wireless* que detecta pontos de acesso não autorizados e clientes usando conexões fracas.

O Wellenreiter (Wellenreiter, 2007) é uma ferramenta de descoberta, penetração e auditoria de redes 802.11, que usa força bruta para identificar APs com baixo tráfego, integrando-se com GPS. Ele roda em todas as plataformas BSD e POSIX, incluindo o Linux. As placas *wireless* suportadas são: Prism2, Lucent e Cisco.

O THC-RUT (THC-RUT, 2007) é uma ferramenta de descoberta de redes locais para OpenBSD. Ele foi desenvolvido para usar força bruta contra APs 802.11b que usam autenticação MAC, além de oferecer uma variedade de ferramentas de descoberta para redes locais.

O Wireshark (Wireshark, 2007) é um analisador de protocolos de rede capaz de capturar dados em vários tipos de redes, inclusive as redes 802.11, com suporte para centenas de protocolos. Ele roda em várias plataformas, como Windows, Linux, OS X, Solaris, Free BSD e Net BSD. Pode realizar a análise através de uma captura em tempo real, ou através de dados importados a partir de vários formatos de arquivos de captura. Também pode exportar sua saída para os formatos XML, PostScript, CSV ou texto simples.

O AirJack (AirJack, 2007) é um *driver* Linux customizado que dá ao usuário um acesso fácil e completo para a montagem de pacotes 802.11, permitindo-o forjar endereços MAC e injetar quadros de gerenciamento na rede. Essa ferramenta explora justamente os problemas do 802.11b relativos à falta de autenticação dos quadros de gerenciamento. O driver suporta as placas baseadas nos chipsets PRISM2 e Hermes.

O Wlan_Jack, que é uma das ferramentas componentes do AirJack, permite a realização de ataques de DoS de desautenticação de quadros em redes inteiras ou em estações associadas. O Essid_Jack, que é outra das ferramentas componentes do AirJack, possibilita a descoberta de falsos APs, inclusive aqueles com a disseminação de quadros SSID desabilitada. Ele faz isso desautenticando estações e observando seus *probe requests* durante a reconexão. Por fim, o Kracker_jack e o Monkey_jack são duas ferramentas componentes do AirJack que possibilitam a realização de ataques MiTM entre uma estação *wireless* e um servidor VPN WAVESec, e entre duas estações *wireless*, respectivamente.

O SMAC (SMAC, 2007) é uma ferramenta fácil e poderosa para Windows, que permite mudar o endereço MAC da interface *wireless*, possibilitando a geração aleatória de um novo endereço, baseado na seleção de um fabricante.

O AirSnort (AirSnort, 2007) é uma ferramenta que recupera chaves criptográficas em WLANs que usam apenas WEP como método de criptografia. Ele trabalha monitorando as transmissões passivamente e calculando a chave criptográfica assim que um número suficiente de pacotes tenha sido capturado. O AirSnort roda tanto no Windows quanto no Linux, requerendo que a placa de interface de rede *wireless* suporte o monitoramento em modo promíscuo.

O WEPCrack (WEPCrack, 2007) é uma ferramenta open source para quebrar chaves WEP, baseado na implementação do ataque descrito por Fluhrer, Mantin e Shamir (Fluhrer, Mantin and Shamir, 2001). Diferentemente do AirSnort, deve ser usado em conjunto com um *sniffer* de pacotes separado, visto que não possui a habilidade de capturar tráfego de rede.

O Host AP (HostAP, 2007) é um driver Linux para placas de interface de redes *wireless* baseadas nos *chipsets* Intersil Prism2/2.5/3. Esse driver assume as funções de gerenciamento 802.11 na estação em que está instalado, funcionando como um AP.

O Fake AP (FakeAP, 2007) é um programa simples que simula uma lista de APs especificada pelo usuário, através da disseminação de quadros de beacon 802.11b. Isto confunde potencialmente qualquer intruso que estiver escutando a rede passivamente. Ele está disponível sob a GPL (*GNU Public Licence*) e roda em Linux e em algumas versões de BSD.

O AirSnarf (AirSnarf, 2007) é uma ferramenta que configura um sistema Linux com uma placa de interface de rede *wireless* baseada em Prism2, para funcionar como um falso AP. Através do uso de servidores Web e DNS virtuais e de redirecionamento Web, um usuário que se associa com esse falso AP pode receber páginas falsas para portais comuns, como Hotmail ou UOL. Essa completa "experiência"muito provavelmente irá convencer o usuário de estar na página verdadeira e entrar com suas credenciais, como nome de usuário e senha, os quais são automaticamente capturados pelos atacantes.

A Tabela 2.2 apresenta uma análise comparativa entre os IDSs *wireless* apresentados.

Tabela 2.2: Comparação entre os IDSs *wireless*.

	DISPONÍVEL		DETECÇÃO		RESPOSTA	
Referência	Pesquisa	Produto	Assinatura	Anomalia	Ativa	Passiva
(Dasgupta, 2003)	X			X	X	
(Lackey, 2003)	X		X			X
(Lim, 2003)	X		X	X	X	
(Pleskonjic, 2003)	X			X	X	
(Schmoyer, 2004)	X		X		X	
(Yang, 2004)	X		X	X	X	
(AirDefense, 2007)		X	X		X	
(AirMagnet, 2007)		X	X		X	
(AirSnare, 2007)		X	X			X
(Kismet, 2007)		X	X			X
(Red-M, 2007)		X	X		X	
(Snort-Wireless, 2007)		X	X			X
(Surveyor, 2007)		X	X		X	

2.5 Conclusão

Este capítulo apresentou uma visão geral do padrão IEEE 802.11 e alguns dos seus aspectos de segurança, como vulnerabilidades, ataques, ferramentas e os principais sistemas existentes para a detecção de intrusos no ambiente *wireless*. O próximo capítulo apresentará a proposta de uma arquitetura para um sistema de detecção de intrusos em redes *wireless*, que tem como base a aplicação da tecnologia de redes neurais artificiais, tanto nos processos de detecção de intrusões quanto de tomada de contramedidas.

Capítulo 3

Arquitetura Proposta e Integração ao NIDIA

3.1 Introdução

Este capítulo apresenta a proposta de uma arquitetura para um sistema de detecção de intrusos em redes *wireless*, que tem como base a aplicação da tecnologia de redes neurais artificiais, tanto nos processos de detecção de intrusões quanto de tomada de contramedidas. Uma visão geral da arquitetura e do seu funcionamento será apresentada, bem como as principais interações entre cada um dos módulos. Ao final do capítulo, a integração da arquitetura proposta ao Sistema NIDIA será apresentada.

3.2 Visão Geral da Arquitetura

A arquitetura proposta apóia-se em diversas arquiteturas disponíveis, citadas no capítulo anterior, no intuito de buscar as melhores soluções para os problemas enfrentados na implementação de um sistema desse nível. Esta arquitetura emprega a detecção por anomalias como estratégia de análise, buscando identificar comportamentos intrusivos, tomando como base a observação de desvios do comportamento normal dos usuários na rede. Visto que esse comportamento normal se baseia em dados históricos, coletados durante um período normal de operação, torna-se possível que o sistema se adapte ao perfil de qualquer comunidade de

usuários na qual for implantado, além de poder reconhecer tipos de ataques que não foram previamente cadastrados no sistema.

No modelo proposto, a detecção de intrusões e a tomada das respectivas contramedidas são realizadas em tempo real. Isso é possível graças ao poder das redes neurais em determinar o diagnóstico da rede *wireless* e as contramedidas apropriadas de uma forma bastante rápida e eficaz. O modelo permite a tomada de contramedidas tanto ativas quanto passivas, embora tal modelo tenha seu enfoque totalmente voltado para as contramedidas ativas, por serem muito mais efetivas na ocorrência de ataques contra a integridade do sistema, como os ataques de DoS. O Apêndice A desta dissertação apresenta uma visão geral das redes neurais artificiais.

Os principais elementos componentes do modelo são os sensores, os atuadores, o servidor WIDS, o servidor de banco de dados e o console de gerenciamento. Esses componentes são conectados entre si através da rede cabeada da organização, como pode ser visto através da Figura 3.1.

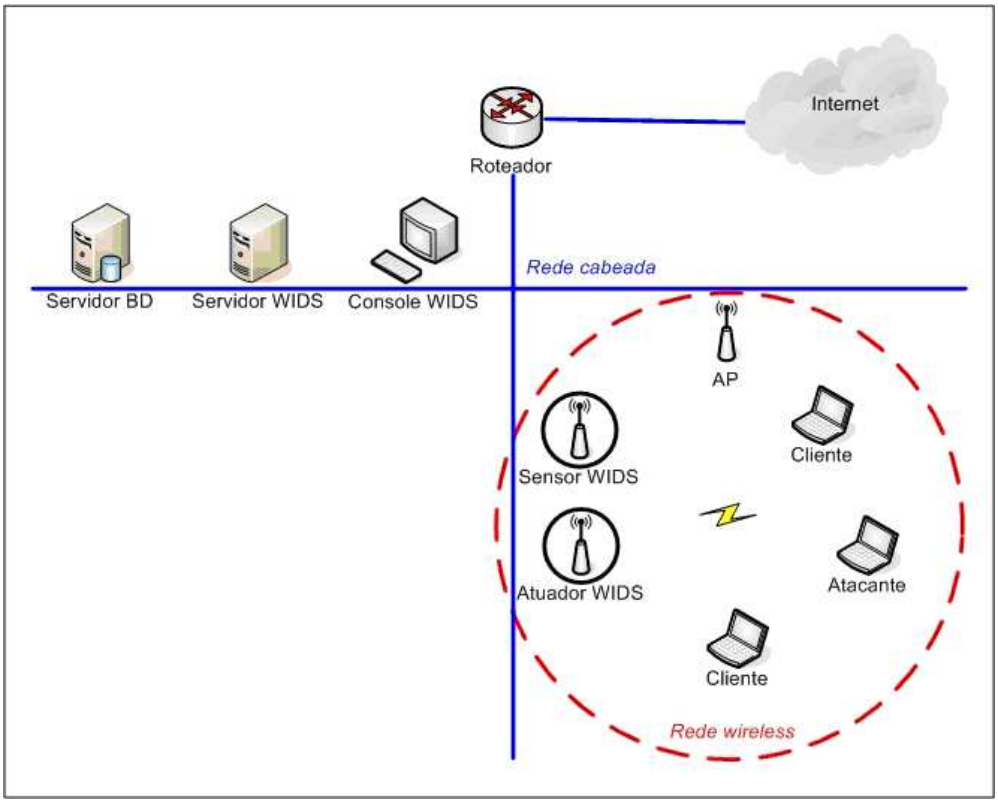

Figura 3.1: Principais Componentes da Arquitetura Proposta.

Tanto os sensores quanto os atuadores são espalhados pelas instalações da organização, buscando cobrir toda a região onde houver sinal da rede *wireless* interna. Essa região está representada na figura anterior através da linha tracejada vermelha. Outro ponto em comum entre os sensores e os atuadores é que ambos possuem pelo menos duas interfaces de rede. Uma delas é para a rede *wireless* e a outra é para a rede cabeada, que é onde se dá a comunicação com os demais elementos do IDS. Isso garante que o IDS tenha disponível uma largura de banda adequada para funcionar mesmo sob condições adversas, como na ocorrência de um ataque de DoS na rede monitorada. Convém destacar que os sensores e atuadores são incapazes de repassar qualquer tráfego entre suas interfaces de rede *wireless* e cabeada, por questões de segurança.

Cada um dos componentes da arquitetura citados anteriormente desempenha as funções referentes a um ou mais módulos do WIDS proposto. Essa organização da arquitetura em módulos pode ser visualizada através da Figura 3.2. Os elementos sensores desempenham as funções do Módulo Sensor. Os elementos atuadores desempenham as funções do Módulo Atuador. O servidor WIDS desempenha as funções do Módulo de Detecção e do Módulo de Contramedidas. O servidor de bancos de dados desempenha as funções do Módulo de BD. Por fim, o console de gerenciamento desempenha as funções do Módulo de Gerenciamento.

O Módulo Sensor é responsável por capturar todo o tráfego da rede *wireless* passivamente, realizar uma pré-formatação sobre esse tráfego capturado, e enviá-lo para o servidor WIDS.

O Módulo de Detecção é responsável por receber as informações enviadas pelos Módulos Sensores e realiza análises sobre essas informações, de modo a tentar identificar atividades intrusivas ocorrendo na rede monitorada. O processo de detecção de intrusões é realizado com o uso de redes neurais artificiais. Caso alguma atividade intrusiva seja detectada, essa informação é enviada para o Módulo de Contramedidas.

O Módulo de Contramedidas é responsável por decidir que contramedidas devem ser tomadas, no intuito de conter em tempo real qualquer intrusão ocorrendo na rede monitorada. Essas contramedidas são mapeadas em ações, que são enviadas para os Módulos Atuadores. O processo de tomada de contramedidas também é realizado com o uso de redes neurais artificiais.

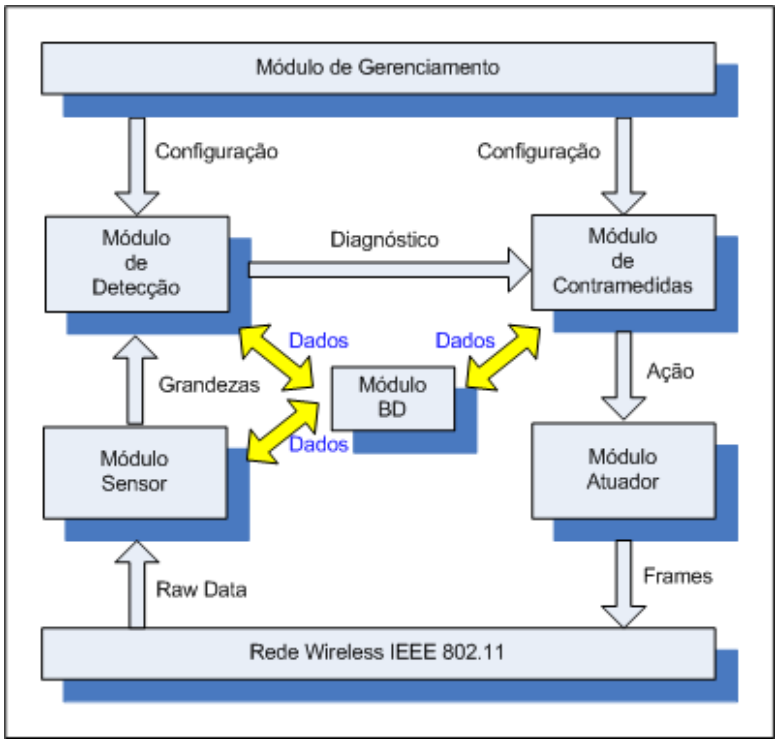

Figura 3.2: Arquitetura Geral do WIDS.

O Módulo Atuador é responsável por executar as ações determinadas pelo Módulo de Contramedidas, no intuito de conter as atividades de intrusão acontecendo na rede monitorada. A execução dessas ações se dá, na maioria dos casos, com o Módulo Atuador injetando tráfego ativamente na rede.

O Módulo de Gerenciamento é responsável por disponibilizar uma interface gráfica para a operação e o gerenciamento do WIDS. Ele permite, entre outras facilidades, a configuração e atualização dos elementos do WIDS, o monitoramento gráfico e análise da rede *wireless* e a geração de relatórios.

O Módulo de Banco de Dados é responsável por gerenciar um repositório para as informações registradas tanto pelos sensores quanto pelo servidor WIDS. Além disso, esse módulo tem um papel fundamental no momento da adaptação do WIDS a um novo ambiente, provendo as informações de entrada para o treinamento das redes neurais.

3.3 Funcionamento Geral da Arquitetura

Todo o funcionamento da arquitetura proposta está baseado na aquisição e emprego de informações provenientes do tráfego de quadros da camada de enlace da rede wireless. Essas informações estão estruturadas de acordo com o Modelo Relacional da Figura 3.3, que é composto pelas seguintes tabelas: Ocorrências, Grandezas, Diagnósticos, Contramedidas, Contramedidas-Ações e Ações. O conteúdo de cada tabela do modelo será detalhado juntamente com o funcionamento de cada módulo da arquitetura.

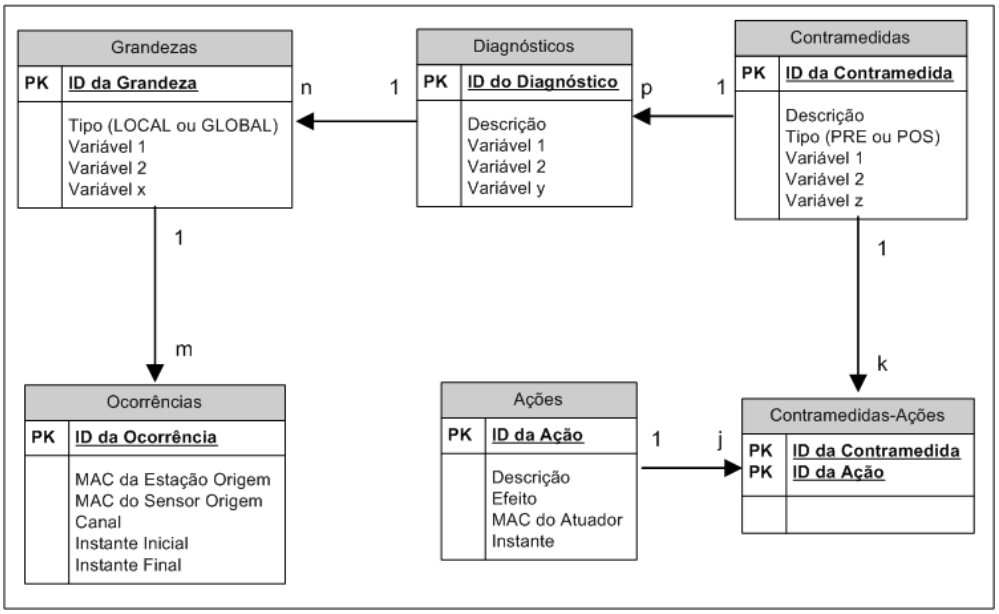

Figura 3.3: Modelo Relacional da Arquitetura.

O funcionamento do sistema é composto basicamente por duas fases: a fase de treinamento e a fase de simulação.

3.3.1 Fase de Treinamento

Na fase de treinamento, o sistema se adapta a uma nova comunidade de usuários, os quais geralmente apresentam um padrão peculiar de uso da rede *wireless*. Isso é necessário para que o sistema possa detectar de forma bastante coerente qualquer desvio no comportamento normal da comunicação na rede, além de manter uma baixa taxa de falsos positivos. A adaptação do sistema se dá com o treinamento

das redes neurais artificiais, que são a parte central dos processos de detecção de intrusos e de tomada de contramedidas.

A Figura 3.4 mostra, através de um diagrama de interação, o treinamento da rede neural do Módulo de Detecção. Inicialmente, o Módulo de Gerenciamento envia para o Módulo de Detecção uma requisição para que o mesmo realize sua adaptação ao novo ambiente. Essa requisição pode ser invocada pelo administrador do sistema através da interface gráfica do Console de Gerenciamento.

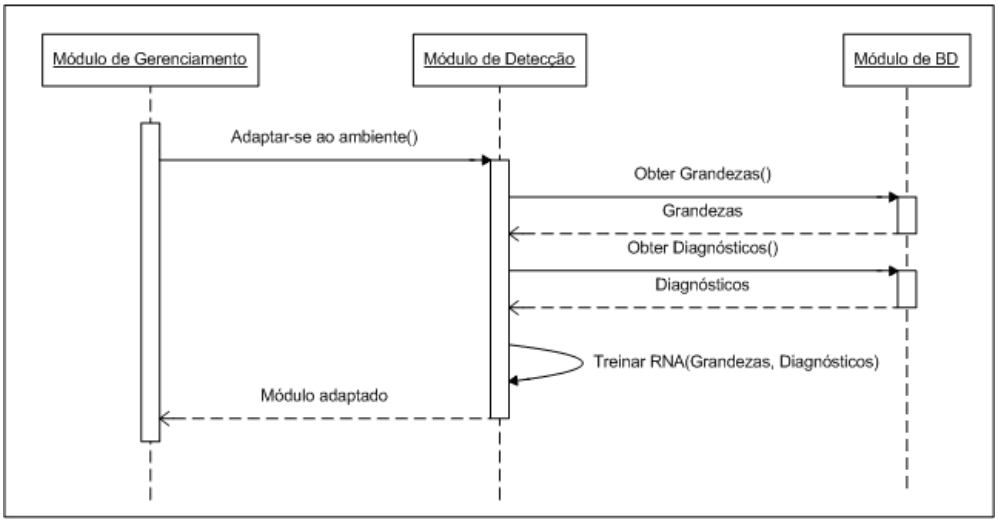

Figura 3.4: Treinamento da RNA do Módulo de Detecção.

Ao receber a requisição, o Módulo de Detecção obtém junto ao Módulo de Banco de Dados as informações necessárias para o treinamento da rede neural, que são os dados das tabelas Grandezas e Diagnósticos.

A tabela Grandezas possui em cada linha um conjunto de grandezas que quantificam aspectos do tráfego na rede wireless em um dado intervalo de tempo. Como exemplos de grandezas possíveis, temos o número de quadros de controle ou o número de requisições de associação em um intervalo de 2 segundos. A cada conjunto de grandezas da tabela Grandezas, corresponde uma entrada na tabela Diagnósticos, que indica o estado da segurança da rede *wireless*, na ocorrência desse conjunto de grandezas.

Depois de obtidos os dados das grandezas e seus respectivos diagnósticos, o Módulo de Detecção dá início ao treinamento da rede neural. Terminado o treinamento, a rede neural é capaz de determinar com boa precisão o diagnóstico mais

apropriado para cada conjunto de grandezas que for apresentado a ela, mesmo que esse conjunto de grandezas não tenha estado inserido no conjunto de dados de treinamento. Essa capacidade das redes neurais de apresentar uma resposta correta mesmo para situações para as quais não tenha sido treinada, é frequentemente denominada poder de generalização.

O treinamento da rede neural do Módulo de Contramedidas se dá de forma bastante similar. A Figura 3.5 mostra, através de um diagrama de interação, o contexto no qual é realizado esse treinamento.

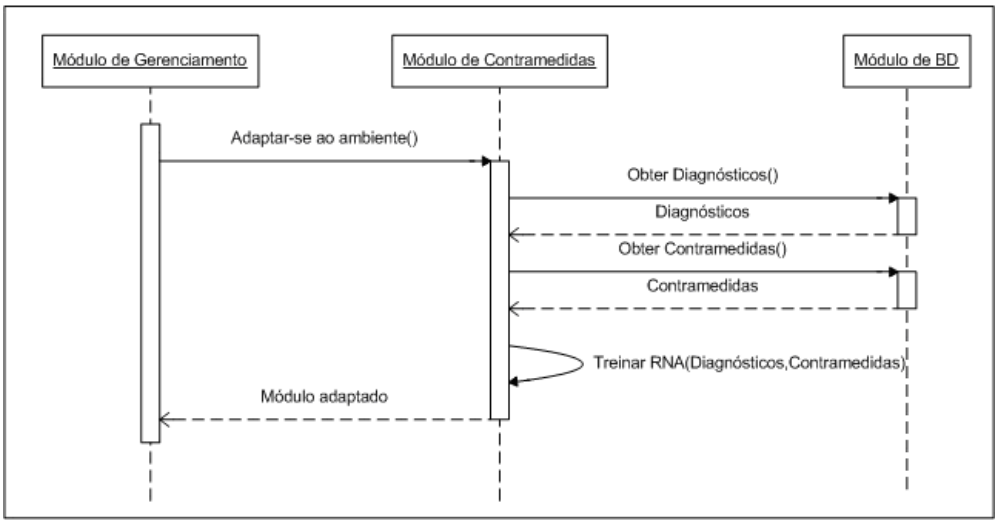

Figura 3.5: Treinamento da RNA do Módulo de Contramedidas.

Inicialmente, o Módulo de Gerenciamento envia para o Módulo de Contramedidas uma requisição para que o mesmo realize sua adaptação ao novo ambiente. Essa requisição pode ser invocada pelo administrador do sistema através da interface gráfica do Console de Gerenciamento.

Ao receber a requisição, o Módulo de Contramedidas obtém junto ao Módulo de Banco de Dados as informações necessárias para o treinamento da rede neural, que são os dados das tabelas Diagnósticos e Contramedidas.

A tabela Contramedidas possui em cada linha um conjunto de variáveis que representam uma contramedida que pode ser tomada por parte do WIDS. Cada contramedida nessa tabela pode estar associada a vários diagnósticos na tabela Diagnósticos.

Depois de obtidos os dados dos diagnósticos e suas respectivas contramedidas,

o Módulo de Contramedidas dá início ao treinamento da rede neural. Terminado o treinamento, a rede neural é capaz de determinar com boa precisão a contramedida mais apropriada para cada diagnóstico que for apresentado a ela, mesmo que esse diagnóstico não tenha estado inserido no conjunto de dados de treinamento.

3.3.2 Fase de Simulação

Na fase de simulação, o sistema realiza a captura do tráfego na rede monitorada, bem como a detecção de intrusões e a respectiva tomada de contramedidas, além de registrar todas as informações em banco de dados para posterior análise e relatórios. Todo esse processo se dá em quatro etapas: a operação do Módulo Sensor, a operação do Módulo de Detecção, a operação do Módulo de Contramedidas e a operação do Módulo Atuador.

A operação do Módulo Sensor pode ser visualizada no diagrama de interação da Figura 3.6. Inicialmente, o Módulo Sensor captura em modo promíscuo todo o tráfego da rede *wireless*, que é disponibilizado pela biblioteca de captura como uma sequência de códigos hexadecimais, sem nenhum significado aparente, sendo denominado de *raw data*, ou dados puros.

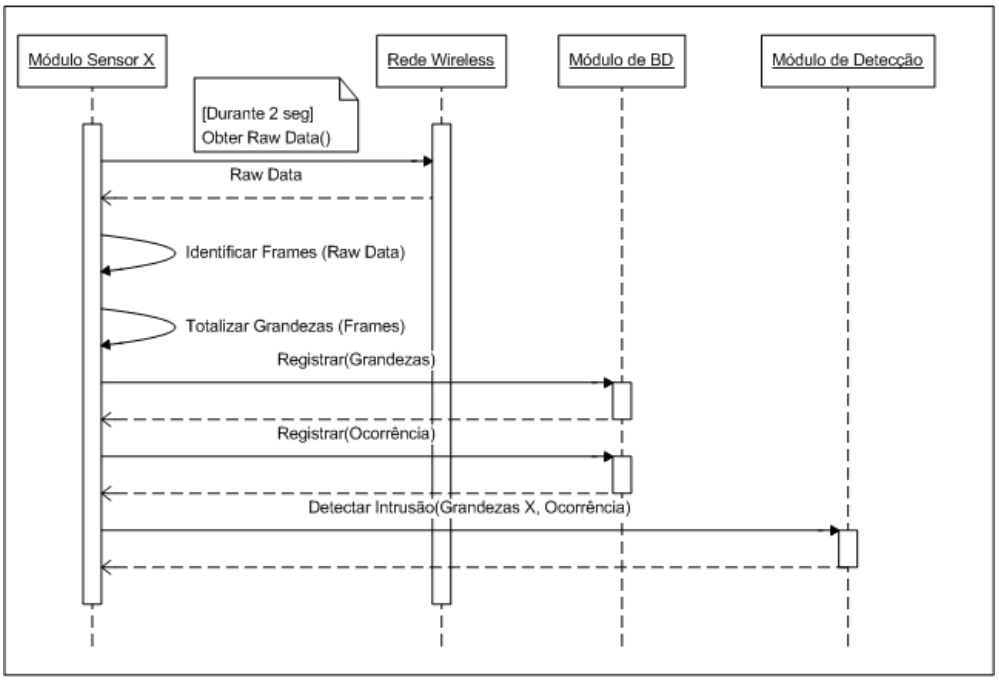

Figura 3.6: Operação do Módulo Sensor.

Uma vez capturada, essa seqüência de hexadecimais passa por um processo minucioso de análise, para que possam ser identificados os quadros, de acordo com a especificação da camada MAC (*Medium Access Control*) do Padrão IEEE 802.11b (ANSI/IEEE, 1999). O Formato desses quadros pode ser visualizado na Figura 3.7.

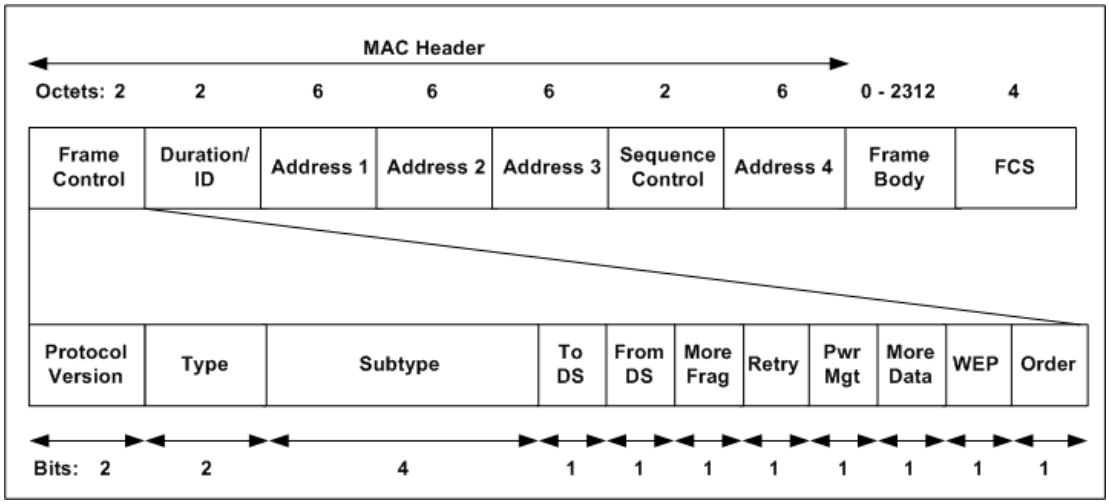

Figura 3.7: Formato dos Quadros IEEE 802.11b.

De acordo com o Padrão IEEE 802.11b, cada quadro possui um campo *Frame Control*, com 11 subcampos, um campo *Duration/ID*, que informa por quanto tempo o quadro e sua confirmação ocuparão o canal, quatro campos *Address*, que informam a origem e o destino de cada quadro, um campo *Sequence Control*, que permite que os fragmentos sejam numerados, um campo *Frame Body*, que contém a carga útil do pacote, e um campo *FCS*, que possui um total de verificação. O campo *Frame Control* possui um subcampo *Protocol Version*, que permite a operação de duas versões do protocolo ao mesmo tempo na mesma célula, um subcampo *Type* (dados, controle e gerenciamento), um subcampo *Subtype* (por exemplo RTS ou CTS), os bits *To DS* e *From DS*, que indicam se o quadro está indo ou vindo do sistema de distribuição entre células, um bit *More Frag*, que indica a existência de mais fragmentos, um bit *Retry*, que indica a retransmissão de um quadro enviado anteriormente, um bit *Pwr Mgt*, usado pelo AP para deixar o receptor em estado de espera ou retirá-lo do estado de espera, um bit *More Data*, que indica que o transmissor tem quadros adicionais para o receptor, um bit *WEP*, que indica que o corpo do quadro foi criptografado com o WEP, e um bit *Order*,

que informa ao receptor que uma sequência de quadros com esse bit tem que ser processada estritamente em ordem.

Depois de identificados, os quadros são agrupados de acordo com dois critérios: em intervalos de 2 segundos e por fonte emissora, de acordo com o esquema da Figura 3.8. Uma análise realizada para um intervalo de 2 segundos possibilita que qualquer alteração no comportamento da rede seja detectada quase instantaneamente (Bellardo and Savage, 2003). Esse intervalo deve ser flexível, permitindo a alteração por parte do administrador do sistema, para que possa ser testado o poder de reação do WIDS diante de várias condições de comprometimento da rede *wireless*. Já uma análise realizada por fonte emissora possibilita a identificação do mais provável responsável por qualquer comportamento anômalo. De posse dos frames agrupados, são totalizadas para cada um desses grupos as grandezas que servem de parâmetro para a determinação do diagnóstico da rede nesse intervalo de tempo. Essas grandezas são registradas no Módulo de Bancos de Dados, para posterior análise.

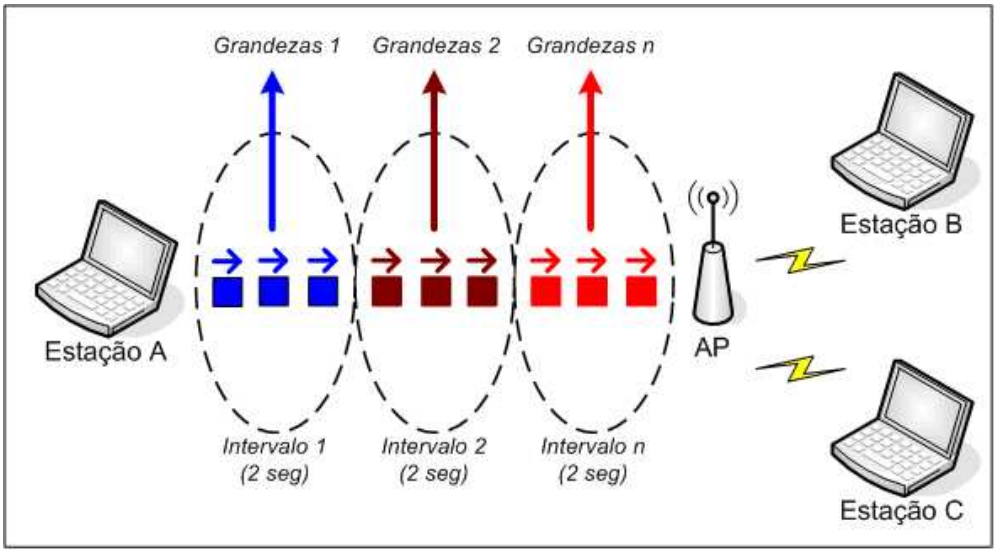

Figura 3.8: Esquema de Agrupamento dos Frames.

Outra informação registrada no Módulo de Bancos de dados é a referente às condições nas quais ocorreu a captura das grandezas, o que corresponde à tabela Ocorrência do banco de dados. Entre as informações registradas nessa tabela estão: o endereço MAC da estação origem, o endereço MAC do sensor que capturou o tráfego, o canal no qual se deu a captura, o instante inicial e o instante

final que delimitaram o intervalo da captura dos quadros.

De posse dos dados da ocorrência e suas respectivas grandezas associadas, o Módulo Sensor envia esses dados juntamente com uma requisição para o Módulo de Detecção, para que o mesmo proceda na detecção de uma possível intrusão.

Ao receber do Módulo Sensor os dados da ocorrência e suas respectivas grandezas associadas, o Módulo de Detecção agrupa essas grandezas com as recebidas de outros sensores, mas capturadas no mesmo intervalo de tempo. Isso permite que o WIDS realize uma detecção de intrusos de caráter muito mais global, utilizando dados capturados por vários dispositivos sensores espalhados por toda a rede. O diagrama de interação com a operação do Módulo de Detecção pode ser visualizado na Figura 3.9.

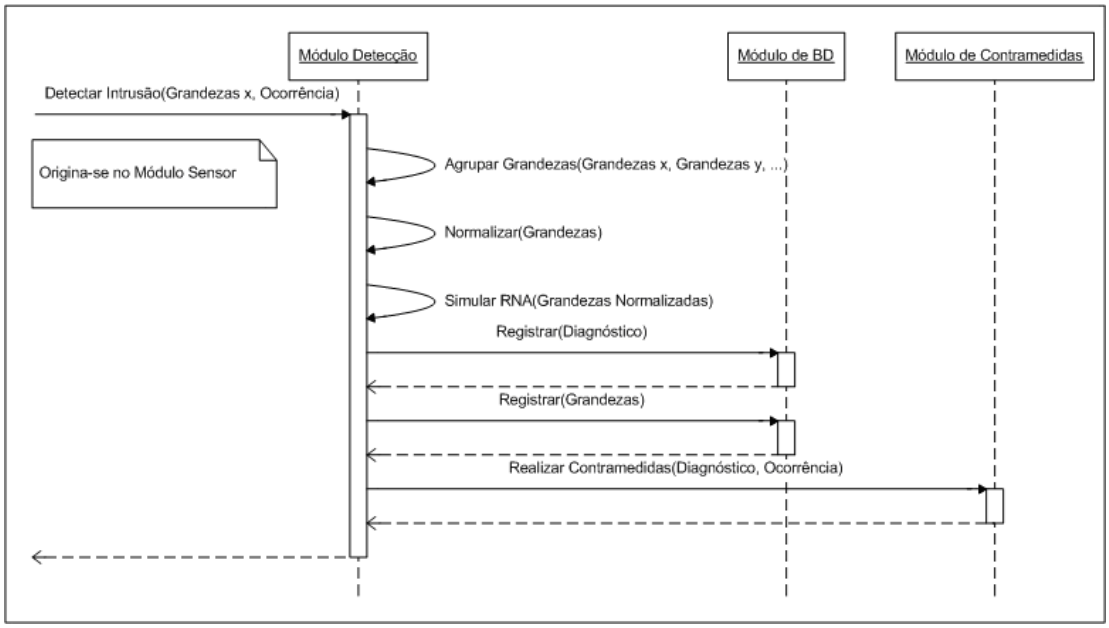

Figura 3.9: Operação do Módulo de Detecção.

As grandezas agrupadas passam por um processo de normalização, que tem como objetivo obter o valor equivalente de cada grandeza no intervalo entre 0 e 1 (intervalo [0, 1]), visto que somente valores dentro desse intervalo podem dar entrada na rede neural.

A simulação da rede neural é realizada aplicando-se em sua entrada essas grandezas normalizadas, obtendo-se como saída um conjunto de variáveis que representam o diagnóstico da rede *wireless*.

Uma vez determinado o diagnóstico, o mesmo é registrado para posterior análise na tabela Diagnósticos, além do conjunto de grandezas associadas, na tabela Grandezas.

O Módulo de Detecção então envia esse diagnóstico juntamente com uma requisição para o Módulo de Contramedidas, para que o mesmo proceda na determinação de uma contramedida apropriada para o diagnóstico em questão.

Ao receber do Módulo de Detecção o diagnóstico, o Módulo de Contramedidas simula a sua rede neural aplicando em sua entrada o diagnóstico já normalizado, obtendo como saída um conjunto de variáveis que representam a contramedida apropriada a ser realizada. O diagrama de interação com a operação do Módulo de Contramedidas pode ser visualizado na Figura 3.10.

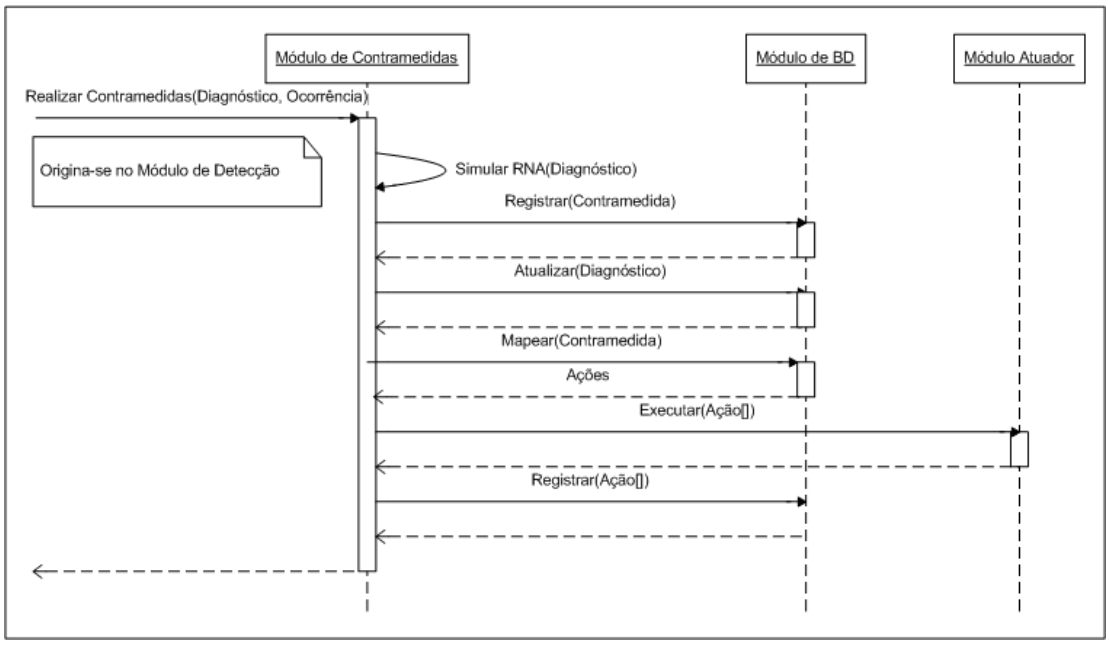

Figura 3.10: Operação do Módulo de Contramedidas.

Uma vez determinado a contramedida, a mesma é registrada para posterior análise na tabela Contramedidas, além do diagnóstico associado, que é atualizado na tabela Diagnósticos.

Nesse ponto, o Módulo de Contramedidas busca junto ao Módulo de Banco de Dados a lista de ações necessárias para implementar a contramedida indicada. Essas ações ficam armazenadas na tabela Ações do banco de dados. De posse dessa lista de ações, O Módulo de Contramedidas envia uma requisição para o

Módulo Atuador, para que o mesmo execute cada ação da lista, passando-a como parâmetro.

Essa lista de ações executadas é registrada junto ao Módulo de Banco de dados, juntamente com o endereço MAC do elemento atuador e o instante de sua execução. Esse registro é muito importante para todas as análises posteriores por parte do administrador do sistema.

Ao receber do Módulo de Contramedidas a lista de ações a serem executadas, o Módulo Atuador dá início à execução de tais ações. Estas ações tipicamente requerem que o Módulo Atuador injete quadros ativamente na rede *wireless*. Como exemplo de uma ação possível, temos o envio de quadros de desautenticação para uma determinada estação cliente, que esteja sob suspeita de ataque na rede *wireless*. O diagrama de interação com a operação do Módulo Atuador pode ser visualizado na Figura 3.11.

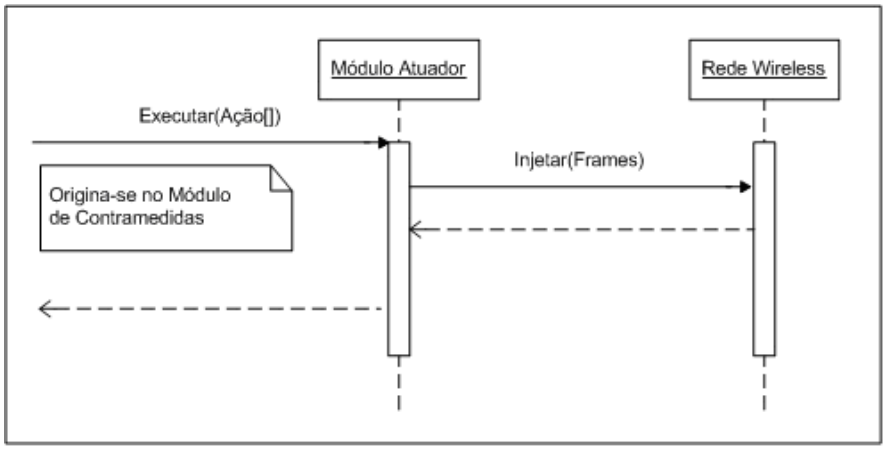

Figura 3.11: Operação do Módulo Atuador.

3.4 Aplicabilidade da Arquitetura ao NIDIA

3.4.1 O Projeto NIDIA

A proposta original do NIDIA (*Network Intrusion Detection System based on Intelligent Agents*) (Lima, 2001) é apresentar um sistema de detecção de intrusão em tempo real, composto por um conjunto de agentes, fornecendo um modelo de

detecção de intrusos baseado na noção de sociedade de agentes inteligentes, capaz de detectar novos ataques através de uma rede neural.

O NIDIA é inspirado no modelo lógico do CIDF (Staniford-Chen, 1998), possuindo para este fim, agentes com função de geradores de eventos (agentes sensores), mecanismos de análise dos dados (agentes de monitoramento e de avaliação de segurança), mecanismos de armazenamento de histórico (bases de dados) e um módulo para realização de contramedidas (agente controlador de ações). Além disso, existem agentes responsáveis pela integridade do sistema e pela coordenação das atividades do IDS como um todo.

Desta forma, os agentes do NIDIA possuem os seguintes objetivos gerais:

- Gerar índices de suspeita de ataque a partir da análise de dados coletados de logs de hosts e de pacotes de tráfego de rede;
- Tomar contramedidas de acordo com os índices obtidos;
- Aprender com os casos obtidos atualizando suas bases de conhecimento.

O modelo proposto prevê a metodologia de detecção por abuso e anomalia para garantir uma robustez maior ao sistema. Entretanto, atualmente, tem sido implementada somente a detecção por abuso como método de análise. A escolha se deu em virtude da grande maioria dos ataques poderem ser codificados, de maneira a capturar e registrar variantes das atividades que exploram as mesmas vulnerabilidades.

A escolha da arquitetura multiagentes (Hegazy, Alarif, Fayed and Fahim, 2003) para o IDS NIDIA visa obter as seguintes vantagens:

- Agentes podem ser adicionados ou removidos para/do sistema sem modificar outros componentes do sistema;
- Agentes podem ser reconfigurados ou atualizados sem causar problemas no restante do sistema;
- Um agente ou um grupo de agentes pode realizar diferentes funções simples. Visto que os agentes podem trocar informações entre si, podem derivar resultados mais complexos.

A Figura 3.12 mostra a arquitetura atual do NIDIA, que é composta por camadas (Oliveira, 2006). Cada camada possui atividades a desempenhar, sendo que estas atividades são executadas através dos comportamentos dos agentes que a compõem. É através destes agentes também, que as camadas se comunicam trocando informações importantes para desempenhar as suas atividades.

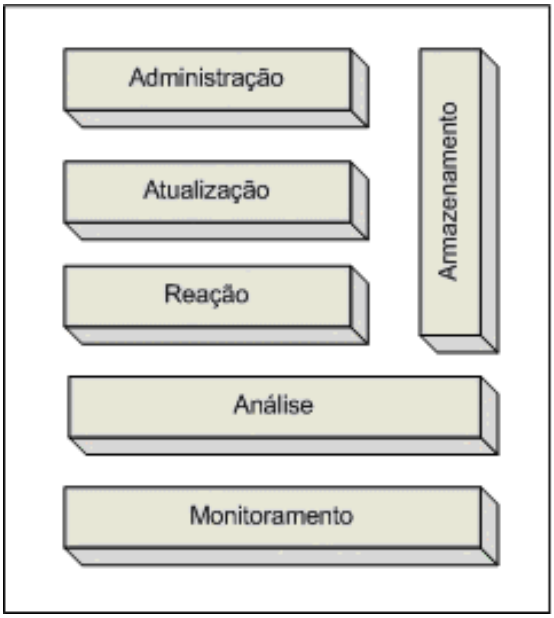

Figura 3.12: Modelo em Camadas do NIDIA.

Segue abaixo uma breve descrição das funcionalidades de cada camada do NIDIA, bem como os respectivos agentes que a compõem:

Camada de Monitoramento

Camada responsável por capturar a ocorrência de eventos no meio exterior e fornecer informações sobre o mesmo para o resto do sistema. Nesta camada, os agentes SMA (*System Monitoring Agent*) estão localizados. Estes agentes funcionam como os "sentidos receptores"do sistema. Os agentes SMAs dividem-se em duas categorias:

- Agentes Sensores de Rede - responsáveis por capturar os pacotes que estão trafegando na rede. Estes atuam em pontos estratégicos e funcionam como monitores de rede passivos, trabalhando em modo promíscuo, desta forma não interferindo na performance e nem no trêfego da rede;

- Agentes Sensores de *Host* - trabalham coletando informações em tempo real de um *host* em particular (geralmente servidores) e disponibilizando-as para análise.

Os dados obtidos recebem uma pré-formatação sendo em seguida repassados para o agente de avaliação de segurança.

Camada de Análise

Camada responsável pela análise dos eventos recebidos da camada de monitoramento. Nesta camada, os eventos coletados são formatados de maneira que padrões de ataques possam ser identificados e posteriormente a haja a confirmação de um verdadeiro ataque. Para isso utiliza bases de conhecimento, como a base de dados de padrões de intrusões (IIDB), a base de dados de incidentes de intrusão (DFDB) e a base de dados de estratégias (STDB). Nesta camada, localizam-se os agentes SEA (*Security Evaluation Agent*). Estes são responsáveis por realizar a análise dos eventos coletados e emitir um grau de suspeita sobre os eventos que foram previamente formatados.

Camada de Reação

Camada responsável por tomar contramedidas caso um incidente de segurança seja detectado. Com base no parecer do SEA, esta camada deve tomar uma contramedida de acordo com as bases de dados de estratégia (STBD) e de ações (RADB). Nesta camada, localizam-se os agentes SCA (*System Controller Agent*).

Camada de Atualização

Camada responsável pela atualização das bases de informações. As consultas poderão ser feitas diretamente de qualquer camada, porém inserções devem ser feitas somente através desta camada. Ela terá também a responsabilidade de manter a integridade e consistência das informações armazenadas. Nesta camada, localizam-se os agentes SUA (*System Updating Agent*). Estes são responsáveis pela atualização das bases DFDB, IIDB, RADB e STDB.

Camada de Administração

Camada responsável pela administração e integridade de todos os agentes do sistema. Nesta camada, localizam-se os agentes MCA (*Main Controller Agent*).

Camada de Armazenamento

Camada responsável por manter de forma persistente informações provenientes das demais camadas. Nesta camada, localizam-se as bases de dados utilizadas pelo NIDIA. Segue uma breve descrição das mesmas:

- STBD (*Strategy DataBase*) é uma base de dados responsável por registrar as estratégias adotadas por uma organização qualquer em relação à sua política de segurança. Ela é importante para garantir a adaptabilidade do IDS aos mais diversos casos;
- RABD (*Reaction DataBase*) estão contidas as informações referentes às ações que devem ser tomadas de acordo com a severidade do ataque detectado. Também varia de acordo com a política de cada instituição;
- IIDB (*Incidents of Intrusion and Forensic Information DataBase*) guarda as assinaturas de intrusão que serão utilizadas para a detecção de atividades suspeitas. Ele deve ser constantemente atualizado para garantir que novas técnicas de ataque possam ser detectadas;
- DFDB (*Standard of Intruders and Intrusions DataBase*) registra os danos causados por ataques bem-sucedidos e tentativas de ataques. Nele ficam contidas as informações que podem ser úteis na identificação de tentativas de ataques provenientes de uma mesma origem ou domínio ou simplesmente para serem utilizadas em investigações futuras.

3.4.2 A Integração da Arquitetura ao NIDIA

A integração do WIDS proposto à arquitetura do NIDIA é totalmente vantajosa e viável, visto que essa integração proporciona capacidades de detecção de intrusos e geração de contramedidas no ambiente *wireless* ao NIDIA, que é um sistema originalmente idealizado para a segurança de redes cabeadas e servidores.

Essa integração se torna facilitada, uma vez que ambas as arquiteturas são estruturadas em camadas, que possuem responsabilidades bastante similares entre os dois modelos. Cada módulo que constitui a arquitetura do WIDS proposto pode ser mapeado em um agente inteligente e ser inserido em sua respectiva camada na arquitetura do NIDIA.

Na Figura 3.13, os elementos de cor azul representam os agentes e bancos de dados da arquitetura NIDIA, enquanto os elementos de cor amarela representam os agentes e o banco de dados da arquitetura do WIDS proposto.

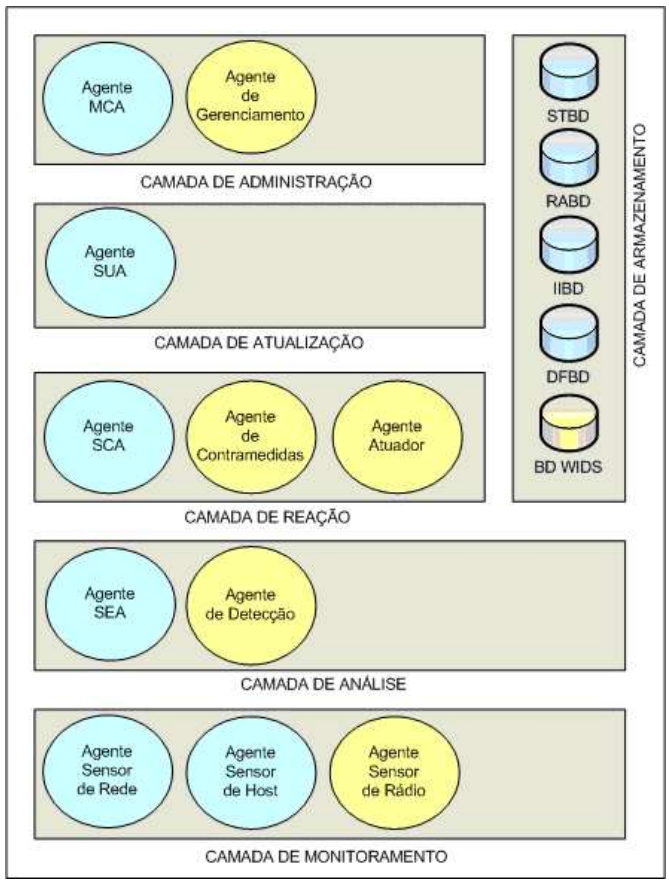

Figura 3.13: Integração do WIDS Proposto ao NIDIA.

O agente Sensor de Rádio é inserido no NIDIA na camada de Monitoramento, juntamente com o agente Sensor de Rede e o agente Sensor de Host. Isso amplia significativamente a capacidade de captura do NIDIA, com a inclusão da captura no ambiente de comunicação sem fio.

O agente de Detecção é inserido no NIDIA na camada de Análise, juntamente com o agente SEA (*System Evaluation Agent*). Isso amplia o poder de detecção do NIDIA, com a inclusão da capacidade de detecção de intrusões ocorrendo no meio *wireless*, através do emprego de redes neurais.

O agente de Contramedidas e o agente Atuador são inseridos no NIDIA na camada de Reação, juntamente com o agente SCA (*System Controller Agent*). Isso

amplia o poder de reação do NIDIA, com a inclusão da capacidade de decidir, com o emprego de redes neurais, as contramedidas mais apropriadas, além de executar ações efetivas contra intrusões no ambiente *wireless*.

Não foi inserido nenhum agente do WIDS proposto na camada de Atualização do NIDIA, visto que essa camada foi idealizada para servir como intermediária para qualquer inserção de informação no banco de dados. No entanto, os próprios agentes do WIDS proposto realizam o acesso diretamente ao banco de dados tanto para consulta, quanto para atualização ou inclusão de novas informações.

O agente de Gerenciamento é inserido no NIDIA na camada de Administração, juntamente com o agente MCA (*Main Controller Agent*). Isso agrega ao NIDIA novas capacidades de gerenciamento por parte do administrador do sistema, com uma interface gráfica rica em opções de configuração, monitoramento, análise e geração de relatórios.

Por fim, o banco de dados WIDS é inserido no NIDIA na camada de Armazenamento, juntamente com os bancos de dados STBD (*Strategy Database*), RABD (*Reaction Database*), IIDB (*Incidents of Intrusion and Forensic Information Database*) e DFDB (*Standard of Intruders and Intrusions Database*). Isso traz para o NIDIA um banco de dados completo com informações detalhadas sobre toda a atividade das comunicações na rede sem fio.

3.5 Conclusão

Este capítulo apresentou a proposta de uma arquitetura para um sistema de detecção de intrusos em redes *wireless*, que tem como base a aplicação da tecnologia de redes neurais artificiais, tanto nos processos de detecção de intrusões quanto de tomada de contramedidas. O próximo capítulo apresentará a implementação de um protótipo para a arquitetura proposta.

Capítulo 4

Prototipagem da Solução

4.1 Introdução

Este capítulo apresenta a implementação de um protótipo para a arquitetura proposta. Foram implementados tanto o dispositivo sensor quanto o mecanismo de detecção de intrusões usando redes neurais. Para que esse mecanismo de detecção com redes neurais pudesse funcionar, foi gerado um arquivo de treinamento com registros de conexões normais e conexões sob ataques de DoS, onde cada conexão foi associada a um diagnóstico de segurança da rede *wireless*. Por fim, foram realizadas simulações e testes com o protótipo, com o objetivo de demonstrar a viabilidade e a eficácia da arquitetura proposta.

A implementação da solução foi dividida nas seguintes etapas:

- Configuração do ambiente de captura;
- Implementação do sensor de rádio;
- Geração do arquivo de registros normais;
- Geração do arquivo de registros de ataques;
- Configuração do ambiente de Simulação;
- Implementação do programa Simulador.

A parte de simulações e testes será apresentada no capítulo 5.

4.2 Configuração do Ambiente de Captura

A placa de rede sem fio utilizada foi um Adaptador DWL-G520 PCI Wireless 2.4 GHz AirPlus Xtreme G, da D-Link (D-Link, 2007). Essa placa permite a conexão a um Ponto de Acesso 802.11b ou 802.11g (para o modo infra-estruturado) ou outra placa *wireless* 802.11b ou 802.11g (para o modo *ad-hoc*, em redes ponto a ponto), podendo operar até 108Mbps de velocidade.

O sistema operacional utilizado foi o Fedora (antigamente chamado Fedora Core) (Fedora, 2007), que é uma distribuição Linux baseada em pacotes RPM, criada pela Red Hat (RedHat, 2007).

Para que o Linux pudesse usar a placa de rede sem fio, foi instalado o MadWifi (MADWIFI, 2007), que é um *driver open source* bastante avançado e estável disponível para placas de rede *wireless* baseadas em *chipsets* Atheros (Atheros, 2007), que é o caso do Adaptador DWL-G520. Este driver suporta os seguintes modos de operação: estação, ponto de acesso, *ad-hoc* e monitor. O modo monitor, também conhecido como modo promíscuo, é o modo utilizado pelo elemento sensor, pois permite que a interface de rede capture todos dos pacotes que trafegam pela rede, inclusive os pacotes não destinados a ela.

Para a implementação da captura de pacotes, a API (*Application Program Interface*) Jpcap (Jpcap, 2007), que roda sobre a API Libpcap (Libpcap, 2007), foi utilizada. Libpcap é uma interface independente de sistema para captura de pacotes em nível de usuário. Ela provê um *framework* portável para o monitoramento de rede de baixo nível, em sistemas Linux, BSD e derivados do Unix. O Jpcap é uma API para o desenvolvimento de aplicações de captura de pacotes em Java (Java, 2007). Além do conjunto de classes Java, ela inclui uma ferramenta para a visualização e análise do tráfego da rede *wireless* em tempo real.

Para a implementação do elemento sensor, Java foi utilizado. Diferentemente das linguagens convencionais, que são compiladas para código nativo, a linguagem Java é compilada para um "bytecode", que é executado por uma máquina virtual, garantindo assim total portabilidade à linguagem.

Como ambiente de desenvolvimento, Eclipse (Eclipse, 2007) foi utilizado, que é um IDE (*Integrated Development Environment*) para o desenvolvimento de aplicativos que possui uma forte orientação ao desenvolvimento baseado em *plug-ins* e é uma das IDEs mais utilizadas no mundo para o desenvolvimento em

Java.

4.3 Implementação do Sensor de Rádio

O diagrama de classes (UML, 2007) do Sensor de Rádio pode ser visualizado através da Figura 4.1. Esse diagrama mostra as classes do sistema, com os seus respectivos relacionamentos.

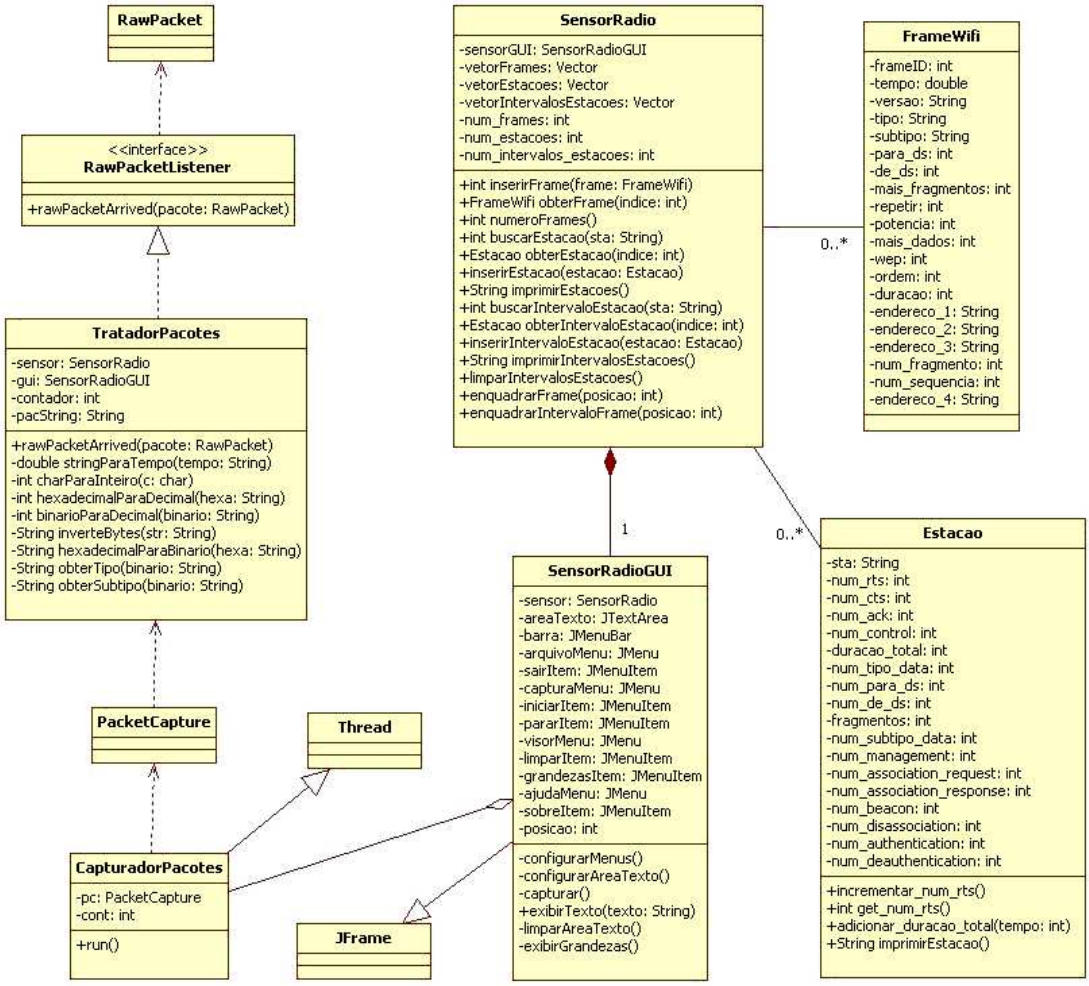

Figura 4.1: Diagrama de Classes do Sensor.

A classe FrameWifi define os objetos que representam os quadros capturados na rede *wireless*. Exemplos de atributos dessa classe são a versão, o tipo e o subtipo do frame, definidos no padrão IEEE 802.11b.

A classe Estação define os objetos que agrupam as grandezas medidas para cada estação em um dado intervalo de tempo. Exemplos de grandezas são: o número de frames RTS (*Request to Send*) enviados pela estação ou a duração total das comunicações na rede envolvendo a estação.

Tanto a classe FrameWifi quanto a classe Estação são associadas à classe SensorRadio, que é onde está localizada a função principal do programa sensor. A classe SensorRadio possui três vetores, nos quais armazena os dados capturados da rede, bem como o pré-processamento que é realizado sobre esses dados. O vetor vetorFrames armazena em cada posição um objeto da classe FrameWifi. O vetor vetorEstacoes armazena em cada posição um objeto da classe Estação. O vetor vetorIntervalosEstacoes armazena em cada posição um objeto Estação com suas grandezas referentes apenas a um dado intervalo de tempo.

Os métodos da classe SensorRadio compreendem métodos para inserir, localizar e obter referências para os objetos FrameWifi e Estacao nos três vetores citados anteriormente. Existem também os métodos que enquadram os quadros recebidos em sua respectiva estação transmissora, nos vetores vetorEstacoes e vetorIntervalosEstacoes.

A classe SensorRadio possui como um de seus componentes a classe SensorRadioGUI, que herda a classe JFrame, do pacote *javax.swing*, e define os atributos e métodos do objeto interface gráfica do sensor. Seus atributos compreendem os itens do menu principal e a área de texto na qual são exibidas as informações do tráfego capturado. Seus métodos compreendem métodos de configuração da interface gráfica, exibição de um texto na tela, limpeza da tela, iniciação do mecanismo de captura de pacotes e exibição das grandezas totalizadas para cada estação em uma captura realizada.

A tela principal da interface gráfica do sensor pode ser visualizada através da Figura 4.2.

Ao ser acionado o mecanismo de captura de pacotes através da interface gráfica, é criado e colocado em execução um objeto da classe CapturadorPacotes. Como essa classe herda da classe Thread, do pacote *java.lang*, seus objetos executam de forma concorrente no processador, deixando assim a interface gráfica livre para receber comandos do usuário, como um pedido de interrupção da captura.

Internamente, a classe CapturadorPacotes faz uso da classe PacketCapture, do

Figura 4.2: Tela Inicial do Sensor.

pacote *net.sourceforge.jpcap.capture*, que é a classe núcleo da captura de pacotes da biblioteca Jpcap. Ela provê uma interface de alto nível para a captura de pacotes de rede através do encapsulamento da biblioteca Libpcap.

Para que a captura de pacotes possa ocorrer, um objeto da classe PacketCapture deve registrar um objeto da classe TratadorPacotes, que é a classe que implementa a interface RawPacketListener. A classe TratadorPacotes implementa o método rawPacketArrived(), através do qual recebe um objeto da classe RawPacket, do pacote *net.sourceforge.jpcap.net*, que é um pacote capturado contendo apenas dados brutos. A Figura 4.3 mostra a captura de pacotes na interface gráfica do sensor de rádio. Cada uma das entradas nesse visor corresponde a um pacote capturado pelo sensor. Em cada entrada, a linha superior indica que foi capturado um pacote da classe *net.sourceforge.jpcap.net.RawPacket* e o número de bytes perdidos durante a captura. No exemplo da figura, nenhum dos pacotes capturados perderam bytes durante a captura. A linha inferior indica o número de bytes efetivamente capturados, o número de bytes do pacote que foi capturado, o instante exato da captura e o conteúdo de cada pacote na forma de uma sequência de bytes.

O método rawPacketArrived() também é responsável, nessa implementação, por analisar os dados brutos recebidos, no intuito de extrair os campos de cada quadro e montar o objeto FrameWifi correspondente, que será armazenado no

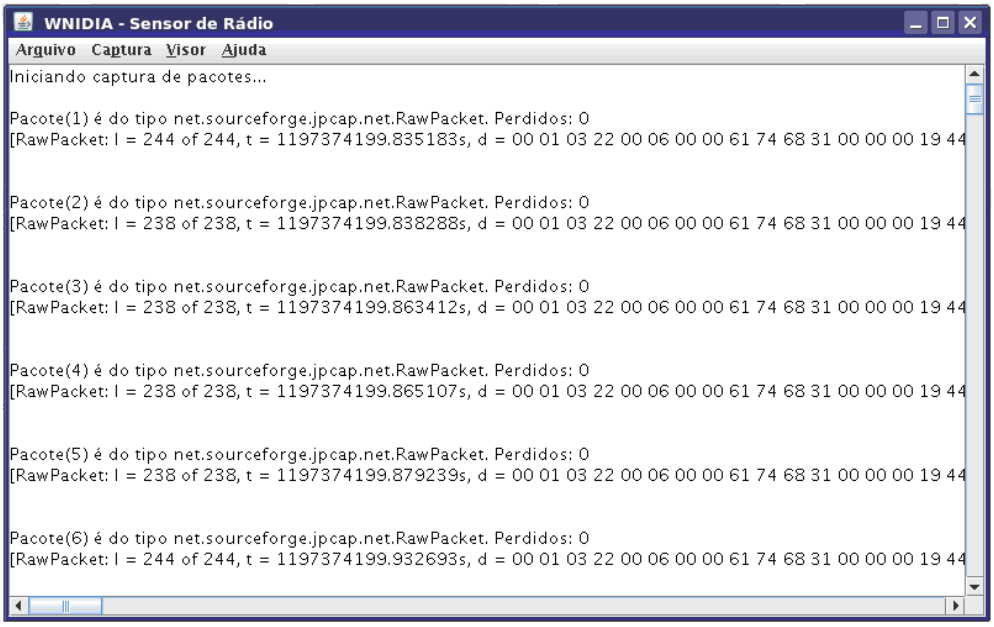

Figura 4.3: Captura de Frames Brutos.

vetor vetorFrames.

A Figura 4.4 mostra a captura de pacotes na interface gráfica do sensor de rádio, sendo que cada pacote está passando por um processo de análise e extração de campos, antes de ser exibido na tela do sensor.

A Figura 4.5 mostra a interface gráfica do sensor de rádio com as grandezas totalizadas para cada estação em uma captura realizada. Nesse exemplo, o sensor indica que a duração acumulada de todos os quadros foi de 117 microsegundos. Além disso, ele indica que foram capturados 17 quadros de dados, todos destinados ao sistema de distribuição. As demais variáveis estão com valor zero indicando que, durante o intervalo, não foram capturados quaisquer outros tipos de quadros.

4.4 Geração do Arquivo de Registros Normais

O arquivo de registros normais foi gerado através da captura do tráfego de uma rede *wireless* real, na qual foi possível garantir que, durante o período da captura, todo o tráfego estava dentro dos padrões habituais de uso da rede. Essa garantia se deu através do monitoramento contínuo de todas as estações acessando o meio *wireless*, com o uso de um computador executando o Kismet (Kismet, 2007).

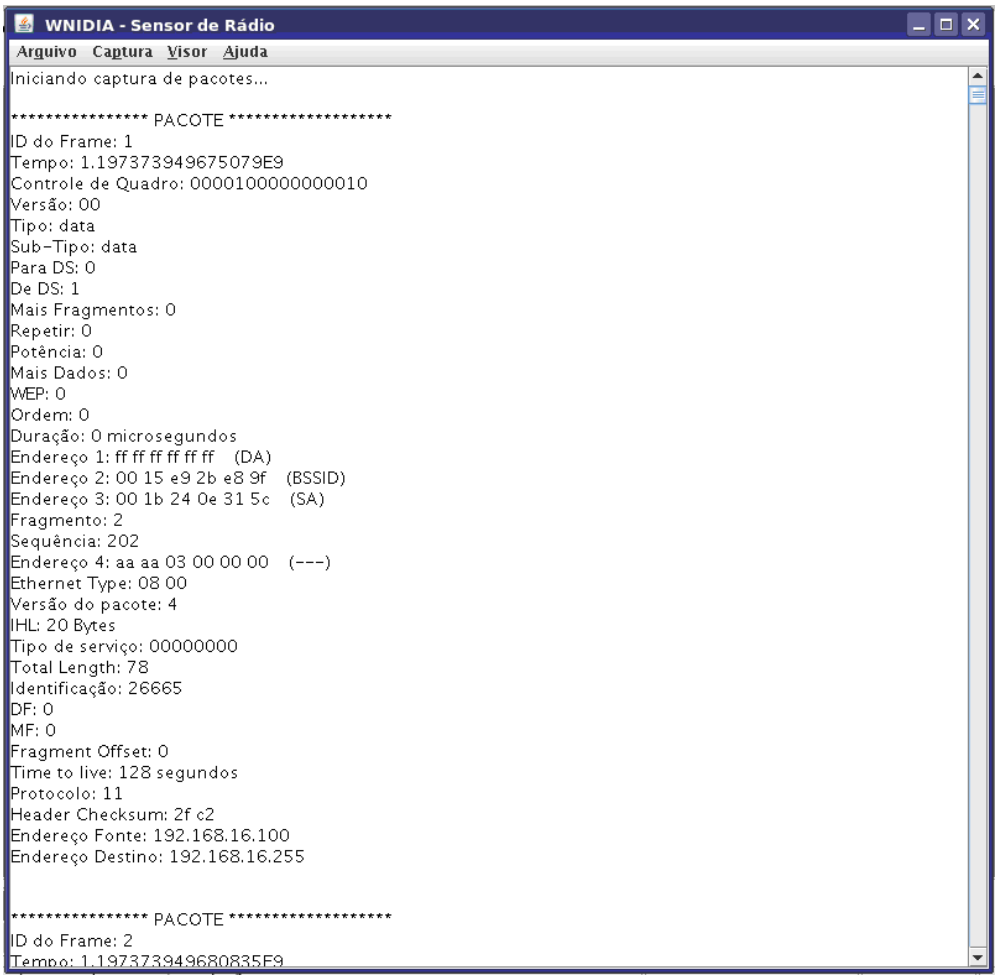

Figura 4.4: Captura de Frames com Análise de Campos.

A rede *wireless* utilizada como ambiente de captura foi a rede do LSAC (*Laboratório de Sistemas e Arquiteturas Computacionais*) do DEE (*Departamento de Engenharia de Eletricidade*) (DEE, 2007) da UFMA (*Universidade Federal do Maranhão*) (UFMA, 2007). O esquema geral do ambiente pode ser visualizado através da Figura 4.6.

O ambiente é composto por 3 PCs, 3 *notebooks*, 1 *palmtop* e 1 AP. Todos eles possuem interfaces de rede *wireless*, e são conectados à Internet através do ponto de acesso, que está conectado à rede cabeada da universidade. Suas configurações são detalhadas no Apêndice B desta dissertação.

A captura do tráfego durou aproximadamente 1 hora, no horário compreendido entre 19:00 hs e 20:00 hs, resultando na gravação de 21 mil registros de grandezas

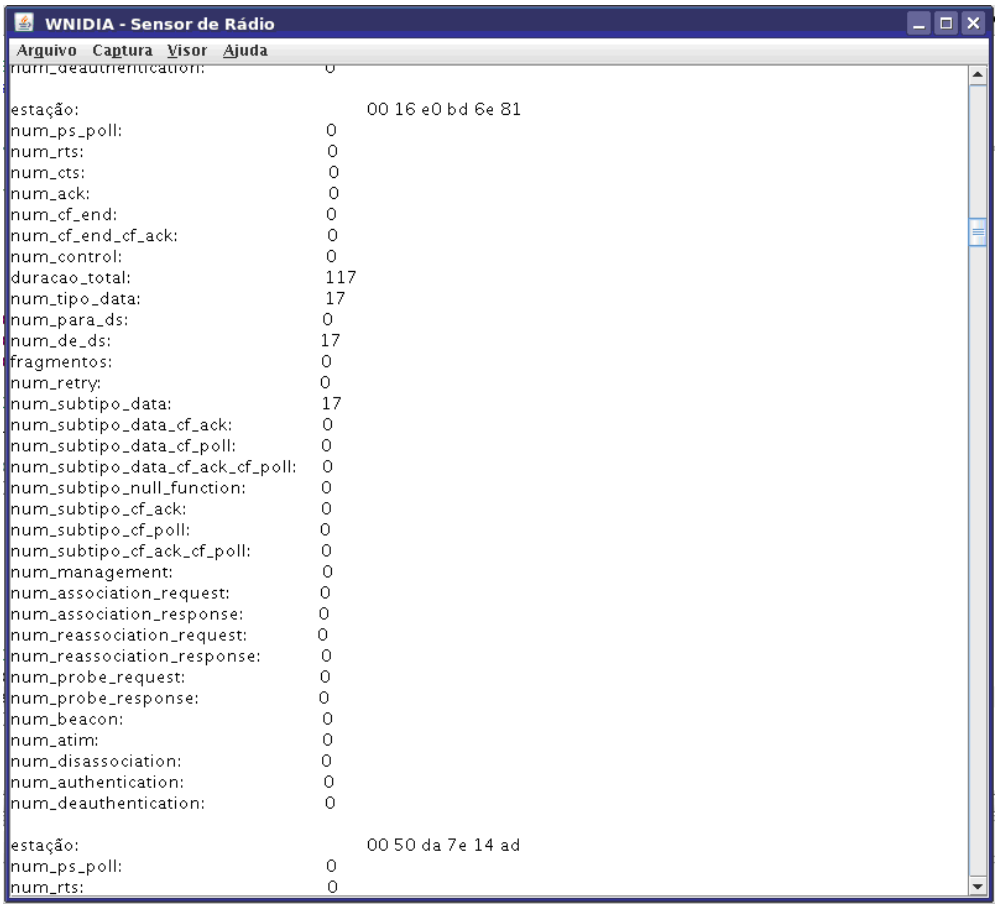

Figura 4.5: Grandezas de Cada Estação.

da rede. Durante esse período de captura, as estações clientes realizaram diversas atividades consideradas corriqueiras, como o acesso a páginas Web, *download* de arquivos e apresentação de vídeos pela Internet.

Cada registro armazenado no arquivo resultante é composto de um conjunto de grandezas, referentes às características do tráfego na rede *wireless* de uma estação transmissora em um intervalo de tempo de 2 segundos, de acordo com o esquema da Figura 3.7. As grandezas registradas, juntamente com uma breve descrição sobre cada uma delas, são listadas na Tabela 4.1.

As Figuras 4.7, 4.8 e 4.9 mostram um trecho do arquivo de registros normais gerado, onde cada linha representa um registro e cada coluna representa uma grandeza medida para esse registro.

Tabela 4.1: Grandezas em Cada Registro do tráfego.

GRANDEZA	DESCRIÇÃO
MAC_ESTACAO	Endereço da estação transmissora
PSPOLL	Nº de frames do subtipo Power Save (PS)-Poll
RTS	Nº de frames do subtipo Request To Send
CTS	Nº de frames do subtipo Clear To Send
ACK	Nº de frames do subtipo Acknowledgment
CF-END	Nº de frames do subtipo Contention-Free (CF)-End
CF-E-A	Nº de frames do subtipo CF-End + CF-Ack
CTRL	Nº de frames do tipo Control
DURAC	Duração acumulada de todos os frames
TPDATA	Nº de frames do tipo Data
PARADS	Nº de frames com o campo To DS ativado
DE-DS	Nº de frames com o campo From DS ativado
FRAGMT	Nº de frames com o campo More Fragments ativado
NRETRY	Nº de frames com o campo Retry ativado
SBDATA	Nº de frames do subtipo Data
DT-C-A	Nº de frames do subtipo Data + CF-Ack
DT-C-P	Nº de frames do subtipo Data + CF-Poll
DC-A-P	Nº de frames do subtipo Data + CF-Ack + CF-Poll
NULLF	Nº de frames do subtipo Null Function (no data)
CF-ACK	Nº de frames do subtipo CF-Ack (no data)
CFPOLL	Nº de frames do subtipo CF-Poll (no data)
CF-A-P	Nº de frames do subtipo CF-Ack + CF-Poll (no data)
MANAGE	Nº de frames do tipo Management
ASSREQ	Nº de frames do subtipo Association Request
ASSRES	Nº de frames do subtipo Association Response
REAREQ	Nº de frames do subtipo Reassociation Request
REARES	Nº de frames do subtipo Reassociation Response
PROREQ	Nº de frames do subtipo Probe Request
PRORES	Nº de frames do subtipo Probe Response
BEACON	Nº de frames do subtipo Beacon
ATIM	Nº de frames do subtipo ATIM
DISASS	Nº de frames do subtipo Disassociation
AUTHEN	Nº de frames do subtipo Authentication
DEAUTH	Nº de frames do subtipo Deauthentication
STATUS	Status do registro (normal ou ataque)

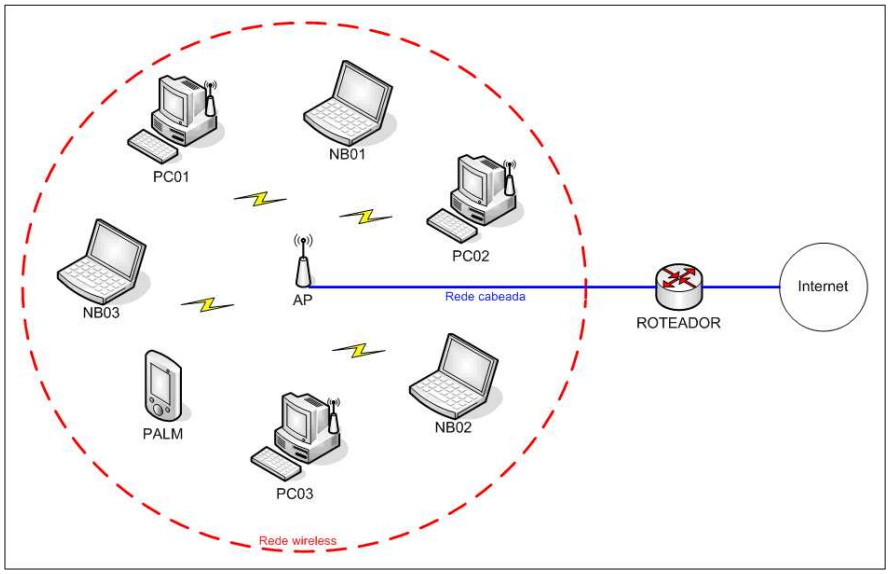

Figura 4.6: Esquema Geral do Ambiente de Captura.

Tráfego NORMAL - WordPad

Arquivo Editar Exibir Inserir Formatar Ajuda

MAC_ESTACAO	PSPOLL	RTS	CTS	ACK	CF-END	CF-E-A	CTRL	DURAC	TPDATA	PARADS	DE-DS
00 13 46 96 d5 0d	0	0	576	3340	0	0	3916	174144	129	129	0
00 15 e9 2b e8 9f	0	0	4268	49	0	0	4317	706030	0	0	0
00 13 20 98 68 b9	0	0	0	0	0	0	0	28082	446	0	446
00 12 a9 ac a0 81	0	0	0	0	0	0	0	0	7	0	7
00 16 e0 bd 6e 81	0	0	0	0	0	0	0	0	138	0	138
00 15 e9 dd 4a 26	0	0	0	0	0	0	0	0	6	0	6
00 13 46 96 ca 7f	0	0	0	65	0	0	65	7744	176	176	0
00 16 6f 7a ee 0f	0	0	67	5	0	0	72	5830	1	1	0

Para obter ajuda, pressione F1 NUM

Figura 4.7: Trecho do Arquivo de Dados Normais - Parte I.

Tráfego NORMAL - WordPad

Arquivo Editar Exibir Inserir Formatar Ajuda

FRAGMT	NRETRY	SBDATA	DT-C-A	DT-C-P	DC-A-P	NULLF	CF-ACK	CFPOLL	CF-A-P	MANAGE	ASSREQ	ASSRES	REAREQ
0	2	122	0	0	0	7	0	0	0	1	0	0	
0	4	0	0	0	0	0	0	0	0	403	0	0	
0	56	446	0	0	0	0	0	0	0	0	0	0	
0	0	7	0	0	0	0	0	0	0	0	0	0	
0	0	138	0	0	0	0	0	0	0	0	0	0	
0	0	6	0	0	0	0	0	0	0	0	0	0	
0	0	176	0	0	0	0	0	0	0	0	0	0	
0	0	1	0	0	0	0	0	0	0	0	0	0	

Para obter ajuda, pressione F1 NUM

Figura 4.8: Trecho do Arquivo de Dados Normais - Parte II.

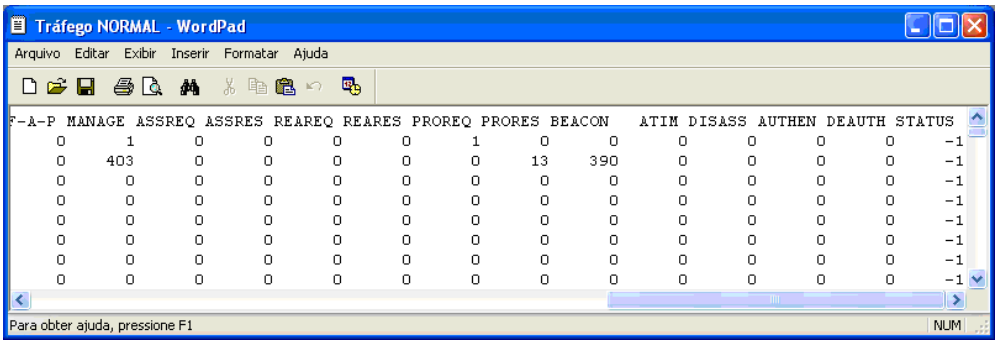

Figura 4.9: Trecho do Arquivo de Dados Normais - Parte III.

4.5 Geração do Arquivo de Registros de Ataques

O arquivo de registros de ataques foi criado baseando-se tanto nas características que definem cada ataque, quanto nas características observadas no tráfego capturado na rede *wireless*.

Na implementação deste trabalho três ataques de DoS, bastante conhecidos em redes *wireless*, foram considerados:

- Ataque *Virtual Carrier Sense*;
- Ataque *Association Flood*;
- Ataque *De-authentication*.

4.5.1 Ataque *Virtual Carrier Sense*

O mecanismo de *virtual carrier sense* é usado para atenuar colisões com terminais escondidos. Ele pode ser visualizado na Figura 4.10. Nesse mecanismo, cada quadro 802.11 carrega um campo de duração que indica o número de microsegundos que o canal está reservado. Esse valor é usado para programar o NAV (*Network Allocation Vector*) em cada estação. Somente quando o NAV da estação alcança zero, é que ela tem permissão para transmitir novos dados. Esse recurso é usado principalmente pelo *handshake* RTS (*Request To Send*) / CTS (*Clear To Send*), que serve para sincronizar o acesso ao canal quando um terminal escondido estiver interferindo com a transmissão. Durante esse *handshake*, a estação transmissora

envia um quadro RTS, que inclui uma duração grande o suficiente para completar toda a seqüência RTS/CTS (incluindo o quadro CTS, o quadro de dados e o quadro subseqüente de *acknowledgment*). A estação de destino responde ao RTS com um CTS, contendo um campo de duração atualizado em relação ao tempo já decorrido durante a seqüência. Após o CTS ser enviado, cada nó na extensão de rádio, tanto da estação transmissora quanto da estação receptora, vai ter atualizado seu NAV e vai reter todas as suas transmissões pela duração da transmissão que está ocorrendo. Embora o recurso RTS/CTS seja raramente usado na prática, respeitar a função de *virtual carrier sense* indicada pelo campo de duração é obrigatório em todas as implementações do padrão 802.11.

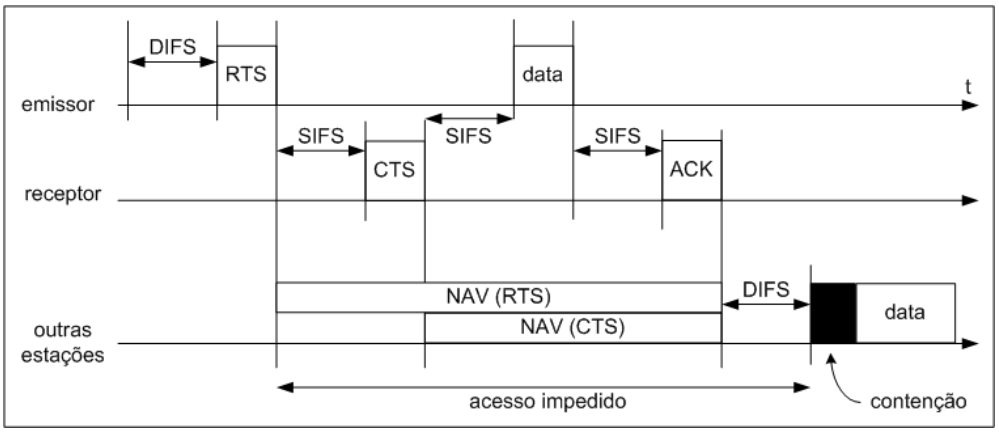

Figura 4.10: Mecanismo *Virtual Carrier Sense*.

Um atacante pode explorar o mecanismo de *virtual carrier sense* informando um campo de duração muito grande, prevenindo desse modo as demais estações de obter acesso ao canal (Bellardo and Savage, 2003). Isso pode ser visualizado através da Figura 4.11. O valor máximo para o NAV é 32767, ou aproximadamente 32 milisegundos em redes 802.11b. Assim, em princípio o atacante somente precisa transmitir aproximadamente 30 vezes por segundo, para obstruir todo o acesso ao canal.

Mesmo sendo possível usar quase qualquer tipo de frame para controlar o NAV, incluindo o próprio ACK, o uso do RTS apresenta algumas vantagens. Visto que uma estação com um comportamento correto vai sempre responder a um RTS com um CTS, o atacante pode fazer com as demais estações propaguem o ataque além do que ele poderia propagar sozinho. Disparando esse ataque repetidas

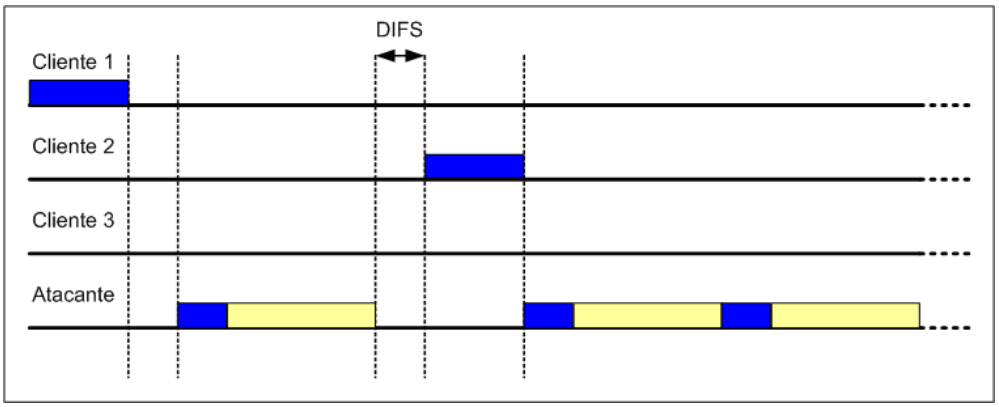

Figura 4.11: Ataque *Virtual Carrier Sense*.

vezes, o atacante pode causar uma degradação perceptível na performance da rede, resultando em um ataque de negação de serviço.

Os registros do ataque *virtual carrier sense* foram montados de acordo com a Tabela 4.2. Nesses registros, os campos determinantes para a ocorrência do ataque são: RTS (*Request To Send*), CTS (*Clear To Send*), CTRL (*número de frames do tipo Control*), DURAC (*duração acumulada de todos os frames*) e STATUS (*status do registro*).

Tabela 4.2: Formação dos Registros do Ataque *Virtual Carrier Sense*.

RTS	CTS	CTRL	DURAC	STATUS
De 50 a 70	Igual a RTS	RTS + CTS	CTRL * 32767	1

Se para obstruir o acesso ao canal, o atacante precisa transmitir aproximadamente 30 vezes por segundo, em 2 segundos são necessárias aproximadamente 60 transmissões. Acrescentando a esse valor uma tolerância de 10 quadros para mais e para menos, estabeleceu-se que o campo RTS deve variar entre 50 e 70. Para cada quadro RTS transmitido, um quadro CTS deve ser recebido. O número de quadros CTRL deve ser igual à soma entre o número de quadros RTS e o número de quadros CTS. O campo DURAC deve ser igual ao número de quadros CTRL multiplicado por 32767, que é a duração máxima permitida em cada quadro. O campo STATUS deve ser igual a 1 para indicar a ocorrência de um ataque.

4.5.2 Ataque *Association Flood*

Quando uma estação se associa a um ponto de acesso, este libera um AID (*Associate Identification*) para a estação na faixa entre 1 e 2007. Esse valor é usado para comunicar informações de gerenciamento de energia para a estação, quando a mesma estiver em estado *"power save"*. O ataque *association flood* é executado através do envio de múltiplas requisições de autenticação e associação para o ponto de acesso, todas com apenas um endereço MAC fonte (Curran and Smyth, 2006). O ponto de acesso é incapaz de diferenciar as requisições de autenticação geradas por um atacante, das criadas por clientes legítimos da rede, desse modo é forçado a processar cada requisição. Eventualmente, o ponto de acesso vai esgotar seus AIDs para alocação e vai ser forçado a desassociar estações, para poder usar AIDs já alocados. Na prática, vários pontos de acesso vão reiniciar após alguns minutos de inundação. Entretanto, esse ataque é bastante eficaz para derrubar segmentos de rede ou até mesmo redes inteiras.

Os registros do ataque *association flood* foram divididos em duas categorias: registros com grande número de quadros de requisição de autenticação e registros com grande número de quadros de requisição de associação. Em todos os registros, os campos determinantes para a ocorrência do ataque são: AUTHEN (*Authentication*), PRORES (*Probe Response*), ASSREQ (*Association Request*), MANAGE (*número de quadros do tipo Management*) e STATUS (*status do registro*).

Os registros com grande número de quadros de requisição de autenticação foram montados de acordo com a Tabela 4.3.

Tabela 4.3: Formação dos Registros do Ataque *Association Flood* - Tipo A.

AUTHEN	PRORES	ASSREQ	MANAGE	STATUS
De 50 a 70	De 29 a 31	De 29 a 31	AUTHEN + PRORES + ASSREQ	1

O campo AUTHEN deve variar entre 50 e 70. Os campos PRORES e ASSREQ devem variar entre 29 e 31, pois se verificou dos dados capturados na rede que, sempre que há um grande número de quadros de requisição de autenticação na rede, há também um número considerável de quadros de resposta de *probing* e de requisição de associação. O número de quadros MANAGE deve ser igual à soma entre o número de quadros AUTHEN, o número de quadros PRORES e o

número de quadros ASSREQ. O campo STATUS deve ser igual a 1, para indicar a ocorrência do ataque.

Os registros com grande número de quadros de requisição de associação foram montados de acordo com a Tabela 4.4.

Tabela 4.4: Formação dos Registros do Ataque *Association Flood* - Tipo B.

ASSREQ	PRORES	AUTHEN	MANAGE	STATUS
De 50 a 70	De 29 a 31	De 29 a 31	ASSREQ + PRORES + AUTHEN	1

O campo ASSREQ deve variar entre 50 e 70. Os campos PRORES e AUTHEN devem variar entre 29 e 31, pois se verificou dos dados capturados na rede que, sempre que há um grande número de quadros de requisição de associação na rede, há também um número considerável de quadros de resposta de *probing* e de requisição de autenticação. O número de quadros MANAGE deve ser igual à soma entre o número de quadros ASSREQ, o número de quadros PRORES e o número de quadros AUTHEN. O campo STATUS deve ser igual a 1, para indicar a ocorrência do ataque.

4.5.3 Ataque *De-Authentication*

Após uma estação cliente ter selecionado um ponto de acesso para se comunicar, ela deve primeiramente se autenticar junto ao mesmo antes que qualquer comunicação possa ocorrer. O *framework* de autenticação do padrão 802.11 possui também uma mensagem que permite que clientes e pontos de acesso requisitem explicitamente a sua desautenticação mútua. Infelizmente, essa mensagem não é autenticada usando nenhum tipo de criptografia. Consequentemente, o atacante pode falsificá-la, tentando se passar pelo ponto de acesso ou mesmo pelo cliente, e direcioná-la para a outra parte, conforme pode ser visualizado através da Figura 4.12. Em resposta, o ponto de acesso ou o cliente vai sair do estado autenticado e vai rejeitar todos os pacotes seguintes, até que a autenticação seja restabelecida (Bellardo and Savage, 2003). Com a repetição persistente desse ataque, um cliente pode ser privado de transmitir ou receber seus dados indefinidamente.

Os registros do ataque *de-authentication* foram montados de acordo com a Tabela 4.5. Nesses registros, os campos determinantes para a ocorrência do ata-

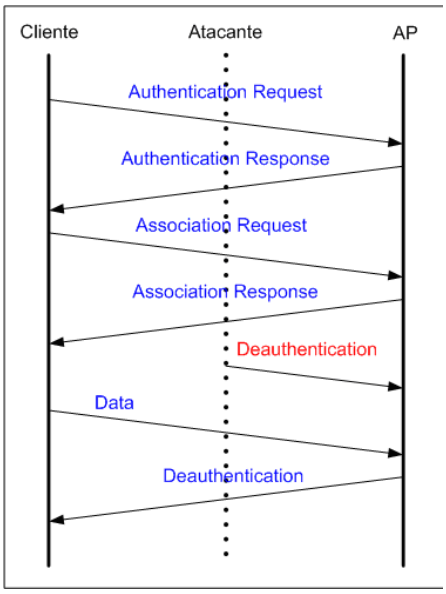

Figura 4.12: Ataque *De-authentication*.

que são DEAUTH (*Deauthentication*), MANAGE (*número de quadros do tipo Management*) e STATUS (*status do registro*).

Tabela 4.5: Formação dos Registros do Ataque *De-authentication*.

DEAUTH	MANAGE	STATUS
De 40 a 80	Igual a DEAUTH	1

O campo DEAUTH deve variar entre 40 e 80. O número de quadros MANAGE deve ser igual ao número de quadros DEAUTH. O campo STATUS deve ser igual a 1, para indicar a ocorrência do ataque.

4.6 Configuração do Ambiente de Simulação

As simulações do Módulo de Detecção foram realizadas em um computador com processador AMD Athlon 64 3700+ (2.2 GHz), com uma placa mãe Asus A8V-E SE e uma memória de 1 GB Ram DDR, rodando o Microsoft Windows XP SP2.

Para a prototipação do mecanismo de detecção de intrusão com redes neurais, foi utilizado o Toolbox de Redes Neurais do MATLAB. O MATLAB (MATLAB, 2007) possui um ambiente interativo que permite a realização de tarefas que

exigem cálculos matemáticos intensos, com uma prototipagem muito mais rápida do que linguagens de programação tradicionais, como C, C++ e Fortran. O Toolbox de Redes Neurais (NNET, 2007) estende o MATLAB com um conjunto poderoso de ferramentas para o projeto, implementação, visualização e simulação de redes neurais.

4.7 Implementação do Programa Simulador

O programa simulador foi implementado em linguagem MATLAB, utilizando o Toolbox de Redes Neurais. A operação desse programa é dividida em três etapas: treinamento, simulação e cálculo de índices, de acordo com a Figura 4.13. Na etapa de treinamento, a rede neural é criada e treinada com os dados do treino. Na etapa de simulação, a rede neural é simulada com os dados de teste e na etapa de cálculo de índices são calculados os índices que avaliam a aprendizagem e eficiência da aplicação da rede neural ao problema em questão.

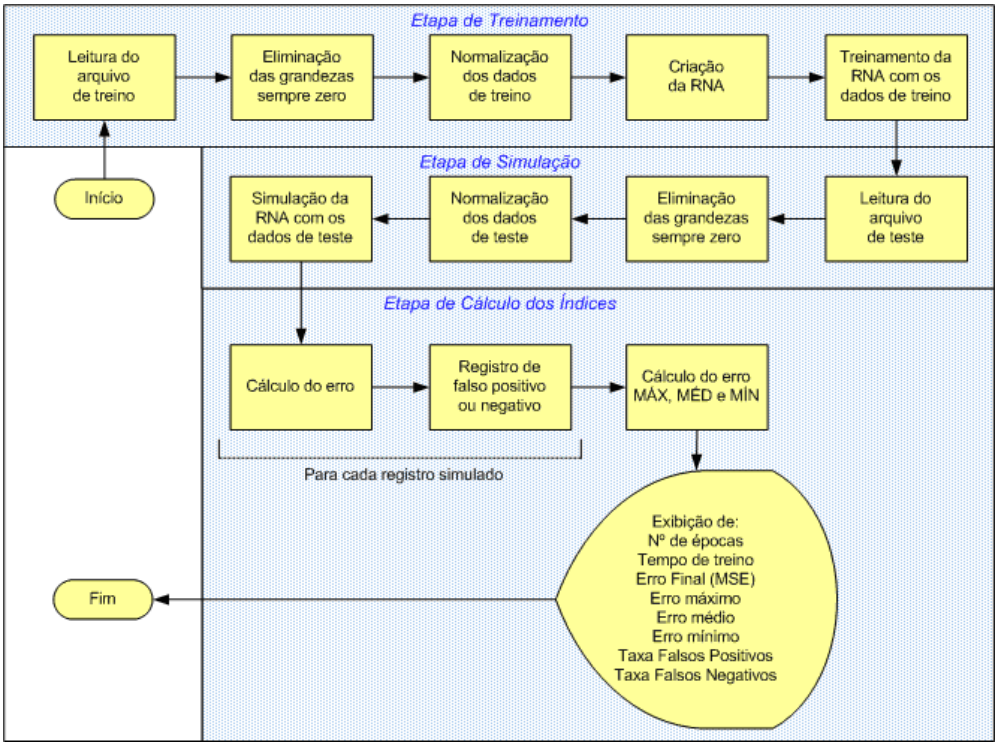

Figura 4.13: Fluxograma do Programa de Simulação.

Inicialmente, o programa realiza a leitura do arquivo com os registros de trei-

namento, denominado "treino.dat", gerando uma matriz preenchida com os dados do mesmo. Nessa matriz, as linhas representam os registros e as colunas representam as grandezas para cada registro.

Algumas grandezas na matriz de treinamento possuem valor zero em todos os registros, por isso são excluídas através da eliminação da coluna correspondente na matriz. As grandezas restantes são então normalizadas, dividindo-se o valor da variável pelo valor máximo entre todas as variáveis da mesma coluna, de acordo com a fórmula (4.1).

$$normalização(X_i) = \frac{X_i}{máximo(X_1, X_2, ..., X_n)} \tag{4.1}$$

O próximo passo é a criação da rede neural multicamada de perceptrons, com vinte unidades de entrada, sete neurônios na primeira camada escondida, cinco neurônios na segunda camada escondida e um neurônio na camada de saída. A arquitetura geral dessa rede neural pode ser visualizada através da Figura 4.14.

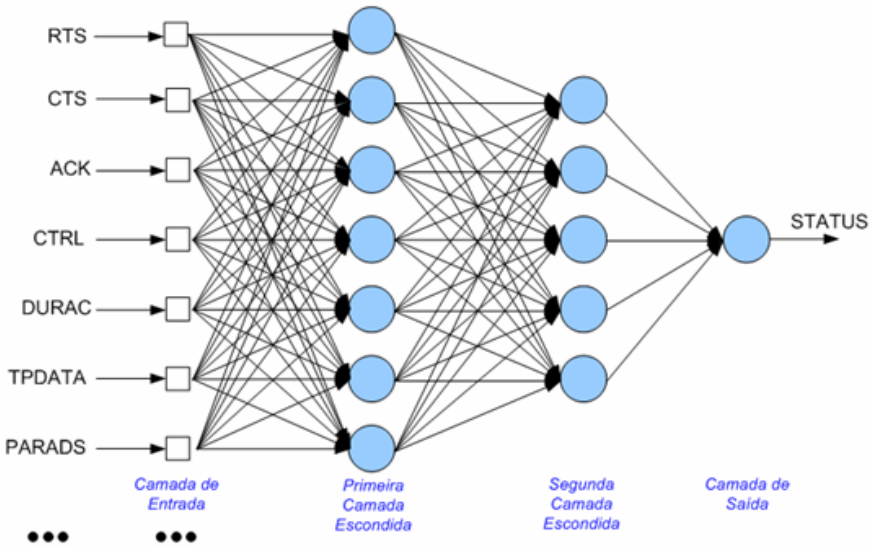

Figura 4.14: Arquitetura da Rede Neural Usada.

Para se chegar à configuração ideal para essa rede neural, foram realizados sucessivos testes com várias configurações possíveis, sendo de fundamental importância a flexibilidade do software MATLAB, que permite o ajuste de vários parâmetros da rede neural e do seu treinamento, possibilitando assim o teste de diversas arquiteturas de redes neurais em tempos relativamente curtos.

Depois de criada, a rede neural é treinada com os dados de treino, utilizando como algoritmo de treinamento o método de segunda ordem de Levenberg-Marquardt (LM) implementado em MATLAB.

No treinamento da rede neural, o número de épocas necessárias para o treinamento, o tempo gasto nesse treinamento e o MSE (*Mean Square Error*), que é uma função de custo que deve alcançar um valor mínimo possível no treinamento, são registrados.

O programa realiza a leitura do arquivo com os registros de teste, denominado "teste.dat", gerando uma matriz preenchida com os dados do mesmo. Nessa matriz, as linhas representam os registros e as colunas representam as grandezas para cada registro.

Algumas grandezas na matriz de teste possuem valor zero em todos os registros, por isso são excluídas através da eliminação da coluna correspondente na matriz. As grandezas restantes são então normalizadas, dividindo-se o valor da variável pelo valor máximo entre todas as variáveis da mesma coluna, de acordo com a fórmula (4.1).

O próximo passo é realizar a simulação da rede neural com os dados de teste, ou seja, os dados de teste são apresentados na entrada da rede neural, que dá como resultado um vetor onde cada posição corresponde ao resultado da simulação de um registro de teste. Esse vetor é comparado a um vetor com a saída correta para esse mesmo registro de teste. Assim, para cada registro é calculado o erro de acordo com a fórmula (4.2).

$$Erro = \left| \frac{resultadocorreto - resultadosimulado}{resultadosimulado} \right| * 100 \qquad (4.2)$$

Se o resultado correto for -1 (registro normal) e o resultado simulado for maior que -1, o erro calculado é registrado como falso positivo. Se o resultado correto for +1 (registro de ataque) e o resultado simulado for menor que 1, o erro calculado é registrado como falso negativo.

Depois de calculados o erro e as taxas de falso positivo e falso negativo para cada registro simulado pela rede neural, são calculados o erro máximo, erro médio, erro mínimo, taxa máxima de falso positivo e taxa máxima de falso negativo. Esses índices são exibidos na tela de resultados, juntamente com o número de épocas de treinamento, o tempo gasto no treinamento e o Erro Final (MSE).

4.8 Conclusão

Este capítulo apresentou a implementação de um protótipo para a arquitetura proposta, na qual foram implementados tanto o dispositivo sensor quanto o mecanismo de detecção de intrusões usando redes neurais. O próximo capítulo apresentará várias simulações e testes do protótipo implementado, com o objetivo de verificar a eficácia da aplicação de redes neurais na solução do problema da detecção de intrusos de redes *wireless*.

Capítulo 5

Simulações e Resultados

5.1 Introdução

Este capítulo apresenta várias simulações e testes do protótipo implementado, com o objetivo de verificar a eficácia da aplicação de redes neurais na solução do problema da detecção de intrusos de redes *wireless*. Os testes concentraram seu foco no poder de generalização das redes neurais, o que garante que o sistema detecte ataques ainda que estes apresentem características ligeiramente diferentes das já conhecidas.

5.2 Divisão dos Testes em Etapas

As redes neurais são capazes de generalizar seu conhecimento a partir de exemplos anteriores, ou seja, elas possuem a habilidade de lidar com ruídos e distorções, correspondendo corretamente a padrões de entrada fora do conjunto de treinamento. Quando são conhecidos os valores de uma função nos conjuntos de pontos X1, X2, ... Xn, resultado de amostragem ou de cálculos complexos, mas não é conhecida a sua expressão analítica f(x), a determinação de f(x) num ponto qualquer entre o menor e o maior dos Xi é denominada interpolação. A determinação de f(x) num ponto qualquer fora desse intervalo é denominada extrapolação. Isso pode ser visualizado através da Figura 5.1.

Para melhor analisar a capacidade de interpolação e extrapolação da rede neural, os testes foram divididos em 5 etapas: interpolação, extrapolação inferior,

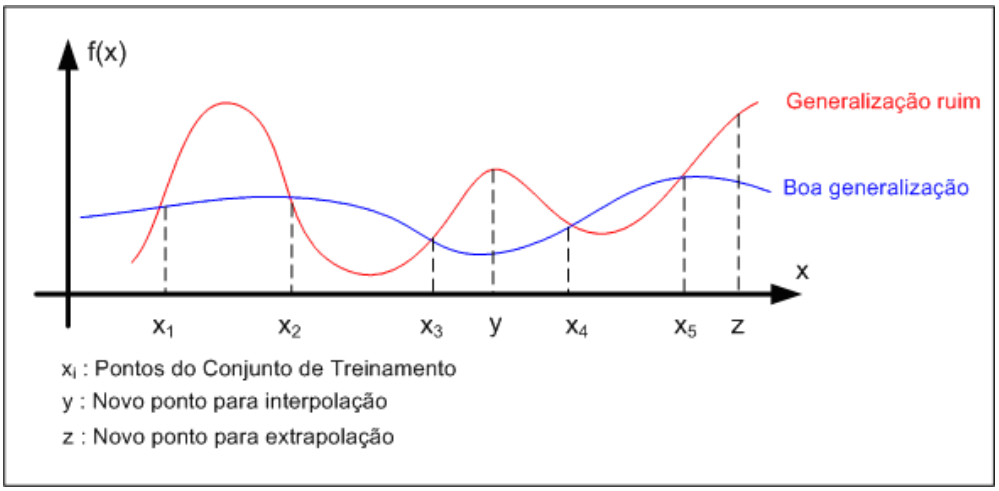

Figura 5.1: Interpolação e Extrapolação de Funções

extrapolação superior, extrapolação geral e generalização, de acordo com a Figura 5.2. Nesta figura, cada linha representa uma faixa de valores disponíveis para divisão entre os conjuntos de treinamento e testes.

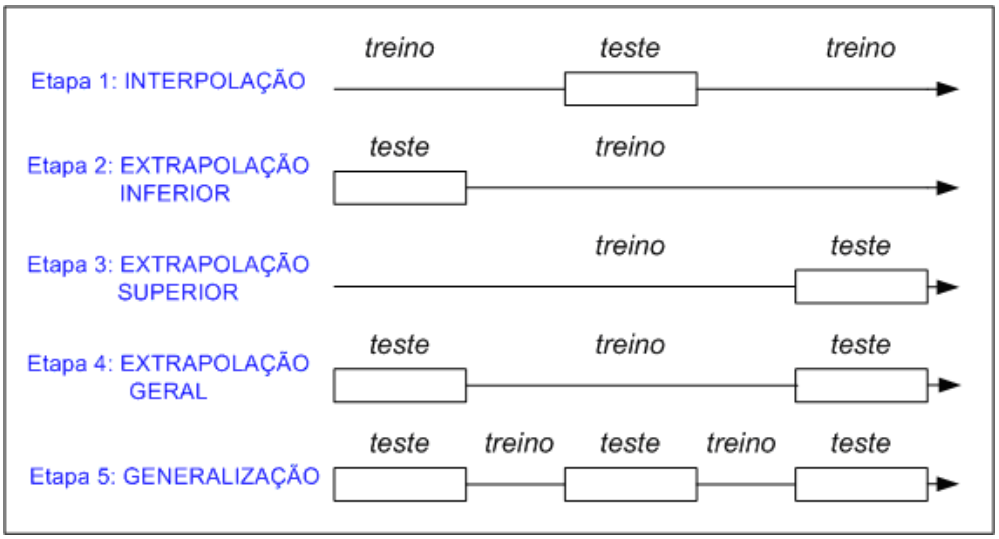

Figura 5.2: Etapas de Realização dos Testes.

Na Figura 5.2, as setas representam a faixa de valores disponíveis para treinamento e teste da rede neural. Na etapa 1 (interpolação), a rede neural é treinada apenas com os valores extremos e testada com os valores intermediários. Na etapa 2 (extrapolação inferior), a rede neural é treinada apenas com os valores intermediários e superiores, e testada com os valores inferiores. Na etapa 3 (ex-

trapolação superior), a rede neural é treinada apenas com os valores inferiores e intermediários, e testada com os valores superiores. Na etapa 4 (extrapolação geral), a rede neural é treinada apenas com os valores intermediários e testada com os valores extremos. Na etapa 5 (generalização), a rede neural é testada com os valores intermediários e extremos, tendo sido treinada com todos os outros valores restantes.

Para tornar possível a divisão dos testes nessas cinco etapas, as faixas de valores disponíveis para os três ataques aplicados nesse trabalho foram divididas de acordo com a Figura 5.3.

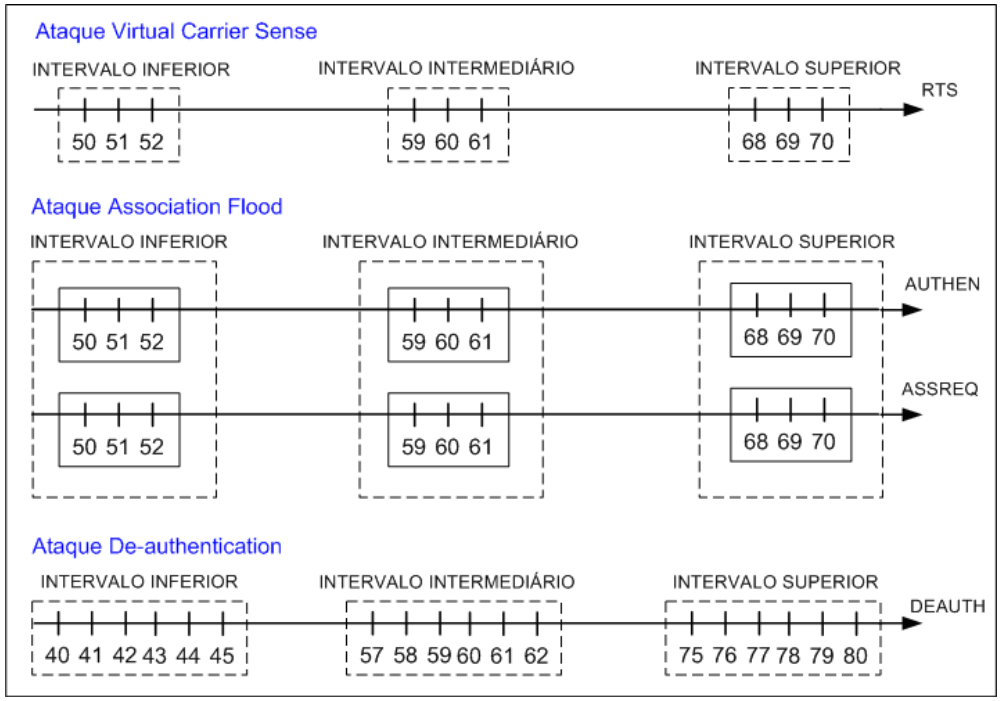

Figura 5.3: Faixas de Valores Para os Ataques.

No ataque *virtual carrier sense*, os registros com o campo RTS entre 50 e 52 foram inseridos no intervalo inferior, os registros com o campo RTS entre 59 e 61 foram inseridos no intervalo intermediário e os registros com o campo RTS entre 68 e 70 foram inseridos no intervalo superior.

No ataque *association flood*, os registros com o campo AUTHEN entre 50 e 52 foram inseridos no intervalo inferior, juntamente com os registros com o campo ASSREQ entre 50 e 52. Os registros com o campo AUTHEN entre 59 e 61 foram inseridos no intervalo intermediário, juntamente com os registros com o campo

ASSREQ entre 59 e 61. Os registros com o campo AUTHEN entre 68 e 70 foram inseridos no intervalo superior, juntamente com os registros com o campo ASSREQ entre 68 e 70.

No ataque *de-authentication*, os registros com o campo DEAUTH entre 40 e 45 foram inseridos no intervalo inferior, os registros com o campo DEAUTH entre 57 e 62 foram inseridos no intervalo intermediário e os registros com o campo DEAUTH entre 75 e 80 foram inseridos no intervalo superior.

Nas próximas seções, serão analisadas as simulações da rede neural com os dados de cada um dos ataques utilizados nesse trabalho: *virtual carrier sense*, *association flood* e *de-authentication*. Inicialmente, cada um desses ataques foi submetido individualmente às cinco etapas de avaliação do poder de generalização, além da validação do treinamento em si para cada ataque. Logo em seguida, eles foram simulados em grupos de dois, e os três ataques juntos. No final, os piores índices encontrados em todas as simulações para Erro Máximo, Taxa de Falsos Positivos e Taxa de Falsos Negativos foram concentrados em tabelas, para uma análise geral da rede neural.

5.3 Ataque 01 (*Virtual Carrier Sense*)

As Tabelas 5.1 e 5.2 apresentam a composição dos arquivos de treinamento e teste da rede neural, para a aplicação ao ataque *virtual carrier sense*. O arquivo de treinamento foi composto de 5000 registros normais e 2000 registros com o ataque *virtual carrier sense*, totalizando 7000 registros. O arquivo de teste foi composto de 2000 registros normais e 800 registros com o ataque *virtual carrier sense*, totalizando 2800 registros.

Tabela 5.1: Arquivo de Treino para o Ataque *Virtual Carrier Sense*.

TIPOS DE REGISTROS	Nº DE REGISTROS	PERCENTUAL
Normal	5000	71.42%
Virtual Carrier Sense	2000	28.57%
TOTAL	7000	100%

Tabela 5.2: Arquivo de Teste para o Ataque *Virtual Carrier Sense*.

TIPOS DE REGISTROS	Nº DE REGISTROS	PERCENTUAL
Normal	2000	71.42%
Virtual Carrier Sense	800	28.57%
TOTAL	2800	100%

5.3.1 Validação do Treinamento

Com o objetivo de avaliar o poder de aprendizagem da rede neural para os dados do ataque *virtual carrier sense*, foi realizada uma seqüência de treinamentos e simulações da rede neural, sendo que as simulações foram realizadas com os próprios dados utilizados nos respectivos treinamentos. Esses resultados são mostrados na Tabela 5.3.

De acordo com os resultados encontrados, o maior erro de treinamento foi de 3.0890%, ou seja, um erro grande considerando-se que a rede neural foi simulada com os próprios dados do treinamento. O maior erro quadrático médio do treinamento foi MSE = 9.04218e-007, que correspondeu também ao maior número de épocas de treinamento (29 épocas) e o maior tempo de CPU gasto no treinamento (13.4850 segundos).

Tabela 5.3: Validação do Treinamento para o Ataque *Virtual Carrier Sense*.

Seq	ÉPOCAS	MSE	ERRO MÁX	ERRO MÉD	ERRO MÍN	FALSO POS	FALSO NEG	TEMPO (seg)
1	9	2,91E-07	0.1382	0.0264	0.0141	0.0870	0.1382	4.3440
2	9	4,81E-07	0.1210	0.0729	6,02E-05	0.1210	0	4.3750
3	11	7,88E-07	2.2409	0.0156	1,06E-05	1.4756	0.2438	5.4690
4	5	6,43E-07	1.5130	0.0763	4,22E-04	1.5130	0.1206	2.6400
5	29	9,04E-07	3.0890	0.0081	2,48E-04	3.0853	0.1340	13.4850
6	8	3,24E-07	0.1014	0.0648	0.0026	0.1014	0.0354	3.9060
7	7	1,12E-07	0.1283	0.0339	0.0053	0.1283	0.0529	3.5000
8	7	2,83E-07	0.2359	0.0013	1,64E-05	0.1447	0.1264	3.5940
9	9	2,23E-07	0.1141	0.0313	3,07E-04	0.0779	0.0817	4.5780
10	8	6,80E-07	1.9147	0.0303	0.0061	1.9147	0.1371	3.9540

5.3.2 Interpolação

Os resultados da seqüência de treinamentos e simulações da rede neural, relativos à etapa de Interpolação para o ataque *virtual carrier sense*, são mostrados na

Tabela 5.4.

De acordo com os resultados encontrados, o maior erro de teste foi de 3.1588%, que corresponde à maior Taxa de Falsos Positivos encontrada, ficando bem acima da maior Taxa de Falsos Negativos encontrada, que foi de 1.4413%. O maior dos erros médios foi de 0.0624% e o maior dos erros mínimos foi de 0.01%.

Tabela 5.4: Resultados da Etapa de Interpolação para o Ataque *Virtual Carrier Sense*.

Seq	ÉPOCAS	MSE	ERRO MÁX	ERRO MÉD	ERRO MÍN	FALSO POS	FALSO NEG	TEMPO (seg)
1	7	1,37E-07	0,0642	0,0107	2,14E-04	0,0638	0,0039	3,453
2	8	3,07E-07	0,9229	0,0159	4,73E-04	0,9229	0,1308	4,11
3	10	5,31E-07	1,5677	0,0522	0,0085	1,5677	0,6921	4,906
4	5	1,97E-07	0,0835	0,006	8,42E-04	0,0835	0	2,625
5	11	2,92E-07	3,1588	0,0567	0,0085	3,1588	0	5,422
6	7	5,53E-07	1,004	0,007	1,30E-04	0,9454	0,373	3,656
7	8	2,76E-07	1,3907	0,0624	9,29E-04	0,1117	0	4,203
8	7	3,60E-07	0,1039	0,001	7,00E-04	0,0762	0,1039	3,718
9	19	8,75E-07	2,0168	0,0941	0,0077	2,0168	1,4413	8,907
10	7	3,02E-07	0,2894	0,0618	0,01	0,2894	0,0126	3,609

5.3.3 Extrapolação Inferior

Os resultados da seqüência de treinamentos e simulações da rede neural, relativos à etapa de Extrapolação Inferior para o ataque *virtual carrier sense*, são mostrados na Tabela 5.5.

De acordo com os resultados encontrados, o maior erro de teste foi de 3.1746%, que corresponde à maior Taxa de Falsos Positivos encontrada, ficando bem acima da maior Taxa de Falsos Negativos encontrada, que foi de 0.7343%. O maior dos erros médios foi de 0.0889% e o maior dos erros mínimos foi de 0.0147%.

5.3.4 Extrapolação Superior

Os resultados da seqüência de treinamentos e simulações da rede neural, relativos à etapa de Extrapolação Superior para o ataque *virtual carrier sense*, são mostrados na Tabela 5.6.

Tabela 5.5: Resultados da Etapa de Extrapolação Inferior para o Ataque *Virtual Carrier Sense*.

Seq	ÉPOCAS	MSE	ERRO MÁX	ERRO MÉD	ERRO MÍN	FALSO POS	FALSO NEG	TEMPO (seg)
1	10	1,73E-07	0,0798	0,0104	1,57E-04	0,0736	0,0798	4,844
2	47	9,35E-07	1,1616	0,0023	5,38E-05	0,2772	0,7343	21,391
3	8	2,45E-07	0,329	0,0576	0,0061	0,0881	0	3,922
4	11	2,62E-07	0,1507	0,0238	5,23E-05	0,0441	0,1507	5,188
5	8	4,22E-07	0,3957	0,0624	0,0014	0,3957	0,0521	3,859
6	9	8,50E-07	0,3812	0,048	0,0065	0,1619	0,1928	4,438
7	6	1,89E-07	0,8782	0,0056	0,0036	0,8782	0,0826	3,171
8	5	4,91E-07	2,7109	0,0488	0,0147	2,7109	0,039	2,829
9	8	5,74E-07	3,1746	0,0889	3,44E-05	3,1746	9,28E-04	4,172
10	7	4,34E-07	0,6935	0,0015	7,86E-04	0,6935	0,1152	3,625

De acordo com os resultados encontrados, o maior erro de teste foi de 3.9975%, que corresponde à maior Taxa de Falsos Positivos encontrada, ficando bem acima da maior Taxa de Falsos Negativos encontrada, que foi de 0.8706%. O maior dos erros médios foi de 0.0913% e o maior dos erros mínimos foi de 0.0152%.

Tabela 5.6: Resultados da Etapa de Extrapolação Superior para o Ataque *Virtual Carrier Sense*.

Seq	ÉPOCAS	MSE	ERRO MÁX	ERRO MÉD	ERRO MÍN	FALSO POS	FALSO NEG	TEMPO (seg)
1	13	6,38E-07	3,1398	0,0913	9,96E-04	3,1398	0,0275	6,125
2	9	9,37E-07	0,8706	0,0088	1,40E-04	0,0723	0,8706	4,312
3	13	8,54E-07	1,434	0,0029	2,16E-04	0,1748	0,4989	6,281
4	9	1,11E-07	0,697	0,0279	6,82E-04	0,0456	0,2166	4,328
5	9	1,59E-07	0,3628	0,0472	9,78E-05	0,3628	0,0015	4,359
6	9	2,10E-07	3,9975	0,054	0,0019	3,9975	0,271	4,5
7	11	2,57E-07	0,2036	0,0515	0,0152	0,2036	0,0486	5,422
8	7	5,56E-07	0,4783	0,0595	9,72E-05	0,4783	0,0596	3,672
9	4	2,06E-07	2,6361	0,0515	0,0024	0,3546	0,0967	2,328
10	11	7,79E-07	0,3922	0,0024	3,62E-05	0,026	0,3922	5,391

5.3.5 Extrapolação Geral

Os resultados da seqüência de treinamentos e simulações da rede neural, relativos à etapa de Extrapolação Geral para o ataque *virtual carrier sense*, são mostrados na Tabela 5.7.

De acordo com os resultados encontrados, o maior erro de teste foi de 1.1879%. A maior Taxa de Falsos Positivos encontrada foi de 1.1576% e a maior Taxa de Falsos Negativos encontrada foi de 0.3620%. O maior dos erros médios foi de 0.0957% e o maior dos erros mínimos foi de 0.0122%.

Tabela 5.7: Resultados da Etapa de Extrapolação Geral para o Ataque *Virtual Carrier Sense*.

Seq	ÉPOCAS	MSE	ERRO MÁX	ERRO MÉD	ERRO MÍN	FALSO POS	FALSO NEG	TEMPO (seg)
1	8	2,12E-07	0,1002	0,0447	0,0077	0,1002	0,0537	3,859
2	9	6,77E-07	1,1576	0,0019	2,76E-04	1,1576	0,362	4,25
3	7	9,85E-07	0,7015	0,0957	0,0122	0,7015	0,121	3,485
4	9	2,82E-07	0,1509	0,0627	0,0013	0,1509	0,0233	4,391
5	7	1,65E-07	0,836	0,0128	1,81E-05	0,0478	0,1005	3,64
6	11	5,64E-07	0,5839	0,0013	6,19E-05	0,5839	0,1447	5,172
7	6	4,27E-07	1,1879	0,0217	0,0026	0,0038	0,1226	3,016
8	9	1,93E-07	0,1041	0,052	0,0044	0,1041	0,0134	4,281
9	6	5,58E-07	0,8167	0,0816	3,35E-05	0,8167	0,0024	3
10	8	1,74E-07	0,804	0,0475	0,0024	0,804	0,0303	3,828

5.3.6 Generalização

Os resultados da seqüência de treinamentos e simulações da rede neural, relativos à etapa de Generalização para o ataque *virtual carrier sense*, são mostrados na Tabela 5.8.

De acordo com os resultados encontrados, o maior erro de teste foi de 8.8316%. A maior Taxa de Falsos Positivos encontrada foi de 3.666% e a maior Taxa de Falsos Negativos encontrada foi de 2.7750%. O maior dos erros médios foi de 0.1015% e o maior dos erros mínimos foi de 0.0046%.

5.4 Ataque 02 (*Association Flood*)

As Tabelas 5.9 e 5.10 apresentam a composição dos arquivos de treinamento e teste da rede neural, para a aplicação ao ataque *association flood*. O arquivo de treinamento foi composto de 5000 registros normais e 2000 registros com o ataque *association flood*, totalizando 7000 registros. O arquivo de teste foi composto de

Tabela 5.8: Resultados da Generalização da Rede Neural para o Ataque *Virtual Carrier Sense*.

Seq	ÉPOCAS	MSE	ERRO MÁX	ERRO MÉD	ERRO MÍN	FALSO POS	FALSO NEG	TEMPO (seg)
1	9	7,93E-07	2,775	0,0022	1,01E-04	1,1664	2,775	4,453
2	9	1,66E-07	0,6028	0,0358	0,004	0,6028	0,054	4,297
3	8	2,60E-07	0,2385	0,0049	4,71E-04	0,2385	0	3,812
4	8	3,61E-07	3,2095	0,0698	1,78E-04	0,3209	0,2021	3,844
5	6	5,49E-07	8,8316	0,0223	0,0025	3,666	0,2981	3,078
6	7	7,43E-07	0,653	0,1015	5,96E-05	0,653	5,47E-04	3,422
7	12	6,91E-07	0,6234	0,0031	1,22E-04	0,6234	0,2397	5,656
8	10	3,69E-07	0,1609	4,00E-04	8,06E-06	0,0366	0,1609	4,875
9	10	5,78E-07	3,809	0,0203	0,0046	0,7531	0,5054	4,781
10	8	1,77E-07	0,0868	0,0492	4,95E-04	0,0868	0,0121	3,922

2000 registros normais e 800 registros com o ataque *association flood*, totalizando 2800 registros.

Tabela 5.9: Arquivo de Treino para o Ataque *Association Flood*.

TIPOS DE REGISTROS	Nº DE REGISTROS	PERCENTUAL
Normal	5000	71.42%
Association Flood	2000	28.57%
TOTAL	7000	100%

Tabela 5.10: Arquivo de Teste para o Ataque *Association Flood*.

TIPOS DE REGISTROS	Nº DE REGISTROS	PERCENTUAL
Normal	2000	71.42%
Association Flood	800	28.57%
TOTAL	2800	100%

5.4.1 Validação do Treinamento

Com o objetivo de avaliar o poder de aprendizagem da rede neural para os dados do ataque *association flood*, foi realizada uma seqüência de treinamentos e simulações da rede neural, sendo que as simulações foram realizadas com os próprios dados utilizados nos respectivos treinamentos. Esses resultados são mostrados na Tabela 5.11.

De acordo com os resultados encontrados, o maior erro de treinamento foi de 1.6236%, ou seja, um erro grande considerando-se que a rede neural foi simulada com os próprios dados do treinamento. O maior erro quadrático médio do treinamento foi MSE = 8.30301e-007, que correspondeu também ao maior número de épocas de treinamento (13 épocas). O maior tempo de CPU gasto no treinamento foi de 8.25 segundos.

Tabela 5.11: Validação do Treinamento para o Ataque *Association Flood*.

Seq	ÉPOCAS	MSE	ERRO MÁX	ERRO MÉD	ERRO MÍN	FALSO POS	FALSO NEG	TEMPO (seg)
1	12	2,99E-07	0,2335	0,0647	5,78E-04	0,2335	0,0013	8,25
2	13	8,30E-07	1,6236	0,0553	1,07E-04	1,6236	0,2607	6,422
3	7	7,23E-07	0,1172	0,0996	0,0086	0,1172	0,0346	3,453
4	8	1,04E-07	1,3708	0,0042	0,0039	1,3708	0,0345	3,875
5	6	6,66E-07	0,2184	0,0429	0,0095	0,1271	0,2184	3,063
6	12	5,92E-07	0,9661	1,34E-04	8,86E-05	0,9661	0,2842	6,031
7	11	1,99E-07	0,0578	0,043	0,005	0,0578	0,0567	5,359
8	11	1,81E-07	0,0938	0,0444	8,32E-05	0,0938	0,0103	5,281
9	9	1,63E-07	0,0893	0,033	3,00E-05	0,0893	0,0717	4,328
10	8	2,56E-07	0,4327	0,0548	4,87E-05	0,3625	0,0968	4,25

5.4.2 Interpolação

Os resultados da seqüência de treinamentos e simulações da rede neural, relativos à etapa de Interpolação para o ataque *association flood*, são mostrados na Tabela 5.12.

De acordo com os resultados encontrados, o maior erro de teste foi de 2.9826%, que corresponde à maior Taxa de Falsos Positivos encontrada, ficando bem acima da maior Taxa de Falsos Negativos encontrada, que foi de 0.4544%. O maior dos erros médios foi de 0.0952% e o maior dos erros mínimos foi de 0.0129%.

5.4.3 Extrapolação Inferior

Os resultados da seqüência de treinamentos e simulações da rede neural, relativos à etapa de Extrapolação Inferior para o ataque *association flood*, são mostrados na Tabela 5.13.

Tabela 5.12: Resultados da Etapa de Interpolação para o Ataque *Association Flood*.

Seq	ÉPOCAS	MSE	ERRO MÁX	ERRO MÉD	ERRO MÍN	FALSO POS	FALSO NEG	TEMPO (seg)
1	11	2,15E-07	0,4023	0,0522	0,0049	0	0	5,219
2	9	7,46E-07	2,9826	0,0952	0,0129	2,9826	0,0197	4,469
3	7	3,67E-07	0,3959	0,0015	1,19E-04	0,3959	0,0347	3,438
4	7	7,07E-07	0,2576	0,0447	0,0037	0,1386	0,2576	3,516
5	7	5,66E-07	0,2729	0,0493	0,0018	0,2729	0,1096	3,454
6	8	3,24E-07	0,2285	0,0268	6,18E-05	0,1986	0,2285	3,859
7	8	7,35E-07	1,9045	0,0696	0,003	1,9045	0,07	4,047
8	8	5,35E-07	2,2171	0,0716	0,0031	2,2171	0,4544	4
9	9	6,45E-07	1,1422	0,0936	0,0018	0,1674	0,1279	4,532
10	7	5,44E-07	0,196	0,0263	6,03E-05	0,032	0,196	3,5

De acordo com os resultados encontrados, o maior erro de teste foi de 4.7420%, que corresponde à maior Taxa de Falsos Positivos encontrada, ficando bem acima da maior Taxa de Falsos Negativos encontrada, que foi de 0.3632%. O maior dos erros médios foi de 0.0639% e o maior dos erros mínimos foi de 0.0180%.

Tabela 5.13: Resultados da Etapa de Extrapolação Inferior para o Ataque *Association Flood*.

Seq	ÉPOCAS	MSE	ERRO MÁX	ERRO MÉD	ERRO MÍN	FALSO POS	FALSO NEG	TEMPO (seg)
1	7	2,09E-07	4,742	0,0396	0,0179	4,742	0,0971	3,625
2	10	8,74E-07	0,2648	0,0639	0,018	0,2648	0,2272	4,781
3	8	8,60E-07	0,3632	0,0011	1,01E-04	0,0039	0,3632	3,813
4	7	4,13E-07	0,2171	0,0197	1,46E-05	0,2171	0,1356	3,422
5	6	4,02E-07	0,2038	0,0443	2,14E-06	0,0653	0,1021	3,078
6	9	3,09E-07	0,0929	0,0371	0,0127	0,0929	0,0923	4,344
7	11	2,26E-07	1,335	0,0533	4,82E-06	1,335	0,0335	5,187
8	13	3,02E-07	1,0329	0,0587	0,0071	1,0329	0,0377	6,172
9	9	7,45E-07	0,1661	0,0117	3,64E-05	0,0183	0,1661	4,391
10	9	1,81E-07	1,924	0,0388	0,0122	0,1404	0,0537	4,328

5.4.4 Extrapolação Superior

Os resultados da seqüência de treinamentos e simulações da rede neural, relativos à etapa de Extrapolação Superior para o ataque *association flood*, são mostrados na Tabela 5.14.

De acordo com os resultados encontrados, o maior erro de teste foi de 2.1359%. A maior Taxa de Falsos Positivos encontrada foi de 1.3515% e a maior Taxa de Falsos Negativos encontrada foi de 2.1359%. O maior dos erros médios foi de 0.0537% e o maior dos erros mínimos foi de 0.0098%.

Tabela 5.14: Resultados da Etapa de Extrapolação Superior para o Ataque *Association Flood*.

Seq	ÉPOCAS	MSE	ERRO MÁX	ERRO MÉD	ERRO MÍN	FALSO POS	FALSO NEG	TEMPO (seg)
1	9	4,43E-07	1,9717	1,06E-04	8,06E-06	0,0014	1,9717	4,406
2	7	3,60E-07	0,2814	0,0537	0,0042	0,1896	0,2814	3,422
3	9	1,79E-07	0,0688	0,0176	0,0023	0,0247	0,0688	4,454
4	12	3,41E-07	1,4021	0,0388	0,0098	0,1207	0,5733	5,75
5	10	2,62E-07	0,1789	0,0166	2,29E-05	0,1789	0,0869	4,875
6	8	8,78E-07	1,9608	0,052	4,35E-05	1,1303	0	4,156
7	5	2,20E-07	2,1359	0,0178	3,63E-04	1,3515	2,1359	3,093
8	8	3,07E-07	0,3455	0,0511	0,0054	0,3455	0	3,829
9	9	1,76E-07	0,1199	0,0163	1,91E-06	0,0712	0	4,422
10	9	9,04E-07	0,1817	0,0054	3,57E-04	0,0285	0,1817	4,39

5.4.5 Extrapolação Geral

Os resultados da seqüência de treinamentos e simulações da rede neural, relativos à etapa de Extrapolação Geral para o ataque *association flood*, são mostrados na Tabela 5.15.

De acordo com os resultados encontrados, o maior erro de teste foi de 3.8336%. A maior Taxa de Falsos Positivos encontrada foi de 2.3701% e a maior Taxa de Falsos Negativos encontrada foi de 1.6884%. O maior dos erros médios foi de 0.0704% e o maior dos erros mínimos foi de 0.0321%.

5.4.6 Generalização

Os resultados da seqüência de treinamentos e simulações da rede neural, relativos à etapa de Generalização para o ataque *association flood*, são mostrados na Tabela 5.16.

De acordo com os resultados encontrados, o maior erro de teste foi de 1.3265%, que corresponde à maior Taxa de Falsos Positivos encontrada, ficando bem acima

Tabela 5.15: Resultados da Etapa de Extrapolação Geral para o Ataque *Association Flood*.

Seq	ÉPOCAS	MSE	ERRO MÁX	ERRO MÉD	ERRO MÍN	FALSO POS	FALSO NEG	TEMPO (seg)
1	8	9,17E-07	0,2781	0,0344	1,20E-04	0,1763	0,2142	4,125
2	10	6,81E-07	0,9048	0,0704	0,0039	0,9048	0,132	4,812
3	6	5,44E-07	2,3701	0,0265	9,46E-05	2,3701	0,0041	3,172
4	6	1,52E-07	0,2164	0,0072	0,0046	0,0611	0,2164	3,016
5	10	1,41E-07	0,3797	0,0407	6,15E-05	0,3797	0,0479	4,719
6	7	3,29E-07	1,6884	0,043	4,27E-04	0,0925	1,6884	3,469
7	7	2,39E-07	0,9051	0,0465	0,0321	0,9017	0,9051	3,578
8	9	7,31E-07	3,8336	0,0184	9,62E-04	0,2038	0,1462	4,547
9	9	1,96E-07	1,451	0,0051	6,10E-06	0,214	1,451	4,329
10	9	6,16E-07	2,3515	0,0667	0,007	2,3515	0,2405	4,312

da maior Taxa de Falsos Negativos encontrada, que foi de 0.3058%. O maior dos erros médios foi de 0.1126% e o maior dos erros mínimos foi de 0.0221%.

Tabela 5.16: Resultados da Generalização da Rede Neural para o Ataque *Association Flood*.

Seq	ÉPOCAS	MSE	ERRO MÁX	ERRO MÉD	ERRO MÍN	FALSO POS	FALSO NEG	TEMPO (seg)
1	9	2,52E-07	0,4449	0,0575	0,0067	0,252	0,1455	4,313
2	9	3,09E-07	0,2203	0,0083	1,55E-07	0,0187	0,2203	4,484
3	8	2,94E-07	0,1177	0,0604	0,0023	0,0622	0,1177	3,828
4	7	8,33E-07	0,2318	0,0419	2,46E-05	0,105	0,2318	3,515
5	9	5,79E-08	0,0851	0,0043	5,33E-05	0,0851	0	4,266
6	11	9,52E-07	0,3058	0,1126	0,0145	0,2673	0,3058	5,234
7	10	2,53E-07	1,3265	0,0277	6,58E-04	1,3265	0,1484	4,813
8	9	2,74E-07	0,4238	0,0454	0,0221	0,4238	0,1137	4,328
9	8	7,46E-07	0,1675	0,0034	5,36E-05	0,0251	0,1675	4,063
10	7	2,32E-07	1,1206	0,0186	0,0032	0,0059	0,1406	3,531

5.5 Ataque 03 (*De-authentication*)

As Tabelas 5.17 e 5.18 apresentam a composição dos arquivos de treinamento e teste da rede neural, para a aplicação ao ataque *de-authentication*. O arquivo de treinamento foi composto de 5000 registros normais e 2000 registros com o ataque *de-authentication*, totalizando 7000 registros. O arquivo de teste foi composto de

2000 registros normais e 800 registros com o ataque *de-authentication*, totalizando 2800 registros.

Tabela 5.17: Arquivo de Treino para o Ataque *De-authentication*.

TIPOS DE REGISTROS	Nº DE REGISTROS	PERCENTUAL
Normal	5000	71.42%
De-authentication	2000	28.57%
TOTAL	7000	100%

Tabela 5.18: Arquivo de Teste para o Ataque *De-authentication*.

TIPOS DE REGISTROS	Nº DE REGISTROS	PERCENTUAL
Normal	2000	71.42%
De-authentication	800	28.57%
TOTAL	2800	100%

5.5.1 Validação do Treinamento

Com o objetivo de avaliar o poder de aprendizagem da rede neural para os dados do ataque *de-authentication*, foi realizada uma seqüência de treinamentos e simulações da rede neural, sendo que as simulações foram realizadas com os próprios dados utilizados nos respectivos treinamentos. Esses resultados são mostrados na Tabela 5.19.

De acordo com os resultados encontrados, o maior erro de treinamento foi de 1.9721%, ou seja, um erro grande considerando-se que a rede neural foi simulada com os próprios dados do treinamento. O maior erro quadrático médio do treinamento foi MSE = 9.48932e-007. O maior número de épocas de treinamento foi de 20 épocas, que correspondeu também ao maior tempo de CPU gasto no treinamento (9.1720 segundos).

5.5.2 Interpolação

Os resultados da seqüência de treinamentos e simulações da rede neural, relativos à etapa de Interpolação para o ataque *de-authentication*, são mostrados na Tabela 5.20.

Tabela 5.19: Validação do Treinamento para o Ataque *De-authentication*.

Seq	ÉPOCAS	MSE	ERRO MÁX	ERRO MÉD	ERRO MÍN	FALSO POS	FALSO NEG	TEMPO (seg)
1	10	2,84E-07	1,3592	0,0528	0,0036	1,3592	0,0364	4,703
2	11	3,43E-07	0,2141	0,0692	0,002	0,2141	0	5,093
3	7	5,62E-07	0,756	0,0832	8,15E-05	0,756	0,0714	3,516
4	14	5,84E-07	0,8589	0,0557	3,75E-05	0,511	0,0805	6,563
5	10	6,98E-07	1,9721	0,0088	1,83E-06	1,4215	0,4287	5
6	12	1,74E-07	0,1045	0,0495	1,78E-05	0,1045	0,0126	5,657
7	11	4,46E-07	1,7781	0,06	2,43E-05	1,7781	0,2041	5,281
8	9	9,49E-07	1,9593	0,0522	0,0289	1,9593	0,3098	4,391
9	7	3,73E-07	0,1568	0,0026	5,53E-06	0,1532	0,1568	3,532
10	20	6,41E-07	0,1506	0,0067	0,003	0	0,1506	9,172

De acordo com os resultados encontrados, o maior erro de teste foi de 5.9363%, que corresponde à maior Taxa de Falsos Positivos encontrada, ficando bem acima da maior Taxa de Falsos Negativos encontrada, que foi de 0.0620%. O maior dos erros médios foi de 0.1061% e o maior dos erros mínimos foi de 0.0367%.

Tabela 5.20: Resultados da Etapa de Interpolação para o Ataque *De-authentication*.

Seq	ÉPOCAS	MSE	ERRO MÁX	ERRO MÉD	ERRO MÍN	FALSO POS	FALSO NEG	TEMPO (seg)
1	9	1,17E-07	0,594	0,0245	0,008	0,594	0	4,343
2	9	5,19E-07	0,1462	0,0763	0,0367	0,1462	0,062	4,328
3	14	8,63E-07	2,7872	0,0413	8,04E-05	2,7872	0	6,593
4	8	2,37E-07	0,0746	0,0442	0,0011	0,0746	0,0577	3,907
5	8	1,52E-07	0,284	0,0457	6,18E-04	0,284	7,06E-04	3,907
6	14	1,95E-07	2,1471	0,0037	1,34E-05	1,5263	0,0075	6,797
7	34	9,75E-07	3,4841	0,0217	6,12E-04	3,4841	0	15,609
8	30	6,85E-07	4,3747	0,0484	2,70E-05	2,7521	0	13,75
9	9	9,03E-07	5,9363	0,1061	0,0074	5,9363	0,0119	4,391
10	10	7,13E-07	3,9091	0,0811	0,0073	3,9091	0,0119	4,875

5.5.3 Extrapolação Inferior

Os resultados da seqüência de treinamentos e simulações da rede neural, relativos à etapa de Extrapolação Inferior para o ataque *de-authentication*, são mostrados na Tabela 5.21.

De acordo com os resultados encontrados, o maior erro de teste foi de 5.3272%, que corresponde à maior Taxa de Falsos Positivos encontrada, ficando bem acima da maior Taxa de Falsos Negativos encontrada, que foi de 0.1529%. O maior dos erros médios foi de 0.11% e o maior dos erros mínimos foi de 0.0099%.

Tabela 5.21: Resultados da Etapa de Extrapolação Inferior para o Ataque *De-authentication*.

Seq	ÉPOCAS	MSE	ERRO MÁX	ERRO MÉD	ERRO MÍN	FALSO POS	FALSO NEG	TEMPO (seg)
1	7	4,17E-07	5,1922	0,0028	6,97E-06	5,1922	0,0831	3,578
2	45	7,51E-07	2,4994	0,0409	5,69E-05	1,8553	0	20,281
3	10	8,55E-07	0,1795	0,11	0,001	0,1795	0,0021	4,968
4	11	6,40E-07	0,1792	0,0615	4,83E-05	0,1792	0,1102	5,235
5	9	8,32E-07	2,1348	0,0041	4,44E-05	0,9918	0,0256	4,578
6	8	9,86E-07	2,0101	0,0663	1,27E-04	1,9682	0	3,984
7	17	8,65E-07	2,3038	0,1018	0,0099	2,3038	0,0536	8,031
8	7	5,36E-07	0,5412	0,0036	9,63E-04	0,5412	0,1218	3,453
9	51	9,61E-07	2,704	0,0247	5,16E-05	2,704	0,0879	22,968
10	39	9,60E-07	5,3272	0,0167	3,27E-05	5,3272	0,1529	17,734

5.5.4 Extrapolação Superior

Os resultados da seqüência de treinamentos e simulações da rede neural, relativos à etapa de Extrapolação Superior para o ataque *de-authentication*, são mostrados na Tabela 5.22.

De acordo com os resultados encontrados, o maior erro de teste foi de 6.8098%, que corresponde à maior Taxa de Falsos Positivos encontrada, ficando bem acima da maior Taxa de Falsos Negativos encontrada, que foi de 0.1511%. O maior dos erros médios foi de 0.0931% e o maior dos erros mínimos foi de 0.0117%.

5.5.5 Extrapolação Geral

Os resultados da seqüência de treinamentos e simulações da rede neural, relativos à etapa de Extrapolação Geral para o ataque *de-authentication*, são mostrados na Tabela 5.23.

De acordo com os resultados encontrados, o maior erro de teste foi de 6.5416%, que corresponde à maior Taxa de Falsos Positivos encontrada, ficando bem acima

Tabela 5.22: Resultados da Etapa de Extrapolação Superior para o Ataque *De-authentication*.

Seq	ÉPOCAS	MSE	ERRO MÁX	ERRO MÉD	ERRO MÍN	FALSO POS	FALSO NEG	TEMPO (seg)
1	12	6,21E-07	0,1135	0,0931	0,0117	0,1135	0,0123	5,625
2	16	5,92E-07	6,7093	0,0722	2,20E-04	4,1293	0,024	7,5
3	11	8,04E-07	6,8098	0,0158	8,81E-05	6,8098	0,1511	5,344
4	6	2,09E-07	1,918	0,0334	0,0033	1,918	0,0249	2,969
5	11	2,46E-07	0,1543	0,0084	1,34E-04	0,0173	0,0938	5,094
6	12	6,78E-07	0,3891	0,0195	1,20E-04	0,3891	0,15	5,594
7	8	2,39E-07	0,752	0,0503	0,0012	0,752	0,0171	3,922
8	7	1,63E-07	1,4849	0,0057	0,0027	0	0,0752	3,469
9	8	5,99E-07	3,2398	0,0459	0,0097	3,2398	0,0287	4
10	9	6,63E-07	3,8906	0,0521	0,0117	3,8906	0,1218	4,359

da maior Taxa de Falsos Negativos encontrada, que foi de 2.7774%. O maior dos erros médios foi de 0.1088% e o maior dos erros mínimos foi de 0.0054%.

Tabela 5.23: Resultados da Etapa de Extrapolação Geral para o Ataque *De-authentication*.

Seq	ÉPOCAS	MSE	ERRO MÁX	ERRO MÉD	ERRO MÍN	FALSO POS	FALSO NEG	TEMPO (seg)
1	10	6,15E-07	1,9337	0,0774	1,30E-05	1,4306	1,9337	5,047
2	7	2,68E-07	0,4464	5,73E-04	1,55E-05	0,4464	0,2286	3,609
3	9	2,00E-07	3,6745	0,0506	0,0028	3,6745	0,6617	4,406
4	13	7,71E-07	2,9448	0,0404	6,39E-04	2,9448	2,7774	6,219
5	7	8,02E-07	0,2916	0,1012	4,19E-04	0,2916	0,1568	3,421
6	21	2,81E-07	6,5416	0,0455	0,0047	6,5416	0,9617	9,875
7	9	1,94E-08	0,255	0,0153	0,0054	0,255	0,2141	4,265
8	8	4,88E-07	2,2275	0,0806	3,13E-04	2,0457	1,861	4
9	8	1,76E-07	0,9627	0,0494	0,0042	0,0794	0,0443	4,046
10	14	8,80E-07	1,6745	0,1088	5,57E-05	1,621	1,3223	6,75

5.5.6 Generalização

Os resultados da seqüência de treinamentos e simulações da rede neural, relativos à etapa de Generalização para o ataque *de-authentication*, são mostrados na Tabela 5.24.

De acordo com os resultados encontrados, o maior erro de teste foi de 6.3488%, que corresponde à maior Taxa de Falsos Positivos encontrada, ficando bem acima

da maior Taxa de Falsos Negativos encontrada, que foi de 3.9490%. O maior dos erros médios foi de 0.1046% e o maior dos erros mínimos foi de 0.0055%.

Tabela 5.24: Resultados da Generalização da Rede Neural para o Ataque *De-authentication*.

Seq	ÉPOCAS	MSE	ERRO MÁX	ERRO MÉD	ERRO MÍN	FALSO POS	FALSO NEG	TEMPO (seg)
1	10	3,50E-07	0,2384	1,74E-04	1,12E-07	0,0012	0,2384	4,657
2	7	3,12E-07	0,1627	0,0289	0,0055	0,1627	0,0708	3,375
3	7	5,20E-07	1,0072	0,0027	6,75E-05	0,2171	1,0072	3,469
4	8	2,90E-07	6,3488	0,046	0,0016	6,3488	2,1067	4,079
5	10	4,29E-07	0,3397	0,0444	6,12E-04	0,1035	0,3397	5,047
6	9	6,59E-07	0,3045	0,0887	7,10E-05	0,3045	0,0829	4,407
7	8	3,07E-07	2,109	0,024	1,15E-04	2,109	0,1668	3,937
8	8	8,62E-07	1,092	0,004	2,28E-04	1,092	0,4928	3,812
9	30	8,31E-07	3,949	0,0307	3,08E-05	1,8106	3,949	13,843
10	8	8,31E-07	0,3426	0,1046	5,37E-04	0,2253	0,3426	4,094

5.6 Ataque 01 com Ataque 02

As Tabelas 5.25 e 5.26 apresentam a composição dos arquivos de treinamento e teste da rede neural, para a aplicação aos ataques *virtual carrier sense* e *association flood* juntos. O arquivo de treinamento foi composto de 10000 registros normais e 4000 registros com os ataques *virtual carrier sense* e *association flood*, totalizando 14000 registros. O arquivo de teste foi composto de 4000 registros normais e 1600 registros com os ataques *virtual carrier sense* e *association flood*, totalizando 5600 registros.

Tabela 5.25: Arquivo de Treino para os Ataques 01 e 02.

TIPOS DE REGISTROS	Nº DE REGISTROS	PERCENTUAL
Normal	10000	71.42%
Virtual Carrier Sense	2000	14.28%
Association Flood	2000	14.28%
TOTAL	14000	100%

Os resultados da seqüência de treinamentos e simulações da rede neural, relativos à etapa de Generalização para os ataques *virtual carrier sense* e *association flood* juntos, são mostrados na Tabela 5.27.

Tabela 5.26: Arquivo de Teste para os Ataques 01 e 02.

TIPOS DE REGISTROS	Nº DE REGISTROS	PERCENTUAL
Normal	4000	71.42%
Virtual Carrier Sense	800	14.28%
Association Flood	800	14.28%
TOTAL	5600	100%

De acordo com os resultados encontrados, o maior erro de teste foi de 9.5958%, que corresponde à maior Taxa de Falsos Negativos encontrada. A maior Taxa de Falsos Positivos encontrada foi de 3.5670%. O maior dos erros médios foi de 0.0872% e o maior dos erros mínimos foi de 0.0231%.

Tabela 5.27: Resultados da Generalização da Rede Neural para os Ataques 01 e 02.

Seq	ÉPOCAS	MSE	ERRO MÁX	ERRO MÉD	ERRO MÍN	FALSO POS	FALSO NEG	TEMPO (seg)
1	8	3,28E-07	0,1738	0,0213	7,19E-04	0,0765	0,0697	11,516
2	17	9,02E-07	1,9162	0,008	3,60E-05	0,5595	1,9162	23,766
3	10	4,68E-07	0,9773	0,002	5,97E-06	0,0543	0,9773	14,172
4	10	6,63E-07	9,5958	0,0758	3,43E-04	3,567	9,5958	14,313
5	9	4,34E-07	0,3005	0,0737	0,0231	0,3005	0,1746	12,765
6	10	7,53E-07	1,4394	0,0099	0,0017	0,0191	0,518	14,047
7	6	5,44E-07	0,088	0,0872	0,0033	0,088	0,017	8,813
8	13	4,55E-07	0,1327	0,0203	0,0012	0,0446	0,1327	17,922
9	10	3,09E-07	1,8115	0,0016	6,92E-05	0,1307	0,3695	14,141
10	8	5,90E-07	0,3187	0,0648	0,0076	0,1645	0,3187	11,39

5.7 Ataque 01 com Ataque 03

As Tabelas 5.28 e 5.29 apresentam a composição dos arquivos de treinamento e teste da rede neural, para a aplicação aos ataques *virtual carrier sense* e *de-authentication* juntos. O arquivo de treinamento foi composto de 10000 registros normais e 4000 registros com os ataques *virtual carrier sense* e *de-authentication*, totalizando 14000 registros. O arquivo de teste foi composto de 4000 registros normais e 1600 registros com os ataques *virtual carrier sense* e *de-authentication*, totalizando 5600 registros.

Tabela 5.28: Arquivo de Treino para os Ataques 01 e 03.

TIPOS DE REGISTROS	Nº DE REGISTROS	PERCENTUAL
Normal	10000	71.42%
Virtual Carrier Sense	2000	14.28%
De-authentication	2000	14.28%
TOTAL	14000	100%

Tabela 5.29: Arquivo de Teste para os ataques 01 e 03.

TIPOS DE REGISTROS	Nº DE REGISTROS	PERCENTUAL
Normal	4000	71.42%
Virtual Carrier Sense	800	14.28%
De-authentication	800	14.28%
TOTAL	5600	100%

Os resultados da seqüência de treinamentos e simulações da rede neural, relativos à etapa de Generalização para os ataques *virtual carrier sense* e *de-authentication* juntos, são mostrados na Tabela 5.30.

par De acordo com os resultados encontrados, o maior erro de teste foi de 9.3851%, que corresponde à maior Taxa de Falsos Positivos encontrada. A maior Taxa de Falsos Negativos encontrada foi de 8.4774%. O maior dos erros médios foi de 0.1048% e o maior dos erros mínimos foi de 0.0075%.

Tabela 5.30: Resultados da Generalização da Rede Neural para os Ataques 01 e 03.

Seq	ÉPOCAS	MSE	ERRO MÁX	ERRO MÉD	ERRO MÍN	FALSO POS	FALSO NEG	TEMPO (seg)
1	8	8,47E-07	8,4774	0,0068	1,33E-05	1,6643	8,4774	11,812
2	10	1,69E-07	1,3216	0,0472	0,0011	0,0672	0,0665	14,234
3	17	8,26E-07	5,7516	0,0389	5,28E-05	5,7516	1,5482	23,516
4	12	4,82E-07	0,2359	0,0801	0,0026	0,2359	0,0313	16,703
5	13	9,89E-07	2,3471	0,0144	2,19E-04	2,0375	2,3471	18,437
6	8	2,59E-07	0,1514	0,0143	1,89E-04	0,1115	0,1514	11,422
7	6	4,16E-07	4,867	0,0252	7,06E-04	4,867	0,2534	9,031
8	10	7,99E-07	0,8843	0,1048	0,0075	0,8843	0,2734	14,266
9	41	9,85E-07	9,3851	0,0266	7,72E-06	9,3851	6,1765	54,469
10	8	6,93E-07	8,3773	0,069	1,32E-04	8,3773	0,8783	11,813

5.8 Ataque 02 com Ataque 03

As Tabelas 5.31 e 5.32 apresentam a composição dos arquivos de treinamento e teste da rede neural, para a aplicação aos ataques *association flood* e *de-authentication* juntos. O arquivo de treinamento foi composto de 10000 registros normais e 4000 registros com os ataques *association flood* e *de-authentication*, totalizando 14000 registros. O arquivo de teste foi composto de 4000 registros normais e 1600 registros com os ataques *association flood* e *de-authentication*, totalizando 5600 registros.

Tabela 5.31: Arquivo de Treino para os Ataques 02 e 03.

TIPOS DE REGISTROS	Nº DE REGISTROS	PERCENTUAL
Normal	10000	71.42%
Association Flood	2000	14.28%
De-authentication	2000	14.28%
TOTAL	14000	100%

Tabela 5.32: Arquivo de Teste para os Ataques 02 e 03.

TIPOS DE REGISTROS	Nº DE REGISTROS	PERCENTUAL
Normal	4000	71.42%
Association Flood	800	14.28%
De-authentication	800	14.28%
TOTAL	5600	100%

Os resultados da seqüência de treinamentos e simulações da rede neural, relativos à etapa de Generalização para os ataques *association flood* e *de-authentication* juntos, são mostrados na Tabela 5.33.

De acordo com os resultados encontrados, o maior erro de teste foi de 9.8141%, que corresponde à maior Taxa de Falsos Positivos encontrada. A maior Taxa de Falsos Negativos encontrada foi de 4.1631%. O maior dos erros médios foi de 0.1115% e o maior dos erros mínimos foi de 0.0349%.

5.9 Ataque 01 com Ataque 02 e Ataque 03

As Tabelas 5.34 e 5.35 apresentam a composição dos arquivos de treinamento e teste da rede neural, para a aplicação aos ataques *virtual carrier sense*, *association*

Tabela 5.33: Resultados da Generalização da Rede Neural para os Ataques 02 e 03.

Seq	ÉPOCAS	MSE	ERRO MÁX	ERRO MÉD	ERRO MÍN	FALSO POS	FALSO NEG	TEMPO (seg)
1	10	8,86E-07	0,3201	0,1115	1,58E-04	0,2477	0,0054	13,906
2	10	5,30E-07	0,4906	0,0136	0,0036	0,4906	0,1778	14,047
3	35	9,89E-07	9,8141	0,0126	9,78E-06	9,8141	2,8599	46,782
4	8	4,49E-07	0,6483	0,0791	9,64E-04	0,1161	0,2491	11,703
5	11	5,03E-07	0,1431	0,0837	6,61E-05	0,1431	0,0463	15,328
6	9	4,19E-07	0,812	0,0505	0,0349	0,0862	0,812	12,907
7	24	3,94E-07	4,1631	0,0291	5,06E-05	4,0223	4,1631	32,563
8	10	5,48E-07	0,2008	0,0129	0,0015	0,0195	0,2008	14,109
9	10	3,43E-08	5,5382	0,0204	0,0016	5,5382	0,0616	14,281
10	12	5,38E-07	3,2999	0,0149	7,14E-04	3,2999	2,6032	16,688

flood e *de-authentication* juntos. O arquivo de treinamento foi composto de 10000 registros normais e 4000 registros com os ataques *virtual carrier sense*, *association flood* e *de-authentication*, totalizando 14000 registros. O arquivo de teste foi composto de 4000 registros normais e 1600 registros com os ataques *virtual carrier sense*, *association flood* e *de-authentication*, totalizando 5600 registros.

Tabela 5.34: Arquivo de Treino para os Ataques 01, 02 e 03.

TIPOS DE REGISTROS	Nº DE REGISTROS	PERCENTUAL
Normal	15000	71.42%
Virtual Carrier Sense	2000	9.52%
Association Flood	2000	9.52%
De-authentication	2000	9.52%
TOTAL	21000	100%

Tabela 5.35: Arquivo de Teste para os Ataques 01, 02 e 03.

TIPOS DE REGISTROS	Nº DE REGISTROS	PERCENTUAL
Normal	6000	71.42%
Virtual Carrier Sense	800	9.52%
Association Flood	800	9.52%
De-authentication	800	9.52%
TOTAL	8400	100%

Os resultados da seqüência de treinamentos e simulações da rede neural, relativos à etapa de Generalização para os ataques *virtual carrier sense*, *association*

flood e *de-authentication* juntos, são mostrados na Tabela 5.36.

De acordo com os resultados encontrados, o maior erro de teste foi de 9.8902%, que corresponde à maior Taxa de Falsos Positivos encontrada. A maior Taxa de Falsos Negativos encontrada foi de 5.0963%. O maior dos erros médios foi de 0.093% e o maior dos erros mínimos foi de 8.4204e-004%.

Tabela 5.36: Resultados da Generalização da Rede Neural para os Ataques 01, 02 e 03.

Seq	ÉPOCAS	MSE	ERRO MÁX	ERRO MÉD	ERRO MÍN	FALSO POS	FALSO NEG	TEMPO (seg)
1	24	4,88E-07	5,3835	0,0525	3,40E-04	5,3835	1,6479	48,469
2	8	3,79E-07	2,4833	6,07E-06	3,93E-08	2,4833	1,4638	16,828
3	46	9,75E-07	7,2239	0,0196	1,73E-05	7,2239	4,9898	91,39
4	12	4,68E-07	0,3574	0,0115	1,74E-04	0,1716	0,3574	24,671
5	34	9,90E-07	7,764	0,0141	1,78E-05	7,764	2,588	67,843
6	7	6,34E-07	1,4334	0,093	8,42E-04	1,4334	0,0948	15,562
7	7	9,07E-07	9,8902	8,44E-04	6,49E-06	9,8902	0,5019	15,109
8	48	9,72E-07	5,0963	0,0245	6,69E-08	3,4064	5,0963	94,547
9	10	3,14E-07	0,3104	0,0662	1,51E-04	0,3104	0,0301	20,859
10	9	9,81E-08	0,2724	0,0032	9,25E-05	0,2724	0,0306	18,75

5.10 Análise dos Piores Casos

Nesta seção, os piores índices encontrados em todas as simulações para Erro Máximo, Taxa de Falsos Positivos e Taxa de Falsos Negativos foram concentrados em tabelas, para uma análise geral da rede neural.

De acordo com a Tabela 5.37, o ataque que apresentou um maior erro foi o ataque 01 (*virtual carrier sense*), com um erro de 8.8316%. Quando simulados dois a dois, os maiores erros apresentados ficaram entre 9 e 10%. Quando simulados os três ataques juntos, o maior erro ficou abaixo de 9.9%.

De acordo com a Tabela 5.38, o ataque que apresentou uma maior taxa de falso positivo foi o ataque 03 (*de-authentication*), com 6.3488%. Quando simulados dois a dois, as maiores taxas de falso positivo apresentadas ficaram entre 9 e 10%, sendo que em uma das combinações essa taxa ficou em 3.567%. Quando simulados os três ataques juntos, a maior taxa de falso positivo foi de 9.8902%.

De acordo com a Tabela 5.39, o ataque que apresentou uma maior taxa de falso

Tabela 5.37: Piores Casos em Relação ao Erro Máximo.

	ATAQUE						
ETAPA	01	02	03	01 e 02	01 e 03	02 e 03	01, 02 e 03
VALIDAÇÃO	3.0890	1.6236	1.9721				
INTERPOLAÇÃO	3.1588	2.9826	5.9363				
EXT INFERIOR	3.1746	4.7420	5.3272				
EXT SUPERIOR	3.9975	2.1359	6.8098				
EXT GERAL	1.1879	3.8336	6.5416				
GENERALIZAÇÃO	8.8316	1.3265	6.3488	9.5958	9.3851	9.8141	9.8902

Tabela 5.38: Piores Casos em Relação à Taxa de Falsos Positivos.

	ATAQUE						
ETAPA	01	02	03	01 e 02	01 e 03	02 e 03	01, 02 e 03
VALIDAÇÃO	3.0853	1.6236	1.9593				
INTERPOLAÇÃO	3.1588	2.9826	5.9363				
EXT INFERIOR	3.1746	4.7420	5.3272				
EXT SUPERIOR	3.9975	1.3515	6.8098				
EXT GERAL	1.1576	2.3701	6.5416				
GENERALIZAÇÃO	3.6660	1.3265	6.3488	3.5670	9.3851	9.8141	9.8902

negativo foi o ataque 03 (*de-authentication*), com 3.949%. Quando simulados dois a dois, a maior taxa de falso negativo ficou abaixo de 9.6%, sendo que em uma das combinações essa taxa ficou em 4.1631%. Quando simulados os três ataques juntos, a maior taxa de falso negativo ficou abaixo de 5.1%.

Tabela 5.39: Piores Casos em Relação à Taxa de Falsos Negativos.

	ATAQUE						
ETAPA	01	02	03	01 e 02	01 e 03	02 e 03	01, 02 e 03
VALIDAÇÃO	0.2438	0.2842	0.4287				
INTERPOLAÇÃO	1.4413	0.4544	0.0620				
EXT INFERIOR	0.7343	0.3632	0.1529				
EXT SUPERIOR	0.8706	2.1359	0.1511				
EXT GERAL	0.3620	1.6884	2.7774				
GENERALIZAÇÃO	2.7750	0.3058	3.9490	9.5958	8.4774	4.1631	5.0963

5.11 Conclusão

Este capítulo apresentou várias simulações e testes do protótipo implementado, com o objetivo de verificar a eficácia da aplicação de redes neurais na solução do

problema da detecção de intrusos de redes *wireless*. Os testes concentraram seu foco no poder de generalização das redes neurais, o que garante que o sistema detecte ataques ainda que estes apresentem características ligeiramente diferentes das já conhecidas.

Capítulo 6

Conclusões e Sugestões para Trabalhos Futuros

Este capítulo apresenta as principais contribuições deste trabalho, algumas considerações finais sobre o trabalho realizado, além de sugestões para trabalhos futuros.

6.1 Contribuições

Apesar de suas grandes vantagens, o advento das redes *wireless* traz consigo uma série de novas ameaças de segurança, cujo tratamento não pode ser realizado através das contramedidas tradicionais, aplicáveis às redes cabeadas. Um sistema de detecção de intrusos é uma ferramenta efetiva para determinar se usuários não autorizados estão tentando acessar, já acessaram, ou até mesmo comprometeram a rede de computadores. Os IDSs convencionais concentram seu foco nas camadas mais altas da pilha de protocolos do modelo OSI da ISO. Já um IDS voltado para as redes *wireless* (WIDS) concentra seus esforços na identificação de problemas nas camadas 1 e 2 do modelo OSI, que são as camadas física e de enlace (Lim, Schmoyer, Levine and Owen, 2003).

A maioria dos WIDSs existentes identificam comportamentos intrusivos apenas tomando como base a exploração de vulnerabilidades conhecidas, comumente chamadas de assinaturas de ataques. Eles analisam a atividade do sistema, observando conjuntos de eventos que sejam semelhantes a um padrão pré-determinado

que descreva uma intrusão conhecida. Com isso, apenas vulnerabilidades conhecidas são detectadas, trazendo a necessidade de que novas técnicas de intrusão sejam constantemente adicionadas ao sistema. Nesse contexto, torna-se necessária a implementação de um WIDS que possa identificar comportamentos intrusivos baseando-se também na observação de desvios do comportamento normal dos usuários, hosts ou conexões da rede. Esse comportamento normal deve se basear em dados históricos, coletados durante um longo período normal de operação.

A principal contribuição deste trabalho é a proposta uma arquitetura para um sistema de detecção de intrusos em redes *wireless* por anomalias, que tem como base a aplicação da tecnologia de redes neurais artificiais, tanto nos processos de detecção de intrusões quanto de tomada de contramedidas. O sistema pode se adaptar ao perfil de uma nova comunidade de usuários, bem como pode reconhecer ataques com características um pouco diferentes das já conhecidas pelo sistema, baseando-se apenas nos desvios de comportamento dessa nova comunidade.

No modelo proposto, a detecção de intrusões e a tomada das respectivas contramedidas são realizadas em tempo real. Isso é possível graças ao poder das redes neurais em determinar o diagnóstico da rede *wireless* e as contramedidas apropriadas de uma forma bastante rápida e eficaz. Além disso, o modelo permite a tomada de contramedidas tanto ativas quanto passivas, embora tal modelo tenha seu enfoque totalmente voltado para as contramedidas ativas, por serem muito mais efetivas na ocorrência de ataques contra a integridade do sistema, como os ataques de DoS.

Foi apresentado um modelo de integração da arquitetura proposta ao NIDIA (*Network Intrusion Detection System based on Intelligent Agents*), que é um sistema de detecção de intrusos baseado em agentes inteligentes que está sendo desenvolvido na Universidade Federal do Maranhão. Essa integração é totalmente vantajosa e viável, visto que ela proporciona capacidades de detecção de intrusos e geração de contramedidas no ambiente *wireless* ao NIDIA, que é um sistema originalmente idealizado para a segurança de redes cabeadas e servidores.

Foram implementados tanto o dispositivo sensor quanto o mecanismo de detecção de intrusões usando redes neurais. Para que esse mecanismo de detecção com redes neurais pudesse funcionar, foi gerado um arquivo de treinamento com registros de conexões normais e conexões sob ataques de DoS, onde cada conexão

foi associada a um diagnóstico de segurança da rede *wireless*.

Foram realizadas várias simulações e testes do protótipo implementado, para três ataques de negação de serviço. Os testes tiveram o objetivo de verificar a eficácia da aplicação de redes neurais na solução do problema da detecção de intrusos de redes *wireless*, concentrando seu foco no poder de generalização das redes neurais. Isto garante que o sistema detecte ataques ainda que estes apresentem características ligeiramente diferentes das já conhecidas.

6.2 Considerações Finais

Após a realização das simulações e testes, constatou-se que as Taxas de Erro Máximo ficaram abaixo de 9.9% para todas as combinações de ataques testadas. Esses resultados são bons, se comparados com resultados de pesquisas similares, na área de detecção de intrusos aplicando técnicas de redes neurais artificiais. Os resultados de três dessas pesquisas são comentados a seguir.

Uma abordagem usando redes neurais para a detecção de intrusos é apresentada em (Moradi and Zulkernine, 2004), detectando não somente a ocorrência de ataque, mas também o tipo de ataque, possibilitando ao sistema sugerir ações apropriadas. Um *perceptron* multicamadas foi utilizado para a detecção de intrusões, baseando-se em uma abordagem de análise *off-line*. Um método de validação de *early-stopping* foi aplicado na fase de treinamento para aumentar o poder de generalização das redes neurais. Quando dados de teste foram apresentados à rede neural, a taxa de acerto foi de 90.9%.

Um sistema de detecção de intrusos para redes de pacotes TCP/IP baseado em redes neurais artificiais é apresentado em (Silva, 2005). Quatro configurações de redes neurais foram treinadas para detectar anomalias e posteriormente testadas com uma base de dados que incluiu ataques novos. Taxas de acerto acima de 90% foram obtidas para todas as configurações testadas.

Um método de detecção de intrusões baseado em DGNN (*Dynamic Growing Neural Network*), que adiciona novos neurônios à rede neural em certas condições, é apresentado em (Yanheng, Tian and Li, 2006). O método de detecção é baseado em anomalias e pode aprender o comportamento normal da rede *wireless*. A entrada da rede neural foi composta por informações extraídas dos pacotes,

como endereços MAC, endereços IP e campos de controle. Resultados de testes indicaram que o sistema pode encontrar novos intrusos com uma baixa taxa de falsos alarmes, sendo que as taxas de erro ficaram em torno de 15%.

Outra constatação do nosso trabalho foi que as taxas de falsos positivos ficaram sempre acima das taxas de falsos negativos, para cada simulação. Isso evidenciou uma das principais características dos sistemas de detecção de intrusos por anomalias, que é a ocorrência de falsos positivos.

Além disso, erros de treinamento em 3.0890% para o ataque de *virtual carrier sense*, 1.6236% para o ataque *association flood* e 1.9721% para o ataque *de-authentication*, erros relativamente grandes, são uma clara indicação de que as variáveis que caracterizam cada um dos ataques devem melhorar em termos de qualidade e quantidade, de modo a possibilitar uma melhor aprendizagem por parte da rede neural.

6.3 Trabalhos Futuros

Como sugestão para trabalhos futuros, bem como para melhoria deste, temos:

- Realização de testes do protótipo implementado com outros tipos de ataques;
- Implementação do mecanismo de detecção dando como saída um conjunto de variáveis, indicando o tipo de ataque que está ocorrendo. O mecanismo de detecção implementado neste trabalho é capaz de fornecer apenas uma variável de saída, indicando se o sistema está (*1*) ou não (*0*) sob o efeito de um ataque;
- Implementação e teste do mecanismo de contramedidas constante na arquitetura proposta neste trabalho;
- Implementação dos agentes necessários para a integração ao NIDIA;
- Integração do protótipo com tecnologias de segurança existentes, como o uso de criptografia para a comunicação entre os elementos do sistema;
- Expansão da arquitetura atual, para que o sistema utilize dispositivos móveis existentes na rede como elementos sensores, dotando-os de certas capacidades de reação local contra ataques.

Referências Bibliográficas

[AirDefense, 2007] AirDefense - Enterprise Lan Security & WLAN Monitoring. (Data de Acesso: 11/2007). AirDefense, disponível em http://www.airdefense.net/.

[AirJack, 2007] SourceForge.net: AirJack. (Data de Acesso: 11/2007). AirJack, disponível em http://sourceforge.net/projects/airjack/.

[AirMagnet, 2007] AirMagnet - Enterprise Wireless Network Security and Troubleshooting. (Data de Acesso: 11/2007). AirMagnet, disponível em http://www.airmagnet.com/.

[AirSnare, 2007] AirSnare - Intrusion Detection Software for Windows. (Data de Acesso: 11/2007). AirSnare, disponível em http://home.comcast.net/ jay.deboer/airsnare/.

[AirSnarf, 2007] AirSnarf - A rogue AP setup utility. (Data de Acesso: 11/2007). AirSnarf, disponível em http://airsnarf.shmoo.com/.

[AirSnort, 2007] AirSnort HomePage. (Data de Acesso: 11/2007). AirSnort, disponível em http://airsnort.shmoo.com/.

[ANSI/IEEE, 1999] ANSI/IEEE Std 802.11, 1999 Edition (1999). Wireless LAN Medium Access Control and Physical Layer Specifications.

[Atheros, 2007] Atheros Communications. (Data de Acesso: 11/2007). Atheros Communications, disponível em http://www.atheros.com/.

[Bellardo and Savage, 2003] Bellardo, J. and Savage, S. (2003). 802.11 Denial-of-Service Attacks: Real Vulnerabilities and Practical Solutions. In *Proceedings of the USENIX Security Symposium*, Washington DC, USA, August 2003.

[Bsd-airtools, 2007] Dachb0den Labs Bsd-airtools. (Data de Acesso: 11/2007). Bsd-airtools, disponível em http://www.dachb0den.com/projects/bsd-airtools.html.

[Cansian, Grégio, Sousa e Filho, 2005] Cansian, A. M., Grégio, A. R. A., Sousa, A. Z. T. e Filho, A. M. (2005). Uma análise crítica sobre a segurança de redes sem fio na cidade de São Paulo. (Data de Acesso: 11/2007). MODULO, disponível em http://www.modulo.com.br/.

[Chauvin and Humelhart, 1995] Chauvin, Y. and Humelhart, D. E. (1995). Backpropagation: Theory, Architectures, and Applications (Developments in Connectionist Theory), Lea.

[Curran and Smyth, 2006] Curran, K. and Smyth, E. (2006). Demonstrating the Wired Equivalent Privacy (WEP) Weaknesses Inherent in Wi-Fi Networks. In *Information Systems Security*, Set/Out 2006, Vol 15 Issue 4, pp. 17-38.

[Dasgupta, Gómez, González, Kaniganti, Yallapu and Yarramsetti, 2003] Dasgupta, D., Gómez, J., González, F., Kaniganti, M., Yallapu, K. and Yarramsetti, R. (2003). MMDS: Multilevel Monitoring and Detection System. In *Proceedings of the 15th Annual Computer Security Incident Handling Conference*, Ottawa, Canada, Junho 22-27.

[DEE, 2007] Departamento de Engenharia de Eletricidade da UFMA. (Data de Acesso: 11/2007). DEE-UFMA, disponível em http://sun.dee.ufma.br/portal/.

[D-Link, 2007] D-Link. (Data de Acesso: 11/2007). D-Link, disponível em http://www.dlink.com/.

[Eclipse, 2007] Eclipse.org home. (Data de Acesso: 11/2007). Eclipse, disponível em http://www.eclipse.org/.

[EMC, 2007] EMC Corporation. (Data de Acesso: 11/2007). EMC, disponível em http://www.emc.com.

[FakeAP, 2007] Black Alchemy Projects - FakeAP. (Data de Acesso: 11/2007). FakeAP, disponível em http://www.blackalchemy.to/project/fakeap/.

[Fausset, 1994] Fausett, L.V. (1994). Fundamentals of Neural Networks, Prentice Hall.

[Fedora, 2007] Projeto Fedora. (Data de Acesso: 11/2007). Fedora, disponível em http://fedoraproject.org/.

[Fluhrer, Mantin and Shamir, 2001] Fluhrer, S., Mantin, I. and Shamir, A. (2001). Weaknesses in the Key Scheduling Algorithm of RC4. In *Eighth Annual Workshop on Selected Areas in Criptography.*

[Haykin, 1998] Haykin, S. (1998). Neural Networks: A Comprehensive Foundation, 2. Edição, Prentice Hall.

[Hegazy, Alarif, Fayed and Fahim, 2003] Hegazy, I., Alarif, T., Fayed, Z. T. and Fahim, H. M. (2003). A framework for multiagent-based system for intrusion detection. In *The Proceedings of the 3rd International Conference in Intelligent Systems Design and Applications*, August 2003.

[HostAP, 2007] Host AP driver. (Data de Acesso: 11/2007). Host AP, disponível em http://hostap.epitest.fi/.

[IEEE, 2007] IEEE - Institute of Electrical and Electronics Engineers. (Data de Acesso: 11/2007). IEEE, disponível em http://www.ieee.org/.

[IETF, 2007] IETF Home Page - The Internet Engineering Task Force. (Data de Acesso: 11/2007). IETF, disponível em http://www.ietf.org/.

[ISS, 2007] ISS - Internet Security Systems. (Data de Acesso: 11/2007). ISS, disponível em http://www.iss.net/.

[Java, 2007] Java Technology. (Data de Acesso: 11/2007). Java, disponível em http://java.sun.com/.

[Jpcap, 2007] SourceForge.net: Jpcap - Network Packet Capture Facility for Java. (Data de Acesso: 11/2007). Jpcap, disponível em http://sourceforge.net/projects/jpcap/.

[Kismet, 2007] Kismet Wireless. (Data de Acesso: 11/2007). Kismet, disponível em http://www.kismetwireless.net.

[Lackey, Roths and Goddard, 2003] Lackey, J., Roths, A. and Goddard, J. (2003). Wireless Intrusion Detection, IBM Global Services, April 2003.

[Libpcap, 2007] SourceForge.net: The Libpcap Project. (Data de Acesso: 11/2007). Libpcap, disponível em http://sourceforge.net/projects/libpcap/.

[Lim, Schmoyer, Levine and Owen, 2003] Lim, Y., Schmoyer, T., Levine, J. and Owen, H. L. (2003). Wireless Intrusion Detection and Response. In *Proceedings of the 2003 IEEE Workshop on Information Assurance*, Darmstadt, Germany, March 2003.

[Lima, 2000] Lima, C. F. L. (2000). The NIDIA Project Network Intrusion Detection System based on Intelligent Agents. In *Proceedings of Tenth Latin-Ibero-American Congress on Operations Research and Systems*, Mexico City, Mexico, pp. 212-217.

[Lima, 2001] Lima, C. F. L. (2001). Agentes Inteligentes para Detecção de Intrusos em Redes de Computadores. In *Dissertação de Mestrado. Programa de Pós-Graduação em Engenharia de Eletricidade. Universidade Federal do Maranhão*, São Luís, Maranhão, Brasil.

[MacStumbler, 2007] MacStumbler. (Data de Acesso: 11/2007). MacStumbler, disponível em http://www.macstumbler.com/.

[MADWIFI, 2007] SourceForge.net: MADWIFI - Multiband Atheros Driver for WiFi. (Data de Acesso: 11/2007). MADWIFI, disponível em http://madwifi.org/.

[MATLAB, 2007] The MathWorks - MATLAB and Simulink for Technical Computing. (Data de Acesso: 11/2007). MATLAB, disponível em http://www.mathworks.com/products/matlab/.

[Moradi and Zulkernine, 2004] Moradi, M. and Zulkernine, M. (2004). A Neural Network Based System for Intrusion Detection and Classification of Attacks. In *Proceedings of 2004 IEEE International Conference on Advances in Intelligent Systems Theory and Applications*, 6 pages, Luxembourg-Kirchberg, Luxembourg, November, 15-18.

[NetStumbler, 2007] NetStumbler.com. (Data de Acesso: 11/2007). NetStumbler, disponível em http://www.netstumbler.com/.

[NIST, 2002] NIST Special Publication 800-48 (2002). Wireless Network Security. (Data de Acesso: 11/2007). NIST 800-48, disponível em http://csrc.nist.gov/publications/nistpubs/800-94/SP800-48.pdf.

[NIST, 2007a] NIST Special Publication 800-94 (2007). Guide to Intrusion Detection and Prevention Systems (IDPS). (Data de Acesso: 11/2007). NIST 800-94, disponível em http://csrc.nist.gov/publications/nistpubs/800-94/SP800-94.pdf.

[NIST, 2007b] NIST Special Publication 800-97 (2007). Etablishing Wireless Robust Security Networks: A Guide to IEEE 802.11i. (Data de Acesso: 11/2007). NIST 800-97, disponível em http://csrc.nist.gov/publications/nistpubs/800-94/SP800-97.pdf.

[NNET, 2007] Neural Network Toolbox - MATLAB. (Data de Acesso: 11/2007). Neural Network Toolbox, disponível em http://www.mathworks.com/products/neuralnet/.

[NSA, 2005] NSA - National Security Agency (2005). Guidelines for the Development and Evaluation of IEEE 802.11 Intrusion Detection Systems (IDS). (Data de Acesso: 11/2007) GUIDE NSA, disponível em http://www.mirrors.au.wiretapped.net/security/info/reference/nsa-guides/other/guidelines-for-development-and-evaluation-of-ieee-802.11-ids-systems.pdf.

[Oliveira, 2006] Oliveira, E. J. S. (2006). Comunicação Segura e Confiável para Sistemas Multiagentes Adaptando Especificações XML. In *Dissertação de Mestrado. Programa de Pós-Graduação em Engenharia de Eletricidade. Universidade Federal do Maranhão*, São Luís, Maranhão, Brasil.

[Omnipeek, 2007] WildPackets - Omnipeek Portable Network Analyzer. (Data de Acesso: 11/2007). Omnipeek, disponível em http://www.wildpackets.com/products/omnipeek/overview.

[Pleskonjic, 2003] Pleskonjic, D. (2003). Wireless Intrusion Detection Systems (WIDS), Las Vegas, USA, december 2003.

[RedHat, 2007] Red Hat Home. (Data de Acesso: 11/2007). Red Hat, disponível em http://www.redhat.com/.

[Red-M, 2007] Red-M Home. (Data de Acesso: 11/2007). Red-M, disponível em http://www.red-m.com/.

[RSA, 2007a] RSA Security (2007). White paper: The Wireless Security Survey of New York City. RSA White Paper, 3th Edition, Jun 2007.

[RSA, 2007b] RSA Security (2007). White paper: The Wireless Security Survey of Paris. RSA White Paper, 3th Edition, Jun 2007.

[RSA, 2007c] RSA Security (2007). White paper: The Wireless Security Survey of London. RSA White Paper, 6th Edition, Jun 2007.

[RSA, 2007d] RSA - The Security Division of EMC. (Data de Acesso: 11/2007). RSA, disponível em http://www.rsa.com.

[Schiller, 2003] Schiller, J. (2003). Mobile Communications, Addison Wesley.

[Schmoyer, Lim and Owen, 2004] Schmoyer, T. R., Lim, Y. X. and Owen, H. L. (2004). Wireless Intrusion Detection and Response: A case study using the classic man-in-the-middle attack. In *IEEE Wireless Communications and Networking Conference*, Atlanta Ga., March 2004.

[Silva, 2005] Silva, R. M. (2005). Redes Neurais Artificiais Aplicadas à Detecção de Intrusos em Redes TCP/IP. In *Dissertação de Mestrado. Programa de Pós-Graduação em Engenharia Elétrica do Departamento de Engenharia Elétrica, PUC-Rio*, Rio de Janeiro, Rio de Janeiro, Brasil.

[SMAC, 2007] PCWorld Award - SMAC MAC Address Spoofer for Windows VISTA, XP, 2003, 2000. (Data de Acesso: 11/2007). SMAC, disponível em http://www.klcconsulting.net/smac/.

[Smolensky, Mozer and Humelhart, 1996] Smolensky, P., Mozer, M. C. and Humelhart, D. E. (1996). Mathematical Perspectives on Neural Networks (Developments in Connectionist Theory), Lea.

[Snort-Wireless, 2007] Snort-Wireless Project. (Data de Acesso: 11/2007). Snort-Wireless, disponível em http://www.snort-wireless.org/.

[Snort, 2007] Snort.org. (Data de Acesso: 11/2007). Snort, disponível em http://www.snort.org/.

[Staniford-Chen, 1998] Staniford-Chen, S. (1998). Common Intrusion Detection Framework(CIDF). In *Computer Emergency Response Team (Coordenation Center)*, Out 1998. Disponível em: http://seclab.cs.ucdavis.edu/cidf/.

[Surveyor, 2007] Surveyor Wireless. (Data de Acesso: 11/2007). Surveyor Wireless, disponível em http://investor.finisar.com/ReleaseDetail.cfm?ReleaseID=89597.

[THC-RUT, 2007] THC-RUT Tool. (Data de Acesso: 11/2007). THC-RUT, disponível em http://freeworld.thc.org/thc-rut/.

[UFMA, 2007] UFMA - Universidade Federal do Maranhão. (Data de Acesso: 11/2007). UFMA, disponível em http://www.ufma.br/.

[UML, 2007] UML - Unified Modeling Language. (Data de Acesso: 11/2007). UML, disponível em http://www.uml.org/.

[Wellenreiter, 2007] Wellenreiter Wireless Penetration Tool. (Data de Acesso: 11/2007). Wellenreiter, disponível em http://sourceforge.net/projects/wellenreiter/.

[WEPCrack, 2007] WEPCrack - An 802.11 key breaker. (Data de Acesso: 11/2007). WEPCrack, disponível em http://wepcrack.sourceforge.net/.

[WI-FI Alliance, 2007] Wi-Fi Alliance Home Page. (Data de Acesso: 11/2007). Wi-Fi Alliance, disponível em http://www.wi-fi.org/.

[WIGLE, 2007] WIGLE - Wireless Geographic Logging Engine. (Data de Acesso: 11/2007). WIGLE, disponível em http://www.wigle.net/.

[Wireshark, 2007] Wireshark Network Protocol Analyzer. (Data de Acesso: 11/2007). Wireshark, disponível em http://www.wireshark.org/.

[Yang, Xie and Sun, 2004] Yang, H., Xie, L. and Sun, J. (2004). Intrusion Detection Solution to WLANs. In *IEEE 6th CAS Symp. On Emerging Technologies: Mobile and Wireless Comm.*, Shanghai, China, Mai 31 - Jun 2.

[Yanheng, Tian and Li, 2006] Yanheng, L., Tian, D. and Li, B. (2006). A wireless intrusion detection method based on dynamic growing neural network. In *1st International Multi-Symposium on Computer and Computational Sciences*, Hangzhou, China.

Apêndice A

Visão Geral das Redes Neurais Artificiais

Em sua forma mais geral, uma rede neural artificial é uma máquina projetada para modelar a forma pela qual o cérebro humano executa uma tarefa particular ou uma função de interesse (Haykin, 1998). Para alcançar uma boa performance, as redes neurais empregam uma intensa interconexão de unidades simples de processamento, denominadas neurônios.

Pode-se oferecer a seguinte definição de rede neural, vista como uma máquina adaptativa: uma rede neural é um processador distribuído massivamente paralelo, constituído de unidades simples de processamento, que tem uma propensão natural a armazenar conhecimento baseado em experiências e torná-lo disponível para o uso. Ela assemelha-se ao cérebro humano em dois aspectos:

1. O conhecimento é adquirido pela rede a partir do ambiente através de um processo de aprendizagem;

2. As intensidades das conexões entre neurônios, conhecidas como pesos sinápticos, são usadas para armazenar o conhecimento adquirido.

Um neurônio artificial é uma unidade de informação-processamento que é fundamental para a operação da rede neural. No diagrama de blocos da Figura A.1 mostra-se o modelo de um neurônio, o qual forma a base para o projeto de redes neurais artificiais. Aqui, identificam-se três elementos básicos do modelo neural:

1. Um conjunto de sinapses, cada um dos quais é caracterizado por um peso. Especificamente, um sinal *x_j* na entrada da sinapse *j* conectada ao neurônio *k* é multiplicado pelo peso sináptico *W_kj*. Diferentemente de uma sinapse no cérebro humano, os pesos sinápticos de uma rede neural podem estar em uma extensão que inclui valores positivos e negativos;

2. Um somador para os sinais de entrada, ponderados pelas respectivas sinapses do neurônio;

3. Uma função de ativação para limitar a amplitude da saída do neurônio. Tipicamente, a extensão da amplitude normalizada da saída de um neurônio é escrita como o intervalo unitário fechado [0,1] ou alternativamente [-1,1].

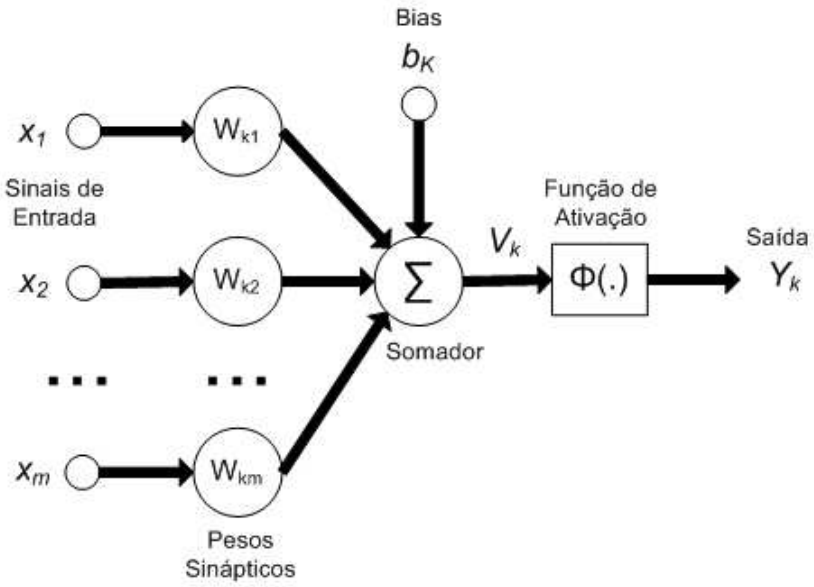

Figura A.1: Modelo de um Neurônio Artificial.

O modelo neural da Figura A.1 também inclui um *bias* ou polarizador aplicado externamente, denotado por *bk*. O polarizador *bk* tem o efeito de aumentar ou diminuir a saída da função de ativação, dependendo de ser positivo ou negativo, respectivamente (Fausset, 1994).

A propriedade mais importante de uma rede neural é a sua habilidade de aprender a partir do seu ambiente e aumentar sua performance através dessa aprendizagem. Uma rede neural aprende acerca do seu ambiente através de um

processo interativo de ajuste aplicado aos seus pesos sinápticos e polarizador. Idealmente, a rede torna-se mais inteligente acerca do seu ambiente após cada iteração do processo de aprendizagem.

Um conjunto fixo de regras bem definidas para a solução de um problema de aprendizagem é denominado algoritmo de aprendizagem. A maneira na qual os neurônios de uma rede neural estão estruturados está intimamente ligada com o algoritmo de aprendizagem usado para treiná-la. Em geral, podem ser identificadas três classes de arquiteturas de rede: rede neural de uma camada, rede neural de várias camadas e rede neural recorrente.

Em uma RNA de uma camada de neurônios, uma camada de neurônios de entrada projeta sobre uma camada de neurônios de saída, mas não vice-versa. Uma rede neural de várias camadas distingue-se pela presença de uma ou mais camadas escondidas (*hidden layers*), cujos nós de processamento são correspondentemente denominados neurônios escondidos ou unidades escondidas. Uma rede neural recorrente apresenta pelo menos uma malha de realimentação.

Tipicamente, uma rede neural multicamada consiste em um conjunto de unidades sensoriais (nós fonte), que constituem a camada de entrada, uma ou mais camadas escondidas e uma camada de saída. O sinal de entrada se propaga através da rede para adiante, camada por camada. Estas redes neurais são comumente referidas MLPs (*Multilayer Perceptrons*), os quais representam uma generalização dos perceptrons de uma única camada. O perceptron é um tipo de neurônio artificial não linear. Na Figura A.2 mostra-se o grafo arquitetural de um perceptron multicamada, com duas camadas escondidas e uma camada de saída. Em sua forma geral, a rede mostrada aqui é completamente conectada. Isso significa que um neurônio em qualquer camada da rede está conectado a todos os neurônios da camada anterior. Para aplicações não-lineares práticas é suficiente um perceptron multicamada com até duas camadas escondidas de neurônios. O número de neurônios pode ser diferente para cada camada.

RNA do tipo perceptron multicamada têm sido aplicados com sucesso para solucionar problemas em diversas áreas, por serem treinados com o algoritmo de retropropagação de erros (*backpropagation*), que é baseado na regra de aprendizagem de correção de erros. (Chauvin and Humelhart, 1995), (Smolensky, Mozer and Humelhart, 1996)

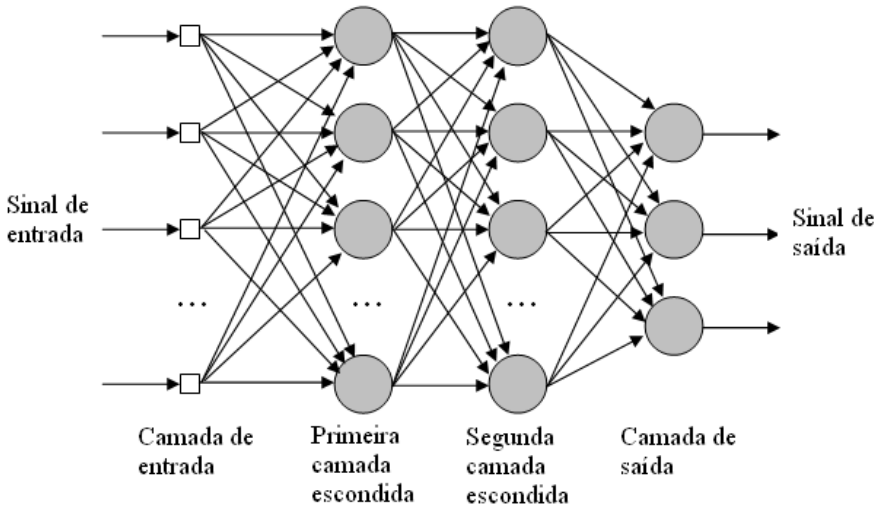

Figura A.2: Arquitetura de um Perceptron Multicamada

O algoritmo *back-propagation* consiste de duas passagens: uma passagem para adiante e uma passagem para trás. Na passagem para adiante, um padrão é aplicado aos nós sensoriais da rede e seu efeito se propaga camada por camada. Com isso, um conjunto de saídas é então produzido como a resposta da rede neural. Logo em seguida, ocorre a passagem para trás, na qual os pesos sinápticos são todos ajustados de acordo com uma regra de correção de erros. Especificamente, a resposta da rede neural é subtraída a partir de uma resposta desejada (alvo) para produzir um erro de sinal. Esse erro de sinal é então propagado para trás através da rede, em direção contrária às conexões sinápticas. Dessa maneira, os pesos sinápticos são ajustados para fazer a atual resposta da rede se mover para a resposta desejada em um sentido estatístico.

Apêndice B

Dispositivos do Ambiente de Captura

As configurações dos dispositivos que constituíram o ambiente de captura são detalhadas nas Tabelas de B.1 a B.8.

Tabela B.1: Configurações do AP.

IDENTIFICAÇÃO	AP
CATEGORIA	D-Link DWL 2100AP Wireless 108G Access Point
ENDEREÇO MAC	00-15-E9-2B-E8-9F

Tabela B.2: Configurações do PALM.

IDENTIFICAÇÃO	PALM
CATEGORIA	Palm T-X Handheld
INTERFACE WIRELESS	Wi-Fi e Bluetooth
ENDEREÇO MAC	00-0B-6C-55-1E-31
PROCESSADOR	Processador Intel 312 Mhz baseado em ARM
MEMÓRIA	128 MB
SISTEMA OPERACIONAL	Palm OS Garnet v5.4.9

Tabela B.3: Configurações do PC01.

IDENTIFICAÇÃO	PC01
CATEGORIA	Computador Pessoal
INTERFACE WIRELESS	D-Link DWL-G520 Wireless PCI Adapter
ENDEREÇO MAC	00-13-46-96-CA-7F
PLACA MÃE	Asus A8V-E SE
PROCESSADOR	AMD ATHLON 64 3700+ (2.2 GHz)
MEMÓRIA	1 GB RAM DDR
HD	80 GB
SISTEMA OPERACIONAL	Fedora Core Linux 6

Tabela B.4: Configurações do PC02.

IDENTIFICAÇÃO	PC02
CATEGORIA	Computador Pessoal
INTERFACE WIRELESS	D-Link DWL-G520 Wireless PCI Adapter
ENDEREÇO MAC	00-13-46-96-CB-91
PLACA MÃE	Asus A8V-E SE
PROCESSADOR	AMD ATHLON 64 3700+ (2.2 GHz)
MEMÓRIA	1 GB RAM DDR
HD	80 GB
SISTEMA OPERACIONAL	Microsoft Windows XP SP2

Tabela B.5: Configurações do PC03.

IDENTIFICAÇÃO	PC03
CATEGORIA	Computador Pessoal
INTERFACE WIRELESS	D-Link DWL-G520 Wireless PCI Adapter
ENDEREÇO MAC	00-13-46-96-D5-0D
PLACA MÃE	Asus A8V-E SE
PROCESSADOR	AMD ATHLON 64 3700+ (2.2 GHz)
MEMÓRIA	1 GB RAM DDR
HD	80 GB
SISTEMA OPERACIONAL	Microsoft Windows XP SP2

Tabela B.6: Configurações do NB01.

IDENTIFICAÇÃO	NB01
CATEGORIA	Notebook Sony VAIO VGN-FJ370
INTERFACE WIRELESS	Intel(R) Pro/Wireless 2200BG
ENDEREÇO MAC	00-16-6F-7A-EE-0F
PROCESSADOR	Intel Pentium M Centrino 1.86 GHz
MEMÓRIA	1 GB RAM
HD	100 GB
SISTEMA OPERACIONAL	Microsoft Windows XP SP2

Tabela B.7: Configurações do NB02.

IDENTIFICAÇÃO	NB02
CATEGORIA	Notebook Acer Aspire 3050-1458
INTERFACE WIRELESS	Broadcom 802.11g
ENDEREÇO MAC	00-19-7E-62-16-F1
PROCESSADOR	Mobile AMD Semprom 1.8 GHz
MEMÓRIA	512 MB RAM DDR2
HD	80 GB
SISTEMA OPERACIONAL	Microsoft Windows XP SP2

Tabela B.8: Configurações do NB03.

IDENTIFICAÇÃO	NB03
CATEGORIA	Notebook Compaq Presario 2108
INTERFACE WIRELESS	Broadcom 802.11b/g WLAN
ENDEREÇO MAC	00-90-4B-95-42-90
PROCESSADOR	AMD Athlon XP-M 2.12 GHz
MEMÓRIA	512 MB DDR
HD	100 GB
SISTEMA OPERACIONAL	Microsoft Windows XP SP2