UNIVERSIDADE FEDERAL DE SANTA MARIA
CENTRO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

DETECÇÃO DE ATAQUES DE NEGAÇÃO DE SERVIÇO EM REDES DE COMPUTADORES ATRAVÉS DA TRANSFORMADA WAVELET 2D

DISSERTAÇÃO DE MESTRADO

Renato Preigschadt de Azevedo

Santa Maria, RS, Brasil

2012

DETECÇÃO DE ATAQUES DE NEGAÇÃO DE SERVIÇO EM REDES DE COMPUTADORES ATRAVÉS DA TRANSFORMADA WAVELET 2D

por

Renato Preigschadt de Azevedo

Dissertação apresentada ao Programa de Pós-Graduação em Informática da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para a obtenção do grau de

Mestre em Computação

Orientador: Prof. Dr. Raul Ceretta Nunes (UFSM)

Co-orientadora: Profª. Drª. Alice de Jesus Kozakevicius (UFSM)

Santa Maria, RS, Brasil

2012

A994d Azevedo, Renato Preigschadt de
Detecção de ataques de negação de serviço em redes de computadores através da transformada Wavelet 2D / Renato Preigschadt de Azevedo. – 2012.
97 f. : il. ; 30 cm

Orientador: Raul Ceretta Nunes.
Coorientadora: Alice de Jesus Kozakevicius.
Dissertação (mestrado) – Universidade Federal de Santa Maria, Centro de Tecnologia, Programa de Pós-Graduação em Informática, RS, 2012

1. Informática 2. Rede de computadores 3. Detecção de anomalias 4. Detecção de intrusão 5. Wavelet 2D 6. Ataques DoS 7. Sistemas distribuídos I. Nunes, Raul Ceretta II. Kozakevicius, Alice de Jesus III. Título.

CDU 004.42
004.492.3

Ficha catalográfica elaborada por Simone G. Maisonave – CRB 10/1733
Biblioteca Central da UFSM

Universidade Federal de Santa Maria
Centro de Tecnologia
Programa de Pós-Graduação em Informática

A Comissão Examinadora, abaixo assinada,
aprova a Dissertação de Mestrado

DETECÇÃO DE ATAQUES DE NEGAÇÃO DE SERVIÇO EM REDES DE COMPUTADORES ATRAVÉS DA TRANSFORMADA WAVELET 2D

elaborada por
Renato Preigschadt de Azevedo

como requisito parcial para obtenção do grau de
Mestre em Computação

COMISSÃO EXAMINADORA:

Raul Ceretta Nunes (UFSM), Dr.
(Presidente/Orientador)

Luciano Paschoal Gaspary, Prof. Dr. (UFRGS)

Renato Machado, Prof. Dr. (UFSM)

Santa Maria, 8 de Março de 2012.

DEDICATÓRIA

Dedico este trabalho a meu irmão Ricardo Lima de Azevedo Junior por ser um exemplo de perserverança e luta.

AGRADECIMENTOS

Ao Programa de Pós-Graduação em Informática (PPGI) da Universidade Federal de Santa Maria por propiciar o acesso a pós-graduação.

Ao professor Dr. Raul Ceretta Nunes, e à co-orientadora professora Dra. Alice de Jesus Kozakevicius pelo apoio, orientações e esforço em garantir as discussões para a realização deste trabalho.

Aos integrantes dos grupos de pesquisa GTSeg e Gmicro pelo apoio. Aos colegas Tiago Perlin, Bruno Mozzaquatro, Cristian Cappo, Douglas Foster pelo companheirismo e discussões sobre ataques e transformada Wavelet.

Aos amigos Jaziel Lobo, Giovani Librelotto, amigos da confraria quartagelada, amigos do Gmicro/GTSeg pelas atividades científicas e sociais.

Aos amigos externos a academia Daniel Darol, Márcio Bolzan, pelo companheirismo, e amizade. À Auriane Camillo por me manter saudável física e mentalmente.

A Capes por me propiciar bolsa de estudos durante parte do período do mestrado.

Em especial a meus pais: Ricardo Azevedo e Maria Augusta, meu irmão Ricardo Junior, e meu afilhado Bruno Azevedo pelo apoio, e por entenderem o período de ausência necessário para a realização desta dissertação.

“Eu não sei pra onde eu to indo,
mas sei que estou no meu caminho!”
— Raul Seixas

RESUMO

Dissertação de Mestrado
Programa de Pós-Graduação em Informática
Universidade Federal de Santa Maria

DETECÇÃO DE ATAQUES DE NEGAÇÃO DE SERVIÇO EM REDES DE COMPUTADORES ATRAVÉS DA TRANSFORMADA WAVELET 2D
AUTOR: RENATO PREIGSCHADT DE AZEVEDO
ORIENTADOR: RAUL CERETTA NUNES (UFSM)
CO-ORIENTADORA: ALICE DE JESUS KOZAKEVICIUS (UFSM)
Local da Defesa e Data: Santa Maria, 8 de Março de 2012.

A análise de tráfego de rede é uma área fundamental no gerenciamento de sistemas tolerantes a falhas, pois anomalias no tráfego de rede podem afetar a disponibilidade e a qualidade do serviço (QoS). Sistemas detectores de intrusão em redes de computadores são utilizados para analisar o tráfego de rede com o objetivo de detectar ataques ou anomalias. A análise baseada em anomalias permite detectar ataques através da análise do comportamento do tráfego de rede. Este trabalho propõe uma ferramenta de detecção de intrusão rápida e eficaz para detectar anomalias em redes de computadores geradas por ataques de negação de serviço (DoS). O algoritmo de detecção é baseado na transformada Wavelet bidimensional (Wavelet 2D), um método derivado da análise de sinais. A transformada wavelet é uma ferramenta matemática de baixo custo computacional, que explora as informações presentes nas amostras de entrada ao longo dos diversos níveis da transformação. O algoritmo proposto detecta anomalias diretamente nos coeficientes wavelets através de técnicas de corte, não necessitando da reconstrução do sinal original. Foram realizados experimentos utilizando duas bases de dados: uma sintética (DARPA), e outra coletada na instituição de ensino (UFSM), permitindo a análise da ferramenta de detecção de intrusão sob diferentes cenários. As famílias wavelets utilizadas nos testes foram as wavelets ortonormais de *Daubechies*: *Haar* (Db1), Db2, Db4 e Db8 (com 1, 2, 4 e 8 momentos nulos respectivamente). Para a base de dados DARPA obteve-se uma taxa de detecção de ataques DoS de até 100% utilizando a wavelet de *Daubechies* Db4 com os coeficientes wavelets normalizados, e de 95% para a base de dados da UFSM com a wavelet de *Daubechies* Db4 com os coeficientes wavelets normalizados.

Palavras-chave: Detecção de Anomalias; Detecção de Intrusão; Wavelet; Wavelet 2D; Sistemas Distribuídos.

ABSTRACT

Master's Dissertation
Computer Science Graduate Program
Federal University of Santa Maria

A BIDIMENSIONAL WAVELET TRANSFORM BASED ALGORITHM FOR DOS ATTACK DETECTION

AUTHOR: RENATO PREIGSCHADT DE AZEVEDO
ADVISOR: RAUL CERETTA NUNES (UFSM)
COADVISOR: ALICE DE JESUS KOZAKEVICIUS (UFSM)
Presentation Place and Date: Santa Maria, March 8th, 2012.

The analysis of network traffic is a key area for the management of fault-tolerant systems, since anomalies in network traffic can affect the availability and quality of service (QoS). Intrusion detection systems in computer networks are used to analyze network traffic in order to detect attacks and anomalies. The analysis based on anomalies allows attacks detection by analyzing the behavior of the traffic network. This work proposes an intrusion detection tool to quickly and effectively detect anomalies in computer networks generated by denial of service (DoS). The detection algorithm is based on the two-dimensional wavelet transform (2D Wavelet), a derived method of signal analysis. The wavelet transform is a mathematical tool with low computational cost that explores the existing information present in the input samples according to the different levels of the transformation. The proposed algorithm detects anomalies directly based on the wavelet coefficients, considering threshold techniques. This operation does not require the reconstruction of the original signal. Experiments were performed using two databases: a synthetic (DARPA) and another one from data collected at the Federal University of Santa Maria (UFSM), allowing analysis of the intrusion detection tool under different scenarios. The wavelets considered for the tests were all from the orthonormal family of Daubechies: Haar (Db1), Db2, Db4 and Db8 (with 1, 2, 4 and 8 null vanishing moments respectively). For the DARPA database we obtained a detection rate up to 100% using the Daubechies wavelet transform Db4, considering normalized wavelet coefficients. For the database collected at UFSM the detection rate was 95%, again considering Db4 wavelet transform with normalized wavelet coefficients.

Keywords: Anomaly Detection; Intrusion Detection; Wavelet; 2D Wavelet; Distributed Systems.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

ARX	*Auto Regressive with eXogenous input*
DARPA	*Defense Advanced Research Projects Agency*
DoS	*Denial of Service*
DDoS	*Distributed Denial of Service*
FN	Falso Negativo
FP	Falso Positivo
GTSeg	Grupo de pesquisa em Gestão e Tecnologia em Segurança da Informação
HIDS	*Host Intrusion Detection System*
HTTP	*Hypertext Transmission Protocol*
ICMP	*Internet Control Message Protocol*
IDS	*Intrusion Detection System*
IIS	*Internet Information Services*
IP	*Internet Protocol*
MIT	*Massachussetts Institute of Technology*
NIDS	*Network Intrusion Detection Service*
P2P	*Peer to peer*
PoD	*Ping of Death*
RMI	*Remote Method Invocation*
SIV	*System Integrity Verifier*
SMTP	*Simple Mail Transfer Protocol*
SNMP	*Simple Network Management Procotol*
TFN	*Tribe Flood Network*
TFN2K	*Tribe Flood Network 2000*
TD	*Taxa de Detecção*
TCP	*Transmission Control Protocol*
UDP	*User Datagram Protocol*
UML	*Unified Modeling Language*
UFSM	Universidade Federal de Santa Maria
VN	Verdadeiro Negativo
VP	Verdadeiro Positivo

SUMÁRIO

1 INTRODUÇÃO

O uso de sistemas web está crescendo a cada dia, tornando necessário prover acesso a redes com alta disponibilidade e qualidade. Um sistema de Detecção de Intrusão em Rede (NIDS) é um conjunto de ferramentas de software que permite a análise e detecção de intrusões em redes de dados. Por causa do aumento do tráfego de rede e ausência de uma distribuição probabilística conhecida, a construção de um NIDS eficiente é um desafio (WANG, 2009).

De acordo com Barford et al. (2002), redes de computadores sem análise de tráfego não podem operar eficientemente ou com segurança. A análise de tráfego é uma atividade essencial para o correto funcionamento de redes. Dentre os ataques em redes de computadores, o ataque de negação de serviço (DoS) é o que ocasiona uma maior perturbação da qualidade de serviço da rede (BADISHI; KEIDAR; SASSON, 2006).

Em uma pesquisa efetuada junto a administradores de redes de diversas organizações dos EUA, o número de ataques de DoS notificados aumentou de 21% em 2008 para 29,1% em 2009 (CSI/FBI, 2009), perante todos os tipos de ataques no período. Em 2010, o número de ataques DoS notificados diminuiu para 17%, mas continua sendo um dos tipos de ataques mais utilizados (CSI/FBI, 2011). No Brasil ocorreram diversos ataques DoS em junho de 2011 destinados a servidores do governo brasileiro (SERPRO, 2011), tornando serviços indisponíveis durante horas.

Os NIDS são usualmente classificados em dois tipos: baseados em regras, e baseados em anomalias. Os NIDS baseados em regras identificam ataques através da análise de assinaturas definidas com base no conhecimento prévio, e baseadas em conhecimento do padrão do ataque (ROESCH, 1999). Por outro lado, os NIDS baseados em anomalias analisam o comportamento do tráfego de rede e identificam perturbações no comportamento padrão, gerando um alarme a cada anomalia devido a um evento que não segue o padrão esperado (CHANDOLA; BANERJEE; KUMAR, 2009).

Considerando os padrões de ataques conhecidos, o número de falsos positivos é menor nos sistemas baseados em regras do que nos baseados em anomalias (GARCÍA-TEODORO et al., 2009). Entretanto, os NIDS baseados em regras não são adequados para detectar variações de ataques que não pertencem ao conjunto de regras pré-definidas. Os sistemas baseados em anomalias são construídos para adaptar-se a mudanças no comportamento da rede, detectando novos tipos de ataques e variâncias de ataques conhecidos (PATCHA; PARK, 2007). A aborda-

gem proposta nesta dissertação concentra-se em NIDS baseado em anomalias.

As técnicas tradicionais utilizadas em NIDS baseado em anomalias incluem: análise estatística (SAMAAN; KARMOUCH, 2008) (SCHERRER et al., 2007) (OHSITA; ATA; MURATA, 2004) e métodos baseados em aprendizagem de máquina (AHMED; ORESHKIN; COATES, 2007) (ZHANG; REXFORD; FEIGENBAUM, 2005). Uma alternativa para NIDS baseado em anomalia é o uso de técnicas de processamento de sinais, que estão sendo utilizadas com sucesso para detectar anomalias no tráfego de rede, devido a sua habilidade de detectar variações (PATCHA; PARK, 2007).

Entre as técnicas de processamento de sinais, a transformada Wavelet (MALLAT, 1998) têm sido utilizada para detectar anomalias em tráfego de rede (LU; TAVALLAEE; GHORBANI, 2008) (BARFORD et al., 2002) (HUANG; THAREJA; SHIN, 2006), (LI; LI, 2009) (DAINOTTI; PESCAPE; VENTRE, 2006), onde um dos focos é especificamente em ataques DoS (LI; LI, 2009) (DAINOTTI; PESCAPE; VENTRE, 2006) (DALMAZO et al., 2009). A transformada Wavelet permite a seleção de características do sinal através da representação combinada de tempo-escala (NIELSEN, 1998). Esta característica permite a análise de tráfego de rede considerando o volume e o tempo dos descritores de tráfego.

Neste trabalho é proposto um algoritmo para detecção de ataques de negação de serviço baseado em uma técnica de processamento de sinais. A transformada Wavelet 2D é utilizada para a análise do tráfego de rede em busca de anomalias geradas por ataques DoS. Os ataques são detectados diretamente nos coeficientes wavelets, desconsiderando a etapa de reconstrução da transformada wavelet.

Para avaliar a capacidade do algoritmo proposto de detectar ataques DoS em diferentes cenários foram efetuados experimentos utilizando duas bases de dados distintas: DARPA (DARPA, 1999) e UFSM. Foram efetuados testes com a transformada wavelet de *Haar* e *Daubechies* Db2, Db4 e Db8. Foram também modificados os parâmetros do algoritmo proposto nesta dissertação como: tamanho da janela deslizante, normalização dos coeficientes wavelets e estratégias de corte recursiva e adaptativa.

Foi obtida uma taxa de detecção de ataques DoS de até 100% na base de dados DARPA utilizando a wavelet de *Daubechies* Db4 com os coeficientes wavelets normalizados e uma janela deslizante de 256 amostras, e de até 95% para a base de dados da UFSM com a mesma configuração da DARPA.

1.1 Objetivos

Diferentemente da maioria dos trabalhos, que utilizam a transformada Wavelet (unidimensional), este trabalho propõe um algoritmo baseado na transformada Wavelet discreta bidimensional para análise de anomalias contidas no tráfego de redes. O algoritmo visa detectar ataques do tipo DoS em redes de computadores, num pequeno intervalo de tempo e sem necessidade de uma fase de treinamento, diferentemente de métodos estatísticos ou de aprendizagem de máquina.

Os principais objetivos desta dissertação são:

- Analisar o comportamento de ataques de negação de serviço.
- Projetar um algoritmo para detecção de anomalias baseado em técnicas de processamento de sinais.
- Detectar anomalias através de Wavelets 2D.
- Validar o algoritmo de detecção de anomalias através de testes efetuados em uma base de dados sintética, e uma com ataques injetados.

1.2 Contribuições

Este trabalho contribui com uma solução do problema de detecção de ataques de negação de serviço através de métodos baseados em anomalias apresentando uma solução baseada em *Wavelet* 2D. A solução proposta aplica a transformada Wavelet 2D em diferentes descritores do tráfego de rede possibilitando obter um relacionamento entre os descritores, não necessitando de algoritmos adicionais para obter o relacionamento entre os descritores. Isto permite uma análise dentre os diferentes protocolos de rede, detectando ataques que não seriam visíveis somente em um protocolo de rede. Os ataques são detectados diretamente nos coeficientes wavelets, desconsiderando a etapa de reconstrução da transformada wavelet.

1.3 Organização do texto

O trabalho está organizado da seguinte forma: no Capítulo 2 são abordados aspectos dos ataques de negação de serviços, comportamento do tráfego de rede, e sistemas detectores de intrusão; no Capítulo 3 alguns aspectos importantes da teoria Wavelet são mostrados, dando ênfase para a transformada Wavelet 2D; no Capítulo 4 o algoritmo proposto e o sistema para

detecção de intrusão são descritos. Os Experimentos e a discussão sobre os testes são apresentados no Capítulo 5, com a análise da ferramenta de detecção de intrusão sob três diferentes cenários. Foi utilizada a família wavelet ortonormal de *Daubechies*: *Haar*, *Daubechies* Db2, Db4 e Db8. No Capítulo 6 as conclusões e trabalhos futuros são apresentados.

2 ATAQUES DE NEGAÇÃO DE SERVIÇO (DOS)

A dependência das informações armazenadas geograficamente em diversos locais sobre a internet, está tornando necessário prover acesso à rede de computadores com maior disponibilidade e qualidade. Sistemas para detecção e prevenção de ataques em redes são ferramentas que possibilitam a análise do tráfego de rede. De acordo com (BARFORD et al., 2002) redes de computadores sem análise de tráfego não operam com eficiência e segurança, apontando a análise de tráfego como uma área fundamental para o correto funcionamento de redes.

Este capítulo retrata o problema dos ataques de negação de serviços disponíveis em redes de computadores, classificando-os e apontando trabalhos que procuram contê-los, e está organizado da seguinte forma: na Seção 2.1 são apresentados ataques de negação de serviço; a Seção 2.2 apresenta uma discussão sobre o comportamento do tráfego de rede; e na Seção 2.4 são apresentadas as principais considerações deste capítulo.

2.1 Ataques de Negação de Serviço

Ataques em redes de computadores são ações originadas por usuários maliciosos que tem por objetivo perturbar o comportamento e a disponibilidade da rede. Existem diversos tipos de ataques, como por exemplo: ataques de força bruta, ataques para obtenção de permissões, ataques de negação de serviço, *spoofing*, entre outros.

Este trabalho é focado na detecção de ataques de negação de serviço. Os ataques DoS são classificados em duas categorias, como pode ser observado na Figura 2.1 (KUMAR; SELVAKUMAR, 2009): esgotamento de largura de banda e esgotamento de recursos.

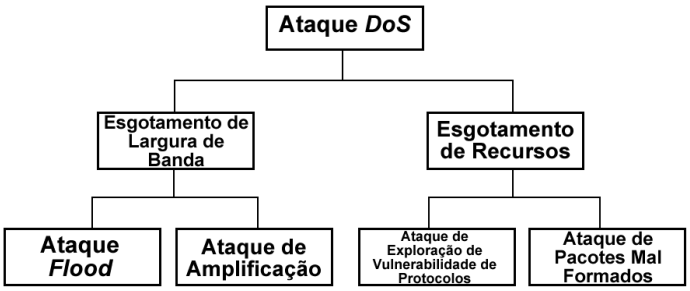

Figura 2.1: Modelo de ataque DoS proposto em (KUMAR; SELVAKUMAR, 2009).

Ataques DoS do tipo de esgotamento de largura de banda são provocados por atacantes que inundam o *host* destino através de requisições inválidas, com o objetivo de não permitir a comunicação de requisições normais (KUMAR; SELVAKUMAR, 2009). Existem diversos

tipos de ataques de esgotamento de largura de banda como, por exemplo, *flood* UDP (POSTEL, 1980) e ICMP (POSTEL, 1981), e ataques de amplificação como, *Smurf* (SPECHT, 2004) e *Fraggle* (SPECHT, 2004). Ataques DoS do tipo *flood* inundam o *host* a ser atacado com mais pacotes que o *host* consegue lidar, ocasionando na negação de serviço das aplicações legítimas que funcionam no *host* atacado. Ataques DoS de amplificação multiplicam o número de pacotes enviados ao *host* a ser atacado utilizando-se de *hosts* legítimos que possuam falhas de segurança, tornando o ataque mais efetivo.

Ataques DoS da categoria de esgotamento de recursos procuram explorar vulnerabilidades em protocolos e serviços ocasionando a negação de serviço (KUMAR; SELVAKUMAR, 2009). Nesta categoria de ataque DoS, são enviados para o *host* a ser atacado pacotes que explorem alguma vulnerabilidade da pilha TCP/IP ou de algum serviço específico, como por exemplo: TCP-SYN (EDDY, 2007), PUSH ACK (SPECHT, 2004), apache2 (APACHE, 2011), IIS (MICROSOFT, 2011). O ataque que explora a vulnerabilidade TCP-SYN envia diversas solicitações de conexão ao *host*, preenchendo totalmente a estrutura responsável por aceitar novas conexões do protocolo TCP, ocasionando em uma negação de serviço e não permitindo assim novas requisições ao *host*. Este ataque utiliza uma vulnerabilidade do projeto do protocolo TCP. O ataque PUSH ACK também explora uma vulnerabilidade do protocolo TCP. Ataques apache2 e IIS exploram vulnerabilidades existentes em serviços disponibilizados no *host*, como um servidor HTTP, ocasionando na negação de serviço destes serviços, indisponibilizando o acesso a um *website* ou *webservice* que está hospedado no *host* atacado.

Segundo PROLEXIC (2011) ataques DoS e DDoS (*Distributed Denial of Service*) são responsáveis por picos de até alguns milhões de pacotes por segundo, sobrecarregando os equipamentos de rede. O tempo médio de duração de um ataque é de 1,5 dias, e 1,5 Gb/s de velocidade média.

Neste trabalho foram considerados ataques de negação de serviço que provocam perturbações no tráfego de rede, e podem ser identificados analisando somente o cabeçalho dos pacotes. Os seguintes ataques são considerados:

- MailBomb: são enviadas milhares de requisições a um servidor de email, ocasionando a negação do serviço aos usuários legítimos.
- Back: são enviadas requisições a um servidor web Apache com diversas barras no endereço do url, tornando o tempo de resposta lento, e ocasionando posterior negação de serviço do servidor web.

- PoD (*Ping of Death*): é enviado um número elevado de requisições ICMP do tipo *ping* malformados ocasionando em negação do serviço no *host* atacado.

- Neptune: são enviadas diversas requisições do tipo TCP-SYN, inundando o *host* atacado, não permitindo ao mesmo a resposta a requisições válidas.

No ataque do tipo *MailBomb* um *host* comprometido envia um elevado número de mensagens a um servidor de email através da porta SMTP (*Simple Mail Transfer Protocol*) (HUANG; THAREJA; SHIN, 2006). Este ataque compromete os recursos do servidor de email, tornando-o indisponível para usuários legítimos, ou ainda, comprometendo a conta de algum usuário com milhares de requisições. A Figura 2.2 mostra o gráfico do descritor de rede IP, TCP, UDP e ICMP contendo um ataque do tipo *MailBomb*. No eixo Y estão descritos os pacotes (amostrados em segundos) e no eixo X as amostras (768 amostras). O ataque está delimitado pelo retângulo pontilhado em azul, exibindo o momento do início até o fim do ataque, e está representado no gráfico por aproximadamente 10 minutos (600 segundos) de duração, e pode ser caracterizado por apresentar uma elevação no número de pacotes trafegados. Apesar do ataque utilizar-se do protocolo TCP para o envio de emails, o ataque também se caracteriza pelo aumento elevado no protocolo UDP, pois o servidor de email ao enviar os emails necessita utilizar o serviço de resolução de nomes (DNS - *Domain Name Service*) que utiliza o protocolo UDP. Conforme o gráfico da Figura 2.2 demonstrada, existe um relacionamento entre diversos protocolos de rede durante o ataque, pois ocorre o aumento de pacotes nos seguintes protocolos: IP, TCP e ICMP.

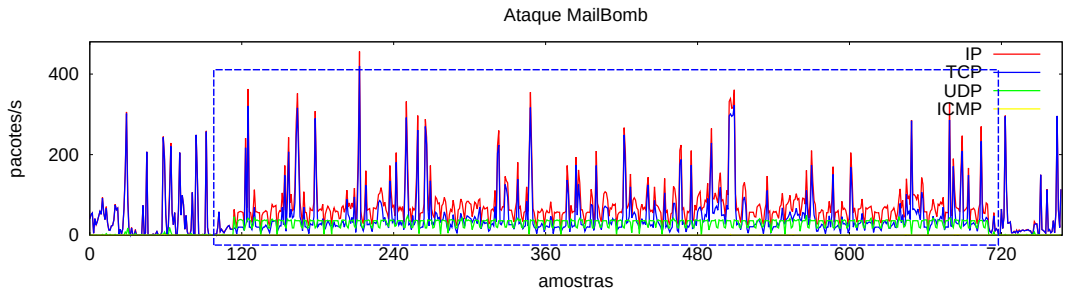

Figura 2.2: Ataque Mailbomb na base de dados DARPA (DARPA, 1999).

O ataque *back* envia diversas requisições a um servidor Web Apache, contendo muitas barras no caminho da URL, dificultando o processo de localização do documento pelo servidor web. Em alguns ataques o número de barras pode chegar a 100 (DARPA, 1999). Na Figura 2.3 é exibido o gráfico do protocolo IP, TCP, UDP e ICMP durante um ataque *Back*, que está delimitado pelo retângulo pontilhado em azul, durando aproximadamente 8 segundos.

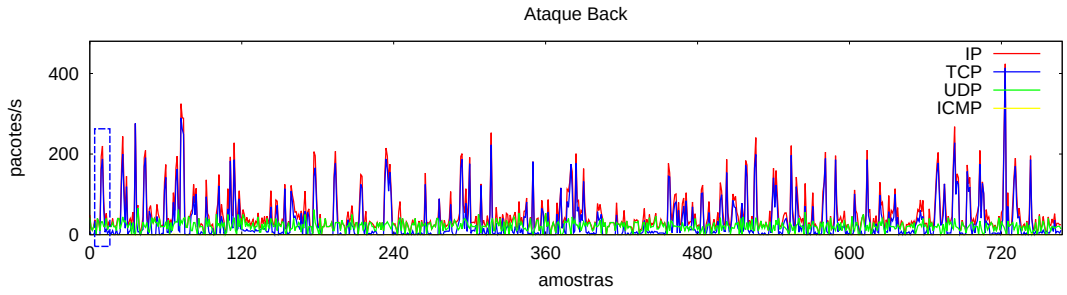

Figura 2.3: Ataque Back na base de dados DARPA (DARPA, 1999).

No ataque PoD (*ping of death*) (DARPA, 1999) são enviadas requisições ICMP do tipo *ping* com um tamanho de pacote maior que 65.535 *bytes* para um *host* a ser atacado. O gráfico na Figura 2.4 mostra 768 amostras do protocolo IP, TCP, UDP e ICMP durante a ocorrência de um ataque PoD na base de dados DARPA. Conforme mostra o retângulo pontilhado durante o ataque, aumenta o número de requisições de forma abrupta por aproximadamente 1 segundo.

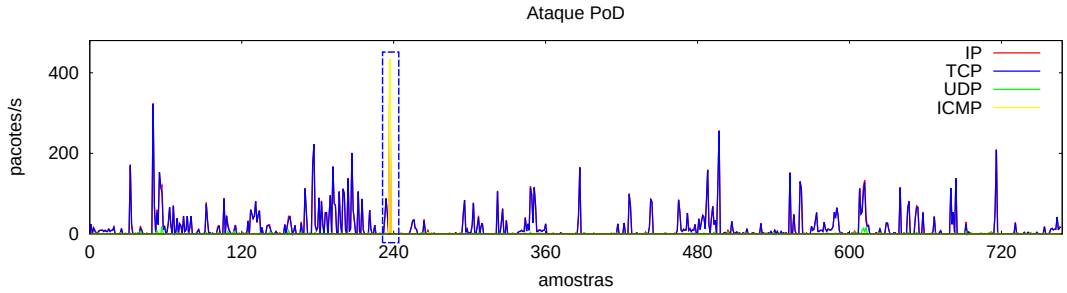

Figura 2.4: Ataque PoD na base de dados DARPA (DARPA, 1999).

O ataque *Neptune* envia diversas requisições do tipo TCP-SYN, explorando vulnerabilidades do protocolo TCP/IP. Quando um *host* de origem deseja se comunicar com outro *host* de destino através de conexão TCP, ele envia uma requisição do tipo TCP-SYN para estabelecer uma nova conexão. O *host* de destino armazena essa requisição em uma tabela, para enviar um ACK (aceite) ou RST dessa nova conexão ao requerente da conexão. No ataque *neptune* (também conhecido como SYN *flood*) o atacante um número elevado de conexão SYN ao *host* atacado com um endereço inválido de retorno, preenchendo a tabela de conexões do *host* destino. Quando a tabela de conexões de um *host* enche, este descarta novas requisições de conexão, ocasionando em negação de serviço. No gráfico da Figura 2.5 é apresentado o descritor de rede IP, TCP, UDP e ICMP da base de dados DARPA, durante a ocorrência de um ataque do tipo *neptune*. O ataque está destacado pelo retângulo pontilhado em azul. A duração do ataque é de aproximadamente 4 minutos (200 segundos). Novamente existe uma relação entre os diversos

protocolos de rede, pois ocorre um aumento dos protocolos IP e TCP, enquanto que ocorre uma diminuição nos outros protocolos em consequência do *host* não aceitar novas requisições (por causa da tabela de novas requisições estar cheia).

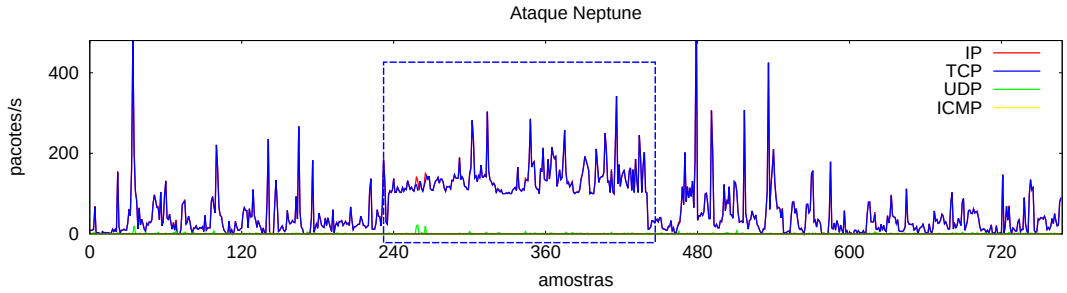

Figura 2.5: Ataque neptune na base de dados DARPA (DARPA, 1999).

Os ataques descritos nesta seção são efetuados em diferentes protocolos, sendo evidente cada ataque em algum protocolo específico, como por exemplo o ataque PoD é melhor visualizado no protocolo ICMP, o ataque *MailBomb* é melhor visualizado no TCP, assim como o ataque *Neptune*. Estas particularidades são importantes para detectar de forma eficiente os ataques, assim como a análise do relacionamento existente entre os diferentes protocolos.

2.2 Comportamento do Tráfego de Rede

O tráfego de rede é composto por uma sequência de pacotes (mensagens) que são trocados entre dispositivos conectados a rede. O tráfego de rede pode ser definido como normal ou anômalo:

- Normal: tráfego legítimo, ou seja, não existe a ocorrência de ataques, ou anomalias;
- Anômalo: presença de ataques ou anomalias como interrupção de segmentos da rede.

O comportamento do tráfego de redes é irregular, variando a intensidade e forma durante o decorrer do tempo. Conforme descrito em (LELAND et al., 1994), (FARRAPOSO; OWEZARSKI; MONTEIRO, 2005) o tráfego de rede possuí diversas correlações temporais como: dependências de longa duração (ERRAMILLI; NARAYAN; WILLINGER, 1996) e autossimilaridade (KIHONG; WALTER, 2000). O conceito de autossimilaridade é explicado graficamente na Figura 2.6.

A Figura 2.6 apresenta de forma gráfica a existência de autossimilaridade (coluna da esquerda) no tráfego de rede capturado, em diferentes escalas de tempo (variando de 10 a 0,01

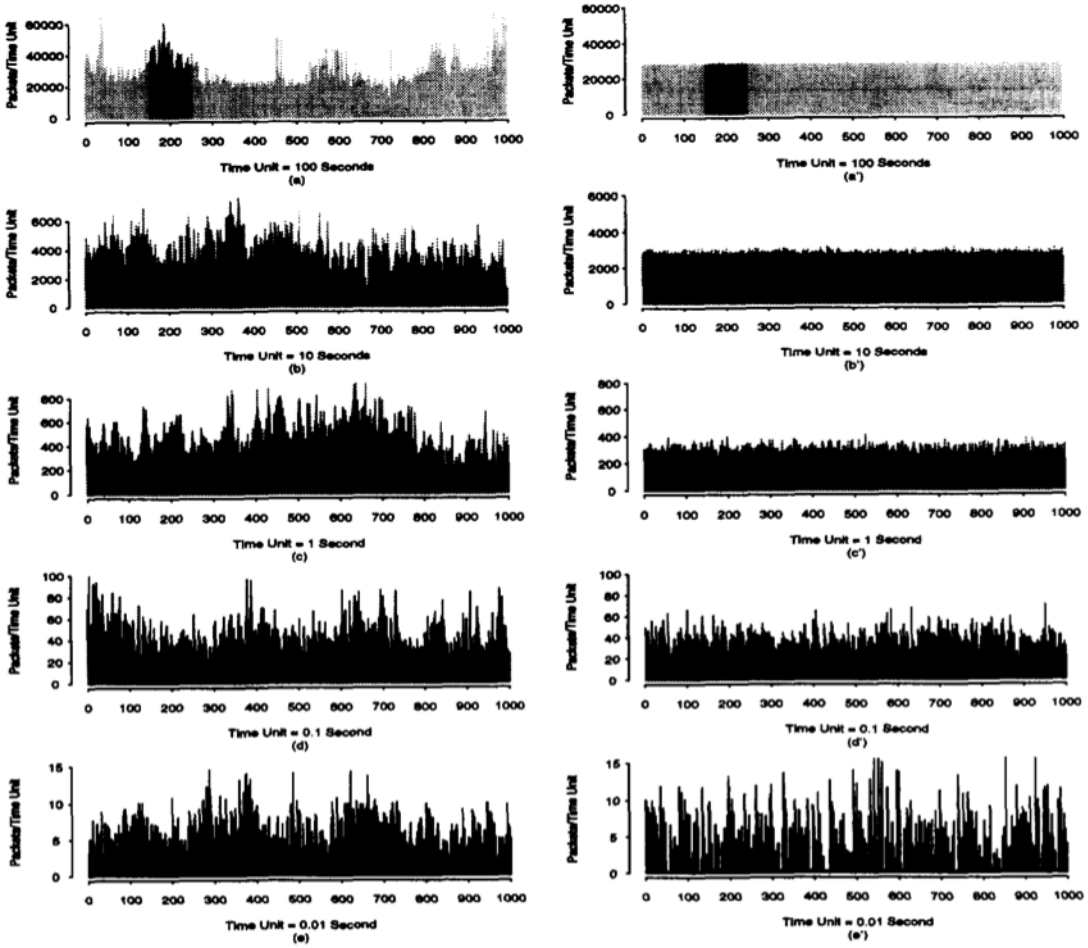

Figura 2.6: Representação gráfica da autossimilaridade do tráfego de rede. O tráfego de rede coletado está representado nos gráficos da esquerda, e possuí uma forma semelhante independente da escala de tempo, e os gráficos da direita representam o tráfego sintético. Fonte: (LELAND et al., 1994).

segundos). Na coluna da direita onde está representado um tráfego de rede sintético, gerado pelo modelo de *Poisson*, não ocorre a autossimilaridade. Pode-se inferir também que quanto maior a escala de tempo, menor a variabilidade do tráfego de rede. Ainda na Figura 2.6 nota-se o comportamento aleatório do tamanho de *burst* da amostra de dados real (coluna da esquerda).

Como o comportamento do tráfego de rede é não estacionário (variante no tempo) (FARRAPOSO; OWEZARSKI; MONTEIRO, 2005), é uma tarefa complexa modelar sintéticamente o comportamento de uma rede de computadores.

2.2.1 Descritores do Tráfego de Rede

Existem diversos protocolos para a troca de mensagens em redes de computadores. Neste trabalho são levados em consideração os seguintes protocolos: IP, TCP, UDP, ICMP, e suas subdivisões: cwr, ece, urg, ack, psh, rst, syn, fin.

Os descritores podem ser divididos em básicos e derivados (ONUT; GHORBANI, 2006). Descritores básicos são características que representam diretamente os pacotes dos protocolos de rede. Como exemplo temos: contadores de fluxo TCP, UDP, ICMP, etc. Para a extração dos descritores básicos não faz-se necessário o uso de processamento adicional após a captura dos pacotes. Todos os dados obtidos diretamente no cabeçalho dos pacotes de rede são definidos como básicos. Descritores derivados representam um conjunto de conexões agrupadas de alguma forma. Também podem ser chamadas de características do tráfego. São exemplos desta categoria: fluxos TCP por intervalo de tempo, conexões a porta "x" por intervalo de tempo, etc. É necessário manter estruturas de dados adicionais para poder calcular os descritores derivados, ocasionando o uso adicional de processamento.

2.3 Sistemas Detectores de Intrusão em Redes de Computadores

Sistemas detectores de intrusão em redes de computadores são utilizados para permitir a monitoração do tráfego de dados de uma rede de computadores, ou um segmento de rede. Um sistema de detecção de intrusão (IDS) é uma ferramenta utilizada para detectar intrusões em sistemas computacionais (KIZZA, 2005). A análise é realizada em dados coletados da rede, ou em base de dados disponíveis ao IDS.

Um ataque (ou intrusão) é uma tentativa de modificação, manipulação, ou perturbação no funcionamento de um sistema, bem sucedida ou não (WANG, 2009). Um IDS possuí normalmente três processos fundamentais (WANG, 2009): Avaliação, detecção e alarme.

O processo de avaliação analisa as necessidades de segurança de um sistema, e produz um perfil para o sistema proposto (WANG, 2009). A detecção é o processo responsável por coletar e analisar os dados do tráfego de rede. A coleta pode ser realizada diretamente pelo IDS, ou através de software ou hardware adicional. Como exemplo de coletor têm-se: *sniffer*[1], SNMP (Simple Network Management Protocol)[2], *NetFlow*[3]. A análise processa os dados coletados procurando intrusões. Neste processo fica o coração do sistema de detecção de intrusão, pois é ele quem gera os alarmes. Existem diversas abordagens de análise, como: baseadas em assinaturas, e baseadas em anomalias, e serão vistas adiante. Os alarmes são a forma que o IDS interage com os usuários, ou sistema. Podem ser proativos e executar determinadas ações para

[1] Um *sniffer* é uma sonda localizada em determinado *host* da rede, com a placa de rede no modo promíscuo. Este tipo de sonda captura todo o tráfego que passa por seu segmento de rede.

[2] Através do protocolo SNMP um *host* pode acessar uma base de dados (MIB - Management Information Base) de algum dispositivo de rede, consultando seus descritores de rede.

[3] *NetFlow* é um protocolo de rede desenvolvido pela Cisco para coleta de informações de tráfego IP.

cessar a intrusão, ou podem ser passivos e somente gerar notificações ao administrador do sistema.

Um IDS pode ser classificado em três categorias (WANG, 2009): IDS baseado em *host* (HIDS), IDS baseado em rede (NIDS) e IDS híbrido. IDS baseado em *host* analisa o comportamento eventos do sistema, e comportamento do usuário, enquanto que IDS baseado em rede analisa o tráfego de rede. IDS híbrido une as funções de HIDS e NIDS.

HIDS é uma ferramenta utilizada para detectar intrusão em um único *host*. Neste tipo de sistema de detecção de intrusão é necessária a instalação do HIDS no computador que deseja-se detectar intrusão. O sistema monitora diversos recursos do computador no qual está instalado (WANG, 2009), como: elevação de usuário, acesso a disco, exclusão de arquivos de sistema, etc. HIDS também é conhecido como verificador de integridade do sistema (SIV) (WANG, 2009).

Como vantagens de um sistema HIDS temos: capacidade de monitorar conexões criptografadas, acesso a *logs* do sistema operacional, capacidade de detectar ataques fragmentados (WANG, 2009), acesso a modificações efetuadas pelos usuários do computador. HIDS também possuem algumas desvantagens, como: necessidade de instalação em todos os *hosts* protegidos, processamento e armazenamento adicional em todos os dispositivos que for instalado, não pode ser instalado em roteadores e *switches*. Na Figura 2.7 é definida a arquitetura básica de um HIDS, onde têm-se a presença do sistema (HBD), que recebe dados provenientes de *logs*. Após a análise, e em caso de detecção de uma intrusão o HIDS comunica-se com um módulo de gerenciamento de alarmes, que pode estar localizado no próprio computador, ou em algum servidor na rede.

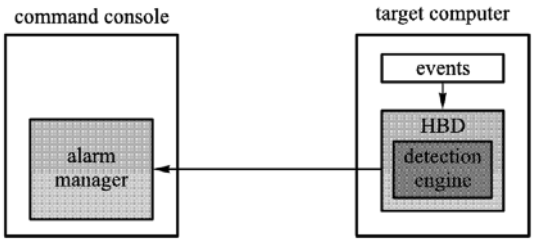

Figura 2.7: Arquitetura de um HIDS, onde o HBD é o HIDS, que possuí um sistema de detecção de intrusão, e recebe dados gerados por eventos. Fonte: (WANG, 2009)

Os NIDS são utilizados para detectar intrusão em redes de computadores, através do monitoramento do tráfego de rede. Um NIDS tipicamente possuí 2 componentes (WANG, 2009): uma sonda conectada na rede, e um componente de detecção. A sonda é responsável por capturar

descritores do tráfego de rede, e normalmente está conectada em algum ponto de interconexão da rede, como por exemplo: roteador de borda, *bridge*, *firewall*, etc. O componente de detecção é responsável por analisar as informações recebidas pela sonda e disparar alarmes em caso de haver intrusão no tráfego de rede.

As principais vantagens de um NIDS são: baixo custo (WANG, 2009) pois são necessários sondas somente em alguns pontos estratégicos da rede, monitoração passiva, isto é, não interfere no tráfego de rede normal, resistente a intrusão pois o processo de detecção pode funcionar em uma *bridge*[4], escalonável. NIDS também possuem algumas desvantagens, como: não é possível analisar protocolos criptografados, dificuldade de identificar ataques fragmentados. Na Figura 2.8 é definida a arquitetura básica de um NIDS, onde têm-se a presença do sistema (NBD), que recebe dados provenientes de uma sonda. Após o processo de detecção, caso exista uma intrusão, o NIDS comunica-se com um módulo de gerenciamento de alarmes, que pode estar localizado no próprio NIDS, ou em algum outro servidor na rede.

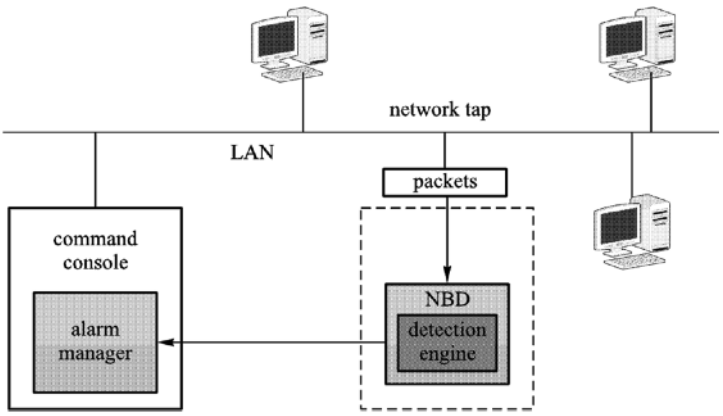

Figura 2.8: Arquitetura de um NIDS, onde o NBD é o NIDS, que possuí um sistema de detecção de intrusão, e recebe dados gerados por eventos. Fonte: (WANG, 2009)

Quanto ao intervalo de detecção, os NIDS podem ser classificados como (WANG, 2009): tempo real, processamento em lote e periódico. A detecção é considerada em tempo real quando os dados são analisados imediatamente após serem disponibilizados. Processamento em lote é quando os dados coletados atingem um determinado tamanho antes de serem processados, e o processamento periódico ocorre quando os dados são analisados de acordo com um intervalo de tempo.

Os NIDS podem ser classificados quanto ao método de análise dos dados normalmente

[4]Quando um computador atua como uma *bridge* de rede, o acesso a este computador pode ser desativado, aumentando a segurança do NIDS, pois o mesmo fica transparente na estrutura de rede.

de duas formas: baseados em assinaturas e baseados em anomalias. Sistemas de detecção de intrusão baseados em assinaturas (ROESCH, 1999) identificam ataques através da análise de assinaturas previamente estabelecidas sobre o comportamento padrão de determinado tipo de ataque, que são geradas por especialistas. Em NIDS baseado em anomalias é analisado o comportamento do tráfego de rede, sendo classificada como anomalia a ocorrência de um evento que não segue o comportamento esperado.

Existem diversos NIDS disponíveis para o uso, como o *Snort* (ROESCH, 1999) (*software* livre), NetRanger (CISCO, 2011) (*software* proprietário). Estes dois NIDS são baseados em assinaturas. Existem *addons*[5] para o *Snort* para incluir a detecção baseada em anomalias neste NIDS.

Os NIDS baseados em assinatura inspecionam eventos atuais e decidem se esses eventos são aceitáveis (WANG, 2009). As assinaturas são um conjunto de regras que permitem determinar o comportamento de uma intrusão. Caso alguma assinatura seja compatível com os dados do tráfego de rede analisado um alarme deve ser disparado.

Como o modelo de inspeção por assinaturas necessita de um conjunto de regras, a cada nova variante de um ataque deve ser construída uma nova assinatura. Por causa disto, é necessária a atualização constante do NIDS. Entretanto a construção de assinaturas é um processo manual, realizado por algum especialista que compreende o ataque. NIDS baseados em assinatura não possuem conjuntos de regras de ataques desconhecidos, assim como ataques mutantes.

A detecção de intrusão em NIDS baseado em assinaturas é bastante precisa (WANG, 2009) pois é comparado o comportamento da rede perante regras definidas. Isto ocasiona num baixo número de falsos positivos, mas como não detecta a variância em ataques conhecidos, ou ataques desconhecidos o número de ataques verdadeiros que não são identificados pode ser elevado.

A detecção de intrusão baseada em anomalias têm por definição que uma intrusão gera uma perturbação no comportamento padrão dos dados coletados (CHANDOLA; BANERJEE; KUMAR, 2009). Os NIDS baseados em anomalias modelam o comportamento do tráfego de rede com base nos dados analisados pelo sistema. Quando ocorre uma perturbação significativa no comportamento da rede, é então disparado um alarme.

Anomalias em redes de computadores podem ser causadas por intrusão, falha em algum *hardware* de rede, queda de algum segmento da rede, ou uso indiscriminado da rede. Um NIDS

[5] Adicionam novas funcionalidades a determinado sistema

baseado em anomalias é capaz de encontrar ou diferenciar anomalias. Os NIDS baseados em anomalias se adaptam a mudanças de comportamento ocorridas na rede, detectando assim ataques mutantes e novos tipos de ataques. Como o comportamento do tráfego de rede é altamente variável (conforme discutido na Seção 2.2) é difícil a construção de um modelo de comportamento da rede, ocasionando que mudanças legítimas podem ser erroneamente classificadas como anomalias.

As abordagens tradicionais utilizada em NIDS baseado em anomalias incluem: análise estatística (SAMAAN; KARMOUCH, 2008) (SCHERRER et al., 2007) (OHSITA; ATA; MURATA, 2004) e métodos baseados em aprendizagem de máquina (AHMED; ORESHKIN; COATES, 2007) (ZHANG; REXFORD; FEIGENBAUM, 2005) (CORRÊA; CANSIAN, 2007) (MAFRA et al., 2008). Nestas abordagens é necessário um período definido como treinamento para poder modelar o comportamento do tráfego de rede. Neste período o comportamento do tráfego de rede deve ser livre de intrusões para o correto treinamento dos algoritmos.

Uma alternativa para NIDS baseado em anomalia é o uso de técnicas de processamento de sinais, que estão sendo utilizadas com sucesso para detectar anomalias no tráfego de rede, devido a sua habilidade de detectar variações e transformar dados (PATCHA; PARK, 2007).

Nos últimos anos, métodos derivados de processamento de sinais, como Wavelets (MALLAT, 1998), têm sido utilizados para detectar anomalias. A transformada Wavelet permite a seleção de características do sinal através da representação de tempo-frequência combinada (NIELSEN, 1998), tendo sido utilizada em diversos analisadores de tráfego, para a detecção de anomalias (SOULE; SALAMATIAN; TAFT, 2005) (LU; TAVALLAEE; GHORBANI, 2008) (LI; LI, 2009) (BARFORD et al., 2002) (DALMAZO et al., 2009) (HUANG; THAREJA; SHIN, 2006) (DAINOTTI; PESCAPE; VENTRE, 2006).

2.3.1 Medidas de análise de desempenho de NIDS

Para permitir a análise dos resultados é necessário a definição de algumas métricas utilizadas neste trabalho:

- Verdadeiro Positivo (VP): é um alarme correto, gerado em resposta a um ataque ocorrido.
- Falso Positivo (FP): é um alarme incorreto, gerado quando não ocorreu um ataque.
- Falso Negativo (FN): é quando ocorre um ataque, e não é gerado um alarme.
- Verdadeiro Negativo (VN): é quando não ocorre um ataque, e não é gerado um alarme.

- Taxa de Detecção (TD): é o número de verdadeiro positivo (VP) vezes 100, dividido pelo número de ataques

2.4 Considerações Finais

Neste capítulo foram abordados aspectos dos ataques de negação de serviços, comportamento do tráfego de rede e sistemas detectores de intrusão.

Os ataques DoS dividem-se em duas categorias: esgotamento de recursos e esgotamento de banda. Ataques da categoria de esgotamento de recursos exploram vulnerabilidades, esgotando recursos de algum serviço ou protocolo, como por exemplo: *mailbomb*, entre outros. Ataques do tipo esgotamento de banda enviam um número excessivo de requisições, inundando o *host* atacado. Como exemplo de ataques DoS desta categoria temos: ICMP *flood*, *neptune*, entre outros.

O comportamento do tráfego de redes é difícil de ser modelado através de métodos numéricos e estatísticos, devido ao seu comportamento não estacionário, irregular, variante de intensidade. O tráfego de rede também possui algumas correlações temporais, como: dependências de longa duração e autossimilaridade.

Os NIDS são classificados quanto ao tipo de análise (assinaturas e anomalias), e intervalo de detecção. Os NIDS basedos em assinaturas são eficientes, mas não detectam ataques desconhecidos, ou mutações em ataques existentes. Como os sistemas de detecção de intrusão baseados em anomalias detectam o comportamento da rede, eles podem ser utilizados para detectar ataques desconhecidos, ou mutantes.

Os ataques de negação de serviço podem ser detectados analisando o *payload* do tráfego de rede, ou protocolos específicos com o uso de técnicas baseadas em assinaturas. Esta abordagem não é eficaz para a detecção de novos ataques, e ataques mutantes. Para permitir a detecção de novos ataques e ataques mutantes métodos baseados em anomalias devem ser utilizados, não devendo ser especificados protocolos ou comportamento para permitir a detecção destes tipos de ataques. Deve ser analisado o relacionamento entre os diferentes protocolos para detectar perturbações que os ataques ocasionam.

Uma das formas de avaliar um sistema de detecção de intrusão em redes de computadores é através do uso de métrica de desempenho. As principais métricas são: verdadeiro positivo, falso positivo, falso negativo, verdadeiro negativo e taxa de detecção.

3 WAVELETS

Neste capítulo, são apresentados os principais conceitos que envolvem a transformada wavelet discreta em uma e duas dimensões (Seção 3.1), com o objetivo de apresentar e explorar as operações que envolvem os coeficientes gerados pela transformada e que são a base do algoritmo do sistema de detecção de anomalias proposto neste trabalho.

A transformada wavelet é utilizada para a análise dos sinais de entrada (descritores do tráfego de rede). Por sua vez, as técnicas de truncamento (*threshold*) dos coeficientes wavelets são utilizadas no processo de geração de alarmes nas anomalias de rede. A Seção 3.2 detalha as técnicas que são responsáveis pela seleção e pelo truncamento dos coeficientes Wavelets, o que permite a identificação de variações significativas nos dados. Esta operação é o coração do sistema de detecção de anomalias proposto neste trabalho.

3.1 Transformada Wavelet Discreta

A Wavelet é uma ferramenta matemática para decomposição hierárquica de funções (STOLLNITZ; DEROSE; SALESIN, 1995) que pode ser utilizada para extrair informações de diferentes tipos de dados em diferentes níveis de resolução (NIELSEN, 1998). Além disso, permite descrever uma função em uma representação grosseira e um conjunto de detalhes.

A cada novo nível da transformação, os dados grosseiros do nível anterior são então decompostos em um novo conjunto de dados grosseiros e um novo conjunto de detalhes.

Para a reconstrução do sinal são combinadas as informações grosseiras e os detalhes (STOLLNITZ; DEROSE; SALESIN, 1995). Através de sua natureza multi-resolução da transformação, o uso de Wavelets é possível em aplicações onde escalabilidade e degradação são importantes, uma vez que estes fenômenos podem ser estudados ao longo dos diferentes níveis da decomposição gerada pela transformada. Neste trabalho, a transformada Wavelet é considerada para a análise do tráfego de rede.

3.1.1 Transformada Wavelet 1D

O ponto de partida para a transformada Wavelet é um sinal discreto $c_j[i]$, $i = 0, ..., N_{j-1}$ com $N_J = 2^{Jmax}$ pontos, considerado o nível mais fino de resolução. De um nível j para $j - 1$ metade dos pontos são transformados em valores médios (ditos coeficientes de escala $c_{j-1}[i]$) e a outra metade é transformada em informação complementar capaz de restaurar os

dados originais (denominadas coeficientes Wavelets $d_{j-1}[i]$).

Os coeficientes de escala são obtidos através da convolução com filtros passa baixa $L[k]$, $k = 0, ..., 2F - 1$ e os coeficientes Wavelets através da convolução com filtros passa alta $H[k]$, $k = 0, ..., 2F - 1$. De acordo com Daubechies (DAUBECHIES, 1992), estes filtros definem univocamente a família de funções Wavelets consideradas na transformação. A transformada discreta rápida envolvendo dois níveis consecutivos pode ser então estabelecida através das relações da Equação (3.1), com o índice i percorrendo todas as posições do vetor de entrada $c_j[i]$:

$$c_{j-1}[i] = \sum_{k=0}^{2F-1} L[k]c_j[2i + k], \qquad d_{j-1}[i] = \sum_{k=0}^{2F-1} H[k]c_j[2i + k]. \tag{3.1}$$

Quando $L[k] = [0.5, 0.5]$ e $H[k] = [0.5, -0.5]$ a transformada é conhecida como Transformada de Haar, cujos dados transformados são médias aritméticas de posições consecutivas (dois a dois) do vetor original.

A Equação (3.1) apresenta a formulação da transformada direta. O vetor de entrada é considerado como sendo um sinal discreto com $N_j = 2^J$ pontos, e o nível de resolução inicial é portanto J. A cada nível da transformada Wavelet discreta de *Daubechies* são calculados os coeficientes de escala c_{j-1} e Wavelets d_{j-1} dos valores discretos originais. Com isso, quando consideramos apenas os coeficientes c_{j-1}, a representação do sinal inicial é feita com a resolução mais baixa. Os dados $dj - 1$ são complementares e necessários para a reconstrução dos dados originais.

Na Figura 3.1 é apresentado um exemplo para ilustrar um sinal linear, com 3 níveis de transformação de Haar. Inicialmente os dados do vetor original $c_j[i]$ são representados como o sinal original. Então, os coeficientes de escala são representados em 3 níveis da transformada. A cada nível, a quantidade de coeficientes de escala é diminuida pela metade, o que faz com que haja uma diminuição na resolução de representação do sinal. Esta propriedade é utilizada diretamente na compactação do sinal. Os coeficientes wavelets também são apresentados nos 3 níveis da transformada. Conforme exibem os gráficos, os coeficientes wavelets de um sinal linear obtidos pela transformada de Haar são sempre constantes em cada nível.

Para reconstruir o sinal original são utilizados todos os níveis de coeficiente de escala, e o último nível dos coeficientes wavelets. Na Figura 3.1 para obter os dados originais é necessário a utilização dos dados dos três níveis de coeficientes de escala, e do terceiro nível de coeficiente wavelet.

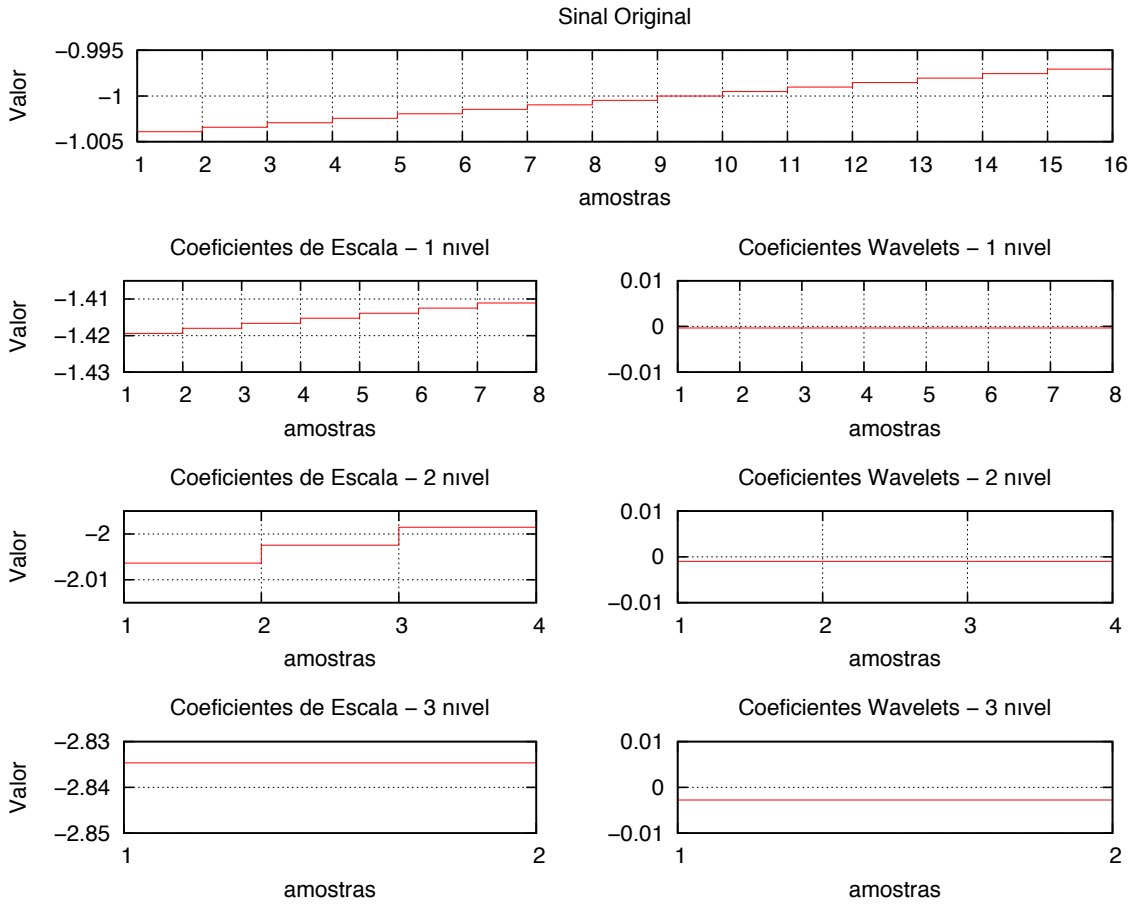

Figura 3.1: Representação de um sinal linear e aplicação da transformada de Haar

Na Figura 3.2 também é aplicada a transformada de Haar, desta vez em um sinal constante de valor igual a 3. Como não há perturbações no sinal de entrada, os coeficientes wavelets não detectam nenhuma perturbação. Os coeficientes wavelets nos três níveis são sempre iguais a zero, pois nestes coeficientes são armazenados as diferenças entre as amostras do sinal. Nos coeficientes de escala o valor aumenta a cada novo nível da transformada wavelet pois nestes coeficientes ficam armazenados a média entre os pontos.

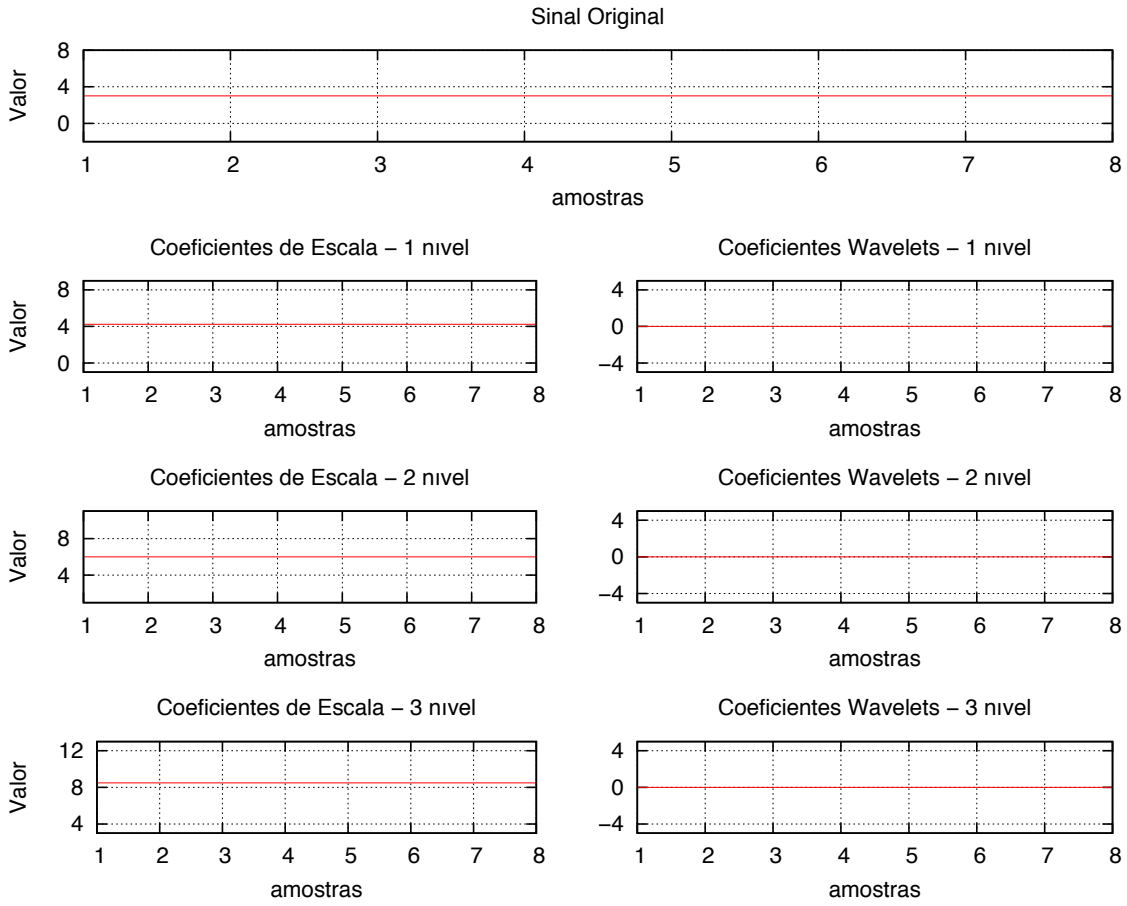

Figura 3.2: Representação de um sinal com valor constante igual a 3 e aplicação da transformada de Haar

Na Figura 3.3 é apresentado um sinal de uma função seno, perturbado com a adição de uma função Gaussiana. Conforme visto nas Figuras 3.1 e 3.2, na Figura 3.3 os coeficientes de escala também armazenam a informação mais grosseira do sinal original nos três níveis de transformada. Os coeficientes wavelets sofrem uma variação considerável somente onde o sinal é perturbado consideravelmente, ficando evidente a perturbação inserida pela função Gaussiana. Em todos os níveis os coeficientes wavelets exibem a perturbação do sinal original efetuada pela função Gaussiana.

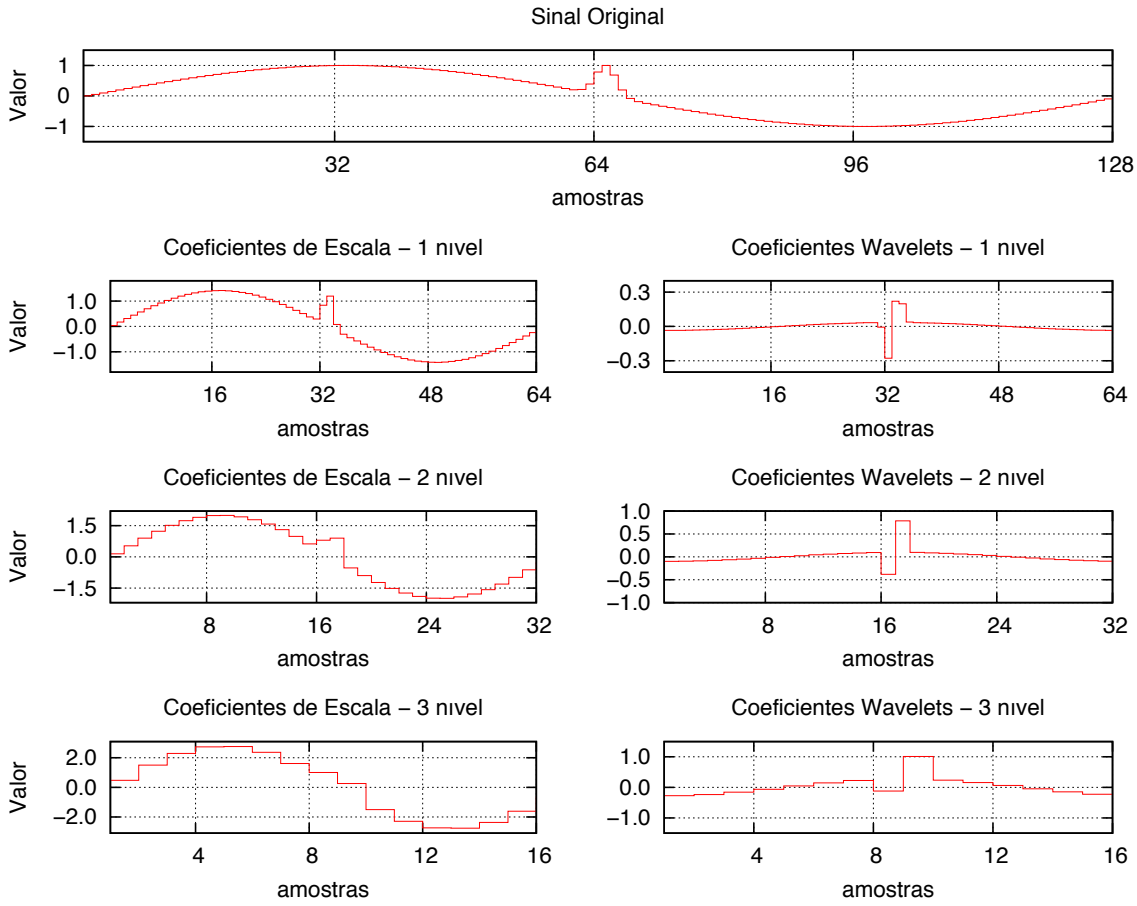

Figura 3.3: Representação de um sinal que demonstra uma função seno com adição de uma Gaussiana e aplicação da transformada de Haar

Ao ser aplicado a transformada Wavelet discreta, devemos utilizar uma família Wavelet (DAUBECHIES, 1992), tendo sido utilizadas neste trabalho as famílias de Haar, Daubechies Db2, Daubechies Db4, e Daubechies Db8. A Figura 3.4 exibe os gráficos das funções Escala e funções Wavelets, da família de Daubechies Db2, Db4 e Db8. A notação Db2, Db4, e Db8 indica que a função é da Família de Wavelets ortonormais de Daubechies, com 2, 4 e 8 momentos nulos respectivamente. Para obtenção das propriedades desta família de funções, ver (DAUBECHIES, 1992).

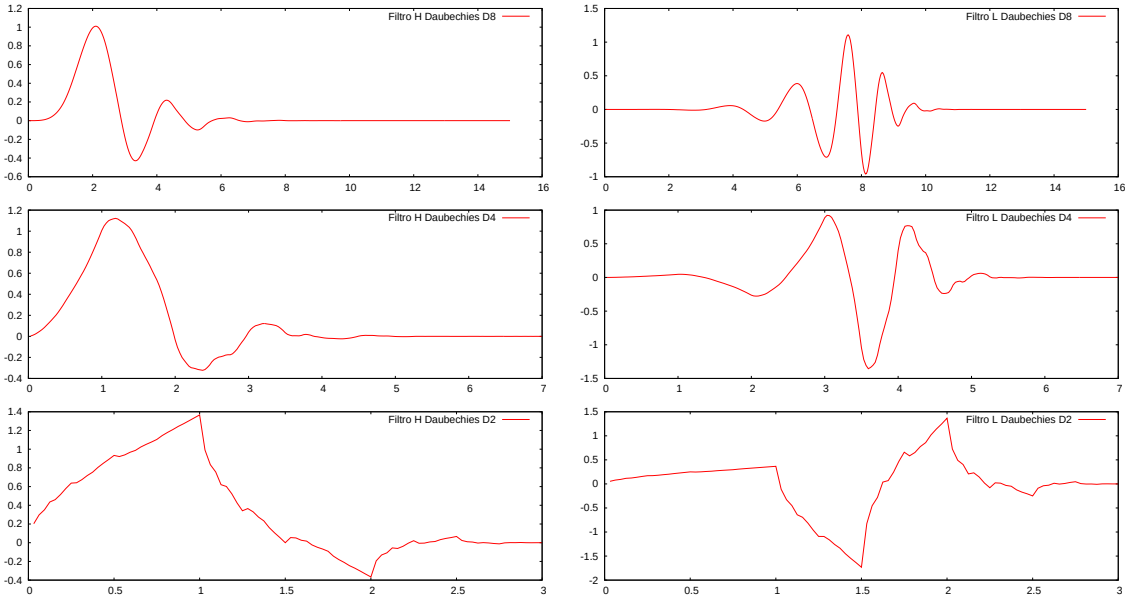

Figura 3.4: Representação do sinal das Wavelets de Daubechies Db8, Daubechies Db4, Daubechies Db2

A Figura 3.5 apresenta os gráficos da transformada wavelet de daubechies Db8 aplicada a um sinal constante igual a 3. Os coeficientes de escala representam de forma grosseira o Sinal Original, e os coeficientes wavelets representam a diferença entre os pontos. Como não houve perturbações no sinal original os coeficientes de escala e wavelets não apresentam nenhuma variação dos valores em nenhum dos 3 níveis da transformada wavelet.

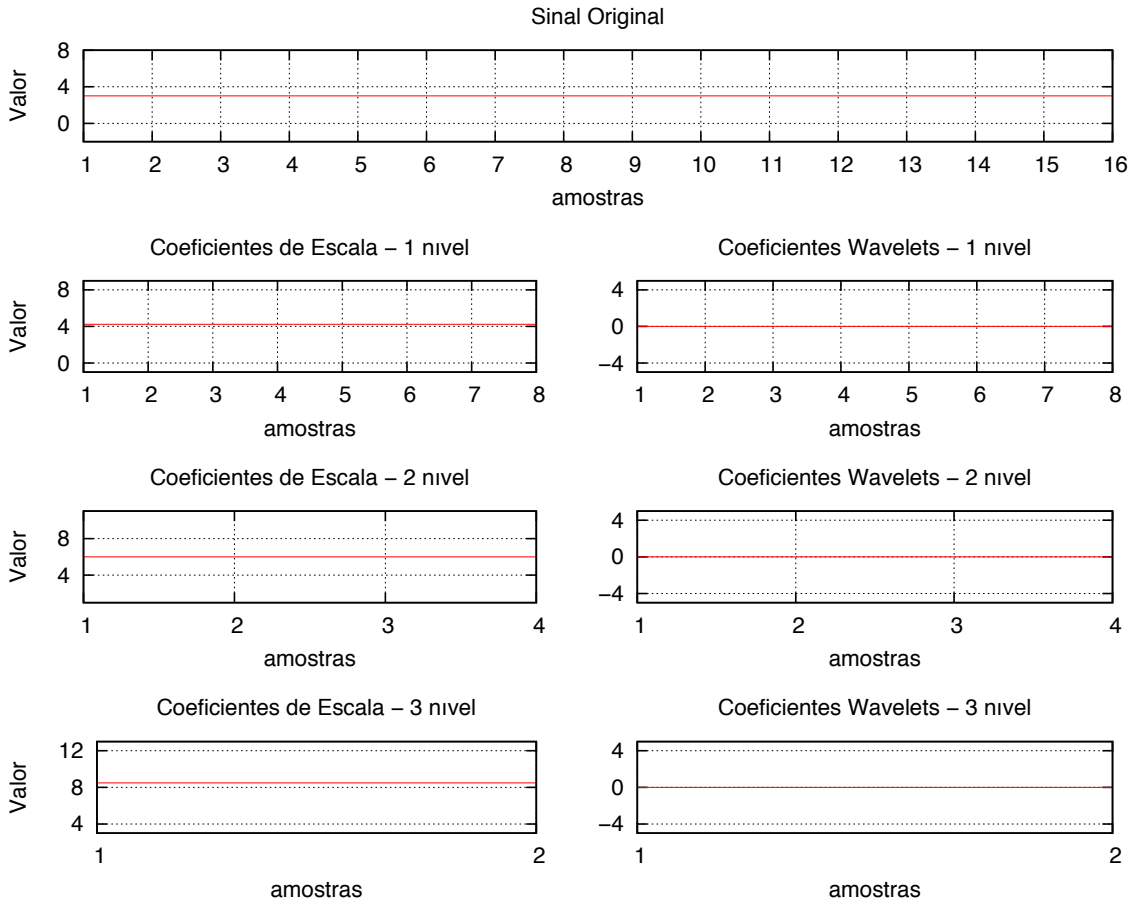

Figura 3.5: Representação de um sinal com valor constante igual a 3 e aplicação da transformada de Daubechies Db8

A Figura 3.6 exibe os gráficos da transformada wavelet de daubechies Db8 aplicada a um sinal linear. Fica evidente nos coeficientes de escala e wavelets uma perturbação ocorrida por um problema de extrapolação de fronteiras ocasionado pela família de Daubechies, demarcada por um retângulo azul . Um modo de extensão é através da periodização da fronteira, ou seja, o algoritmo interpola as primeiras posições do vetor de entrada para o final do mesmo, permitindo assim calcular as posições finais do vetor. Este modo é o que permite uma reconstrução dos níveis com uma maior precisão (AZEVEDO et al., 2010). Nas Figuras 3.5, 3.6, 3.7 são utilizados os modos periódicos de extensão de fronteiras.

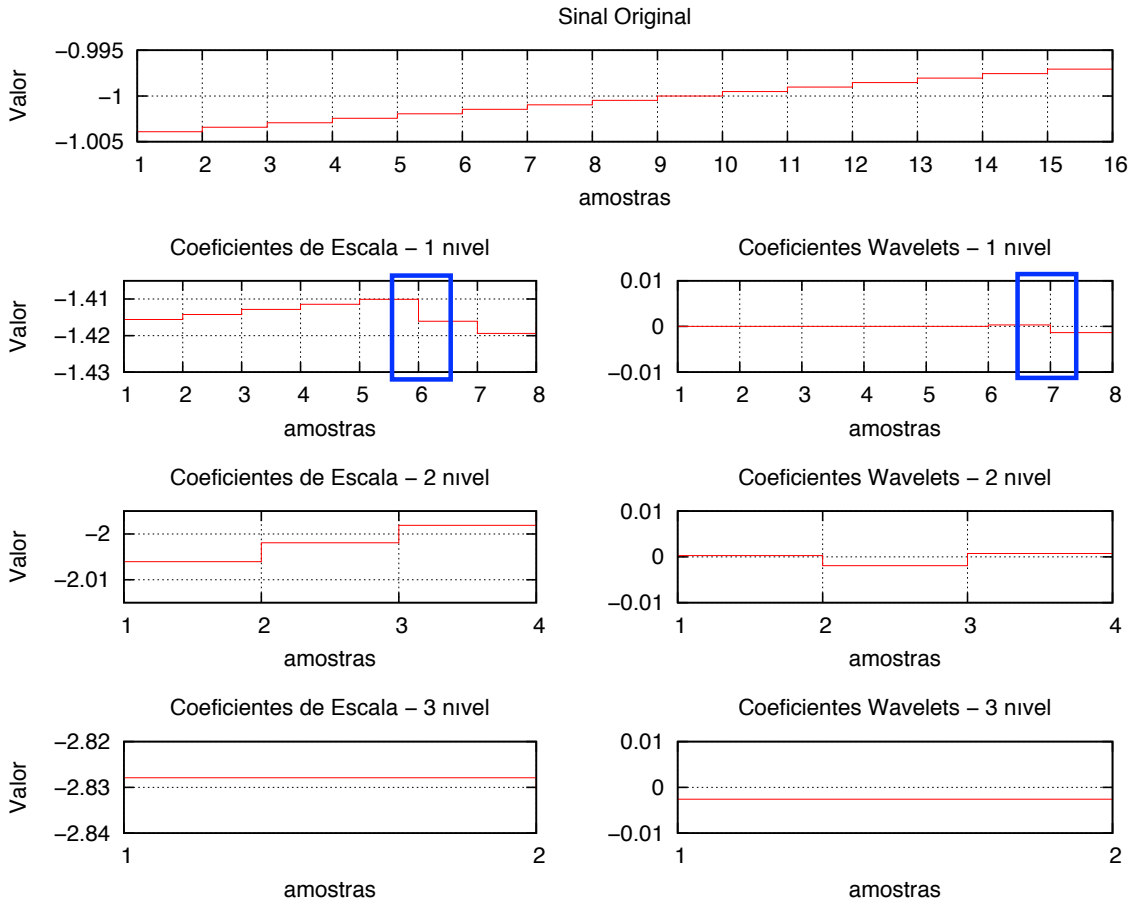

Figura 3.6: Representação de um sinal linear e aplicação da transformada de Daubechies Db8

A Figura 3.7 apresenta os gráficos da transformada wavelet de daubechies Db8 aplicada a um sinal de uma função seno perturbada por uma função gaussiana. No sinal original é apresentado uma perturbação na curva suave da função seno ocasionada pela função gaussiana. Nos coeficientes de escala a cada nova aplicação da transformada wavelet a perturbação fica menos evidente, conforme mostra a Figura 3.7 - Coeficientes de Escala - 3 nível. Isto ocorre pois a cada nível da transformada wavelet o sinal original é representado de forma mais grosseira, ocasionando a perda de informações (MALLAT, 1998). Os coeficientes wavelets detectam em todos os níveis da transformada wavelet a perturbação que ocorre no sinal original.

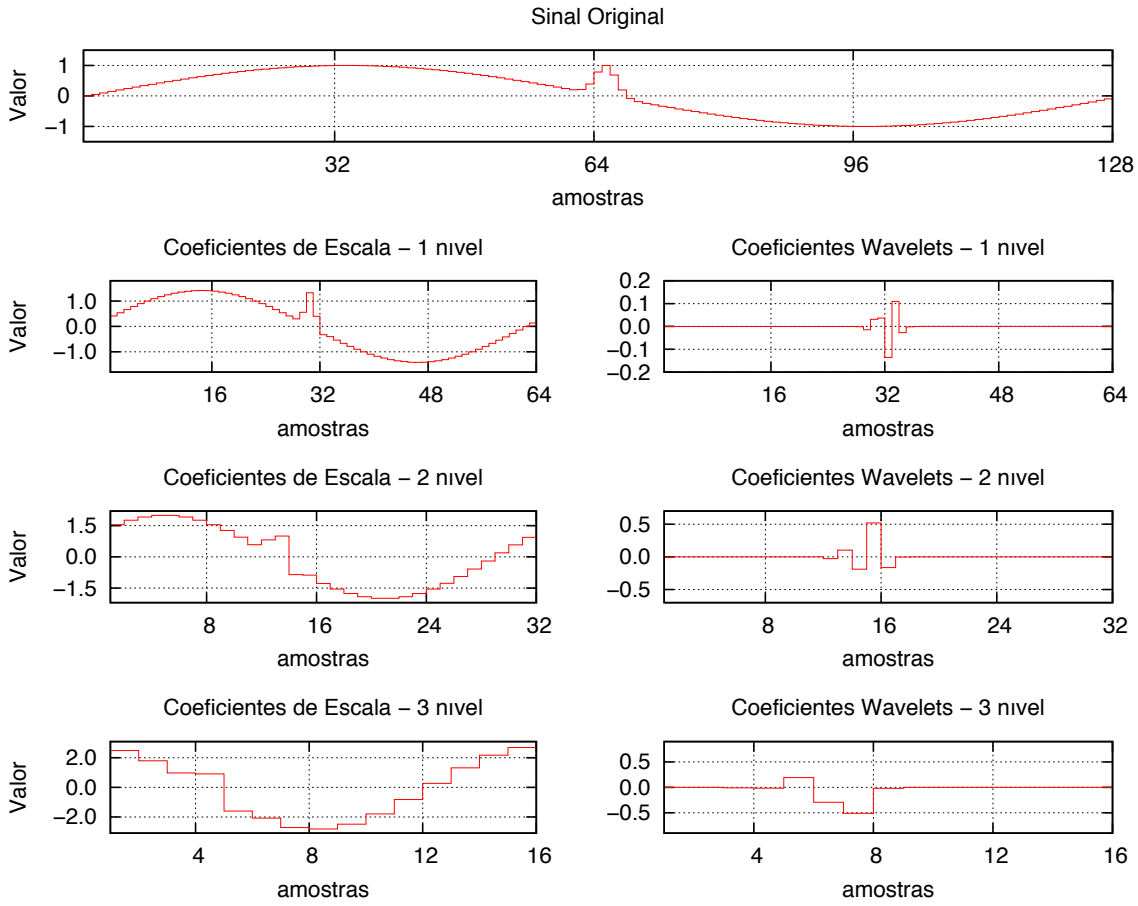

Figura 3.7: Representação de um sinal oriundo de uma função seno e gaussiana e aplicação da transformada de Daubechies Db8

A Figura 3.8 representa um esquema para ilustrar a aplicação da transformada Wavelet em vários níveis de decomposição. Uma vez obtidos os detalhes entre dois níveis, estes não serão mais decompostos. A operação é então realizada sobre os dados médios calculados em cada nível. Ao ser aplicado nos dados iniciais do vetor são gerados dois novos vetores: $c[j-1]$ que contém os coeficientes de escala, e $d[j-1]$ contendo os coeficientes wavelets. A cada aplicação da transformada wavelet nos coeficientes de escala é gerado novamente um novo nível de coeficientes de escala e wavelets.

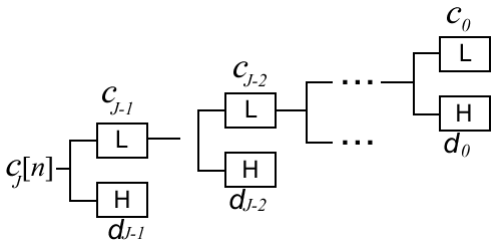

Figura 3.8: Decomposição de um sinal de entrada $c_j[n]$ com J níveis, sendo $c_j[n]$ o vetor de entrada e n o índice percorrendo as posições do vetor.

3.1.2 Transformada Wavelet 2D

A transformada Wavelet 2D é construída através da aplicação da transformada Wavelet 1D (Expressão (3.1)) em todas as linhas e depois em todas as colunas dos dados presente na matriz de entrada. A ordem da operação nas linhas e colunas depende dos diferentes tipos de algoritmos para a transformação bidimensional. Em (STOLLNITZ; DEROSE; SALESIN, 1995) é apresentada uma abordagem experimental e algorítmica sobre a transformação.

A transformada Wavelet é aplicada em todas as linhas, depois em todas colunas, repetindo este processo em cada nível de decomposição Wavelet. A Figura 3.9 ilustra este processo para um nível completo de transformação. Quando mais níveis são gerados, o único bloco que é decomposto é o bloco associado com a "média das médias", isto é, somente os coeficientes de escala do nível obtido anteriormente. Quando aplicada a transformada Wavelet 1D nas linhas, os dados originais são comprimidos (c) e um nível de detalhes (d) é gerado. O mesmo ocorre quando se aplica a transformada Wavelet 1D nas colunas. Como resultado da aplicação da transformada Wavelet 2D na matriz de entrada, as linhas e colunas são automaticamente analisadas e relacionadas. O relacionamento existente nas colunas pode ser analisado, trazendo uma compreensão complementar ao comportamento do ataque, não disponível quando somente a transformada unidimensional é aplicada ao tráfego de rede.

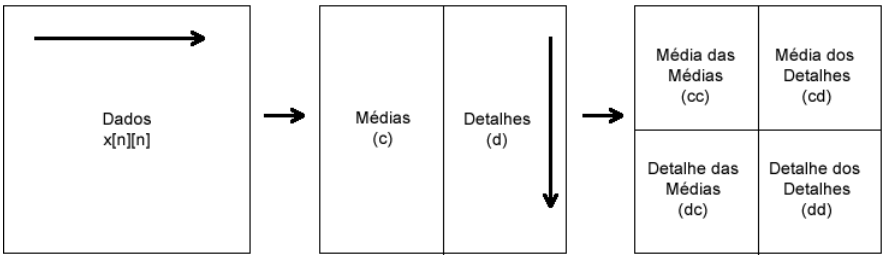

Figura 3.9: Decomposição de uma matriz $x[n][m]$ através da aplicação da transformada Wavelet bidimensional.

3.2 Técnicas de truncamento dos coeficientes Wavelets

Para a redução de ruídos em processamento de sinais, deve ser utilizada alguma técnica de truncamento (corte) dos coeficientes wavelets. A operação de corte é utilizada para selecionar os coeficientes Wavelets mais significativos e descartar informações irrelevantes (DONOHO; JOHNSTONE, 1995).

O processo de corte coerente da representação de tempo-escala, também conhecido como de aplicação de um *threshold*, é uma ferramenta promissora para expor características de anomalias

nos dados. A eficiência da detecção baseia-se na seleção dos coeficientes Wavelets (detalhes) mais relevantes, pois sua variação é o indicador de mudanças de comportamento nos dados.

A operação de corte consiste em comparar os coeficientes Wavelets com um limiar de corte τ. Os coeficientes Wavelets serão modificados de acordo com algum critério, que pode ser mais ou menos restritivo. A estratégia de corte denominada *hard thresholding* exibida no algoritmo da Figura 3.10 (DONOHO; JOHNSTONE, 1995) descarta todos os coeficientes menores que a limiar de corte τ (linha 2-3). Todos os coeficientes menores que τ são desconsiderados por serem considerados ruído.

```
   Entrada: coeficientes wavelets w
   Saída: coeficientes wavelets w após truncamento
1  para i ← 0 até N_j faça
2  |  se |d_j[i]| < τ então
3  |  |  d_j[i] = 0;
4  |  fim
5  fim
```

Figura 3.10: Algoritmo de *hard threshold* para corte de coeficientes Wavelet.

A estratégia de corte denominada *soft thresholding* exibida na Figura 3.11 (DONOHO; JOHNSTONE, 1995) diminui o limiar de corte τ de todos os coeficientes wavelets (linha 2-6). Nesta abordagem é considerado que o ruído está distribuído por todos os coeficientes wavelets (maiores e menores que τ).

```
   Entrada: coeficientes wavelets w
   Saída: coeficientes wavelets w após truncamento
1  para i ← 0 até N_j faça
2  |  se |d_j[i]| < τ então
3  |  |  d_j[i] = d_j[i] + τ
4  |  fim
5  |  senão se |d_j[i]| ≥ τ então
6  |  |  d_j[i] = d_j[i] - τ
7  |  fim
8  fim
```

Figura 3.11: Algoritmo de *soft threshold* para corte de coeficientes Wavelet.

Como o objetivo deste trabalho é detectar coeficientes que ultrapassem o valor de corte, os coeficientes menores podem ser descartados. Por este motivo é utilizado o *hard threshold*.

Depois que os coeficientes Wavelets modificados são obtidos, para reconstruir a informação no domínio de tempo é necessário aplicar a transformada Wavelet inversa, gerando os dados

filtrados correspondentes. Como neste trabalho o foco está na detecção de anomalias, e não na reconstrução do sinal esta operação não é tratada. Para informações sobre a transformada wavelet inversa consulte (MALLAT, 1989).

O valor de corte pode ser uma constante numérica, ou um valor calculado levando em consideração os dados de entrada.

O valor de corte pode ser determinado pelo valor de corte universal, proposto por Donoho (DONOHO; I., 1994), dado por $\tau = \sqrt{2 * \log(N)}\sigma$, em que $\sigma =$ variância dos coeficientes Wavelets e $N =$ número total de coeficientes wavelets. Variações deste valor podem ser obtidas quando apenas valores de um certo nível específico são considerados tanto no cálculo do desvio padrão, quanto na estimativa de N. Este valor de corte universal pode ser calculado a partir de um algoritmo adaptativo que gera o valor de corte τ levando em consideração os valores dos dados de entrada, adaptando-se a eventuais mudanças no comportamento nos dados. Para calcular o valor é necessário descobrir a variância dos dados, e após calcular o valor de corte através da raiz quadrada da variância e número de amostras vezes 2.

O valor de corte também pode ser calculado através de um algoritmo recursivo que utiliza o algoritmo adaptativo para calcular o valor inicial de corte τ, armazena o número de valores descartados (menores que o valor de corte τ) e calcula novamente a variância dos coeficientes wavelets descartados. Após calcular o novo valor de τ, o algoritmo verifica quantos valores foram descartados repetindo a operação recursivamente até que se estabilize o número de coeficientes descartados.

3.3 Considerações Finais

Neste capítulo foram inicialmente apresentados conceitos sobre a transformada wavelet discreta, para permitir o embasamento dos conceitos utilizados neste trabalho. Foram também discutidas as transformadas unidimensionais e bidimensionais.

A aplicação da transformada Wavelet permite extrair informações em diferentes níveis, permitindo uma função ser descrita em informações grosseiras e coeficientes de detalhes (ou wavelets). Os coeficientes wavelets armazenam informações complementares do sinal original.

Para calcular a transformada wavelet pode ser utilizado o algoritmo rápido de MALLAT (1998), que decompõem o sinal original por meio de sucessivos passos usando os filtros de uma determinada família wavelet.

A transformada wavelet 2D pode ser utilizada para extrair informações em dados bidimensi-

onais, e utiliza sucessivamente a transformada wavelet 1D nas linhas, e depois nas colunas. Para remover coeficientes waveletes irrelevantes são utilizadas técnicas de corte, como: adaptativa e recursiva.

Neste trabalho a família de funções wavelets considerada nas transformadas unidimensional e bidimensional é a família de *Daubechies* de funções ortonormais, de suporte compacto (funções não nulas apenas em intervalos finitos e fechados da reta), e que tem como propriedade fundamental a possibilidade de representarem exatamente funções polinomiais até um certo grau (DAUBECHIES, 1992).

4 PROPOSTA

Neste capítulo é apresentado um algoritmo para detecção de ataques DoS, baseado em wavelet Bidimensional e estratégias de corte dos coeficientes Wavelets. O algoritmo de detecção proposto é um dos módulos do sistema de detecção de intrusão de rede (NIDS) e é responsável pelo processo de detecção de anomalias no comportamento do tráfego de rede.

Na Seção 4.1 é apresentado o algoritmo de detecção baseado em wavelets bidimensionais. A Seção 4.2 detalha o sistema de detecção de intrusão em redes, do qual faz parte o algoritmo de detecção de anomalias; na Seção 4.3 são apresentados os trabalhos relacionados; e na Seção 4.4 são apresentadas as considerações finais deste capítulo.

4.1 Algoritmo para detecção de ataques DoS

Anomalias geradas por ataques ou problemas na estrutura da rede podem ocorrer no tráfego de rede de forma genérica. O foco deste trabalho é em um algoritmo para detecção de ataques de negação de serviço (DoS), que destina-se a prevenir o acesso legítimo de usuários a recursos da rede (LOUKAS; ÖKE, 2009).

Para definir o algoritmo proposto, foram consideradas duas hipóteses: o ataque DoS gera variação significante em um ou mais descritores de rede; e quando dois ou mais descritores são afetados, eles são relacionados no tempo.

Nesta seção é discutido o funcionamento em detalhes do algoritmo para detecção de anomalias. Na Subseção 4.1.1 é apresentada a modelagem dos dados de entrada do algoritmo; na Subseção 4.1.2 é discutida a transformada wavelet 2D, que é responsável pelo processamento dos dados de entrada; a Subseção 4.1.3 apresenta o algoritmo responsável por detectar e classificar perturbações nos dados, e por fim a Subseção 4.1.4 demonstra o processo de geração de alarmes.

4.1.1 Modelagem dos Dados

Para o correto funcionamento do algoritmo de detecção de ataques DoS é necessário que os dados sejam disponibilizados de forma consistente e definida. Os dados de entrada foram modelados como uma matriz ($m[][]$), em que as linhas são diferentes amostras de sinais que estão sendo analisados p, e as colunas são as instâncias de cada sinal amostrado a. O número de sinais analisados é n^p e o número de amostras é n^a, ambos devem ser múltiplos de 2 para

satisfazer a exigência do cálculo da transformada wavelet. A Figura 4.1 mostra a matriz $m[p][a]$, em que p identifica os sinais amostrados e a especifica as amostras no decorrer do tempo.

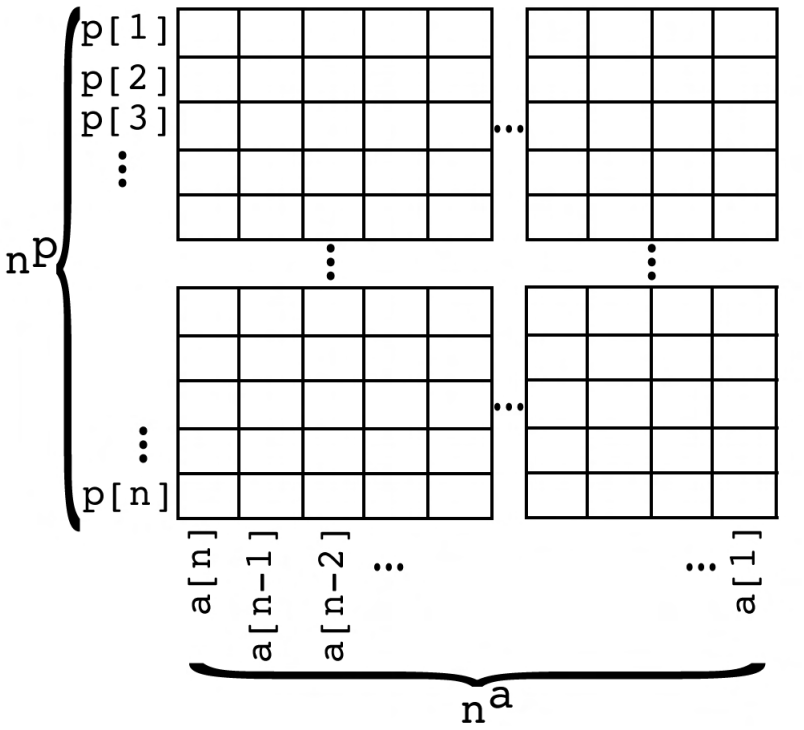

Figura 4.1: Matriz utilizada para aplicar a transformada Wavelet discreta 2D

O número de amostras é configurável, definido como uma janela deslizante. Uma janela deslizante é uma estrutura de dados que possui um número finito e constante de posições. A adição de novos valores na estrutura se dá através da inserção do novo valor e exclusão do valor mais antigo, possuindo a janela deslizante sempre o mesmo tamanho n^a.

A cada nova amostra de sinal disponível em um intervalo de tempo $\triangle t$ as colunas da matriz $m[p][a]$ são atualizadas com os novos valores disponibilizados. É importante salientar que o número de sinais analisados (n^p) e o tamanho da janela deslizante (n^a) são parâmetros configuráveis do algoritmo.

O uso de um intervalo finito e constante faz-se necessário para limitar o conjunto de dados a ser submetido a transformada wavelet, não aumentando o tempo de análise com o passar do tempo. Esta abordagem de janela deslizante também permite que informações antigas não interfiram no processo de análise.

Para que o algoritmo da transformada wavelet possa ser executado a matriz de dados $m[p][a]$ deve ser preenchida com n^a amostras, ou seja, somente após o preenchimento da matriz $m[p][a]$ o cálculo pode ser efetuado. Portanto o tempo de inicialização do algoritmo é: $\triangle t * n^a$ segundos, onde $\triangle t$ é a taxa de amostragem dos dados e n^a a quantidade de amostras.

Os dados de entrada do algoritmo de detecção de anomalias são atualizados através da janela deslizante, onde a cada intervalo de tempo $\triangle t$ são descartadas as amostras antigas e adicionadas as novas amostras.

O tempo de resposta do processo de detecção depende da escolha de um intervalo $\triangle t$ que seja capaz de detectar as anomalias presentes nos dados de entrada. Se for escolhido um tempo muito longo de atualização pequenos ataques podem passar desapercebidos. Se o intervalo escolhido for muito pequeno o algoritmo de detecção pode utilizar muito processamento.

4.1.2 Fazendo uso da Transformada Wavelet 2D

Neste trabalho após a construção da matriz de dados de entrada $m[p][a]$ é aplicada a transformada Wavelet discreta bidimensional na matriz $m[p][a]$. Como resultado do processamento da transformada Wavelet têm-se 4 novas matrizes: $medias[][]$, $detalhes\_medias[][]$, $medias\_detalhes[][]$ e $detalhes\_detalhes[][]$. A matriz $medias[][]$ contém os coeficientes de escala da transformação, que representam as informações originais em um nível de resolução mais grosseiro do que o inicialmente considerado. As outras três matrizes contém os diferentes coeficientes wavelets, os quais representam as variações de informação ocorridas nos dados da matriz $m[p][a]$ nas direções horizontal, vertical e diagonal.

Para o processamento da transformada wavelet deve ser utilizada uma família wavelet, como: *Haar*, *Daubechies Db2*, *Daubechies Db4*, ou *Daubechies Db8*. A escolha da família Wavelet utilizada é um parâmetro configurável do algoritmo, para permitir a escolha da família Wavelet apropriada para cada situação. Também é um parâmetro configurável do algoritmo o número de vezes que a transformada wavelet discreta bidimensional é aplicada nos dados de entrada da matriz $m[p][a]$.

O algoritmo exibido na Figura 4.2 mostra o processo de aplicação da transformada wavelet bidimensional. Para o cálculo da transformada Wavelet 2D é efetuada a transformada Wavelet unidimensional em todos os protocolos (linhas da matriz), conforme mostra as linhas 1 a 3. Após o cálculo da transformada wavelet nas linhas é calculada a transformada wavelet unidimensional das colunas, exibido nas linhas 4 a 6. O algoritmo é executado para cada nível de decomposição da transformada Wavelet bidimensional. Este processo resulta na análise das amostras e entre os descritores.

Como a transformada wavelet é uma ferramenta matemática que pode ser utilizada para detectar variações em dados, quando é aplicada a transformada 2D são detectadas variações que

ocorrem nos diferentes protocolos, e entre os protocolos de rede. O processo de aplicação da transformada unidimensional nas colunas da matriz $m[p][a]$ (linhas 4 a 6) relaciona os descritores presentes na matriz que é utilizada no cálculo. Este fato torna possível a detecção de ataques DoS que afetam diversos descritores de rede.

Como os ataques DoS são detectáveis em diferentes protocolos, de acordo com particularidade de cada ataque, a utilização de diversos protocolos permite a detecção baseada no comportamento da rede, e não em algum ataque específico.

Entrada: matriz de dados $m[p][a]$
Saída: matriz com os dados da transformada wavelet bidimensional w
`/* Calcula a TWD Unidimensional das linhas */`
1 **para** $i \leftarrow 0$ **até** n^p **faça**
2 $\quad w[i][] \leftarrow TWDUnidimensional(m[i][])$
3 **fim**
`/* Calcula a TWD Unidimensional das colunas */`
4 **para** $i \leftarrow 0$ **até** n^a **faça**
5 $\quad w[][i] \leftarrow TWDUnidimensional(w[][i])$
6 **fim**
7 **retorna** $medias[][]$, $detalhes\_medias[][]$, $medias\_detalhes[][]$, $detalhes\_detalhes[][]$

Figura 4.2: Algoritmo para Aplicação da Transformada Wavelet Discreta Bidimensional.

Caso seja utilizada a família *Daubechies* ocorre o problema de tratamento de fronteiras, conforme discutido na sub-Seção 3.1.1. Como os sinais utilizados neste trabalho são provenientes de descritores de rede, o comportamento é variável ocasionando na detecção de uma variação no final do sinal. A Figura 4.3 apresenta um sinal arbitrável variável no decorrer do tempo, contendo 32 amostras. A figura demonstra a transformada wavelet 2D de Daubechies Db8 com tratamento de fronteira periódica aplicada no sinal. Nos coeficientes wavelets demarcados com um retângulo azul fica evidente a variação do sinal no final das amostras, ocasionando pelo tratamento de fronteira periódica.

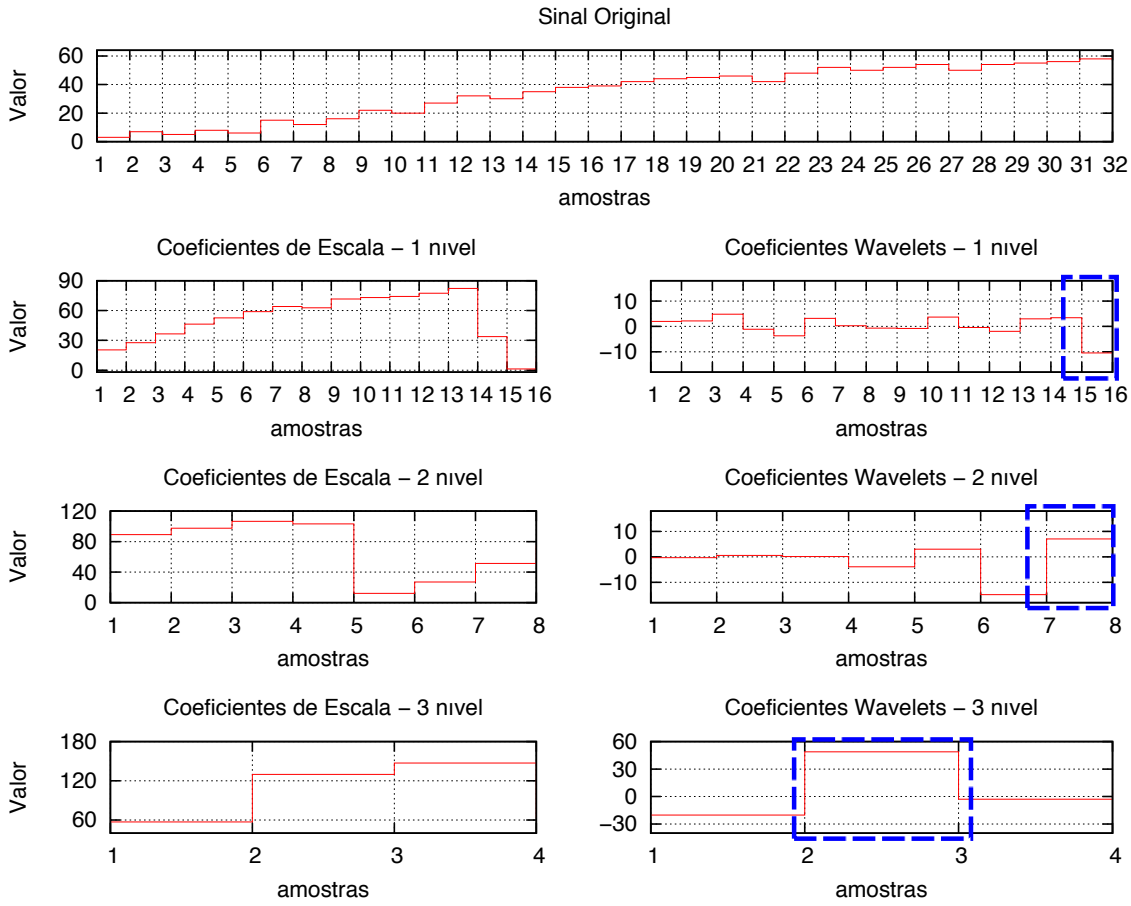

Figura 4.3: Representação de um sinal arbitrário variando no tempo e aplicação da transformada de Daubechies Db8 com tratamento de fronteira periódica

O problema apresentado na Figura 4.3 ocorre pois ao ser necessário dados adicionais para calcular as últimas posições do sinal, são extrapolados os primeiros valores de forma a permitir o cálculo da transformada wavelet. Neste trabalho optou-se por não utilizar a extrapolação periódica de fronteira, tendo sido utilizado a extrapolação via repetição do último valor. Nesta abordagem os dados da última amostra do sinal são replicados de forma a permitir o cálculo da transformada Wavelet da família de Daubechies.

A Figura 4.4 apresenta os resultados da transformada Wavelet com o tratamento de fronteiras via repetição do último valor com o mesmo sinal presente na Figura 4.3. Com esta forma de tratar a fronteira não ocorre o problema de variação do sinal presente no final das amostras, detectando somente as variações presentes no sinal original.

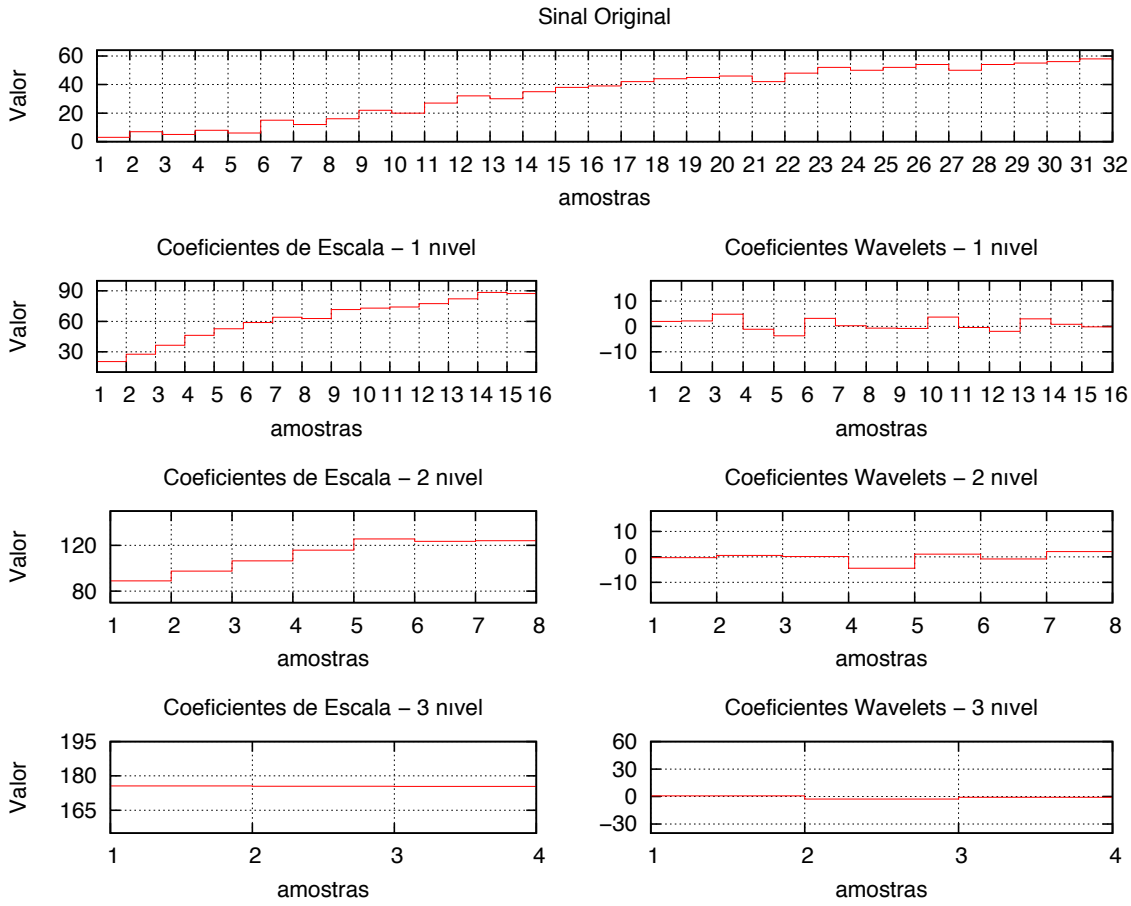

Figura 4.4: Representação de um sinal aleatório variando no tempo e aplicação da transformada de Daubechies Db8 com tratamento de fronteira via repetição do último valor

Como o tráfego de redes não segue uma distribuição normal, podem ocorrer pequenas variações de curta duração nos descritores de rede. Este fenômeno pode gerar alarmes que são falsos positivos. Para remover pequenas variações de curta duração os coeficientes wavelets são normalizados. Neste trabalho os coeficientes wavelets são normalizados utilizando a técnica da raiz quadrada: a normalização é realizada através do cálculo da raiz quadrada de todos os coeficientes.

4.1.3 Operação de Corte

Para detectar ataques DoS são aplicadas operações de corte nos coeficientes Wavelets obtidos no último nível da transformada, sendo esta a operação responsável por gerar alarmes no algoritmo. A operação de corte tem por objetivo descartar os coeficientes wavelets irrelevantes, mantendo somente os coeficientes relevantes ao problema, isto é, os coeficientes com valores elevados. Como os coeficientes menores que o valor de corte não são necessários para a detecção de ataques DoS, foi utilizado a técnica de *hard threshold*.

A escolha dos coeficientes que serão mantidos se dá pela aplicação de uma estratégia de corte nos coeficientes Wavelets. Todos os coeficientes devem ser analisados no cálculo do valor de corte, e após o cálculo todos os valores menores ou iguais ao valor de corte (τ) serão descartados. Neste trabalho foram aplicados dois métodos distintos para calcular o valor de corte: Adaptativo e Recursivo. Nestes dois métodos o valor de τ varia de acordo com os valores dos coeficientes wavelets, adaptando-se a variação de tráfego existente em uma rede de computadores.

Este valor de corte universal é calculado a partir de um algoritmo adaptativo, descrito no algoritmo presente na Figura 4.5. Primeiramente é calculada a variância dos coeficientes wavelets, através do somatório do quadrado de cada coeficiente wavelet (linhas 2 a 4) e posterior divisão pelo número de elementos nos coeficientes wavelets (linha 5). Como o parâmetro de corte utilizado é o universal (DONOHO; JOHNSTONE, 1995), o cálculo é efetuado através da raiz quadrada de 2 vezes o logaritmo do número de amostras vezes a variância, tudo multiplicado por uma constante C (linha 6).

```
   Entrada: coeficientes wavelets w
   Saída: parâmetro de corte τ
   /* Calcula a variância dos coeficientes                    */
1  σ ← 0
2  para i ← 0 até length(w) faça
3  |  σ ← σ + w[i]^2
4  fim
5  σ ← σ/length(w)
   /* Calcula o valor do parâmetro de corte                   */
6  τ ← sqrt(2 * log n^a * σ) * C
```

Figura 4.5: Algoritmo adaptativo para o Cálculo do parâmetro de corte (τ).

O valor de corte também pode ser calculado através de um algoritmo recursivo, descrito na Figura 4.6. O algoritmo recursivo utiliza o adaptativo para calcular o valor da variância e o valor inicial de corte τ (linhas 1-2). Após o valor inicial ser calculado ele define uma variável N_w para armazenar o número de coeficientes wavelets descartados. O bloco de código recursivo (linhas 4 a 9) começa com a variável $\bar{N_w}$ recebendo o número de coeficientes wavelets descartados pelo último valor de τ (linha 5). No bloco de repetição é calculado a variância dos coeficientes descartados (linha 6). Na linha 7 é calculado o valor de corte utilizando o parâmetro de corte universal, com a variância obtida somente dos coeficientes wavelets descartados. Após o novo valor de corte ser calculado são verificados quantos coeficientes wavelets são descartados pelo

valor de τ através da função $Descartados()$ (linha 8). Enquanto o número de coeficientes descartados (N_w) não for igual ao número de coeficientes descartados na rodada anterior ($\bar{N_w}$) os passos das linhas 4 a 9 são repetidos.

Entrada: coeficientes wavelets w
Saída: coeficientes wavelets w relevantes
/* Calcula a variância dos coeficientes */
1 $\sigma \leftarrow Variancia(w)$
/* Calcula o valor do parâmetro de corte */
2 $\tau \leftarrow \sqrt{2 * \log n^a * \sigma} * C$
3 $N_w \leftarrow 0$
4 **repita**
  /* Armazena o número de coeficientes descartados previamente */
5 $\bar{N_w} \leftarrow N_w$
  /* Calcula a variância dos coeficientes descartados */
6 $\sigma \leftarrow Variancia(w\_descartados)$
  /* Calcula o novo valor de corte */
7 $\tau \leftarrow \sqrt{2 * \log n^a * \sigma} * C$
  /* Verifica quantos coeficientes foram descartados pelo novo valor de corte */
8 $N_w \leftarrow= Descartados(w, \tau)$
9 **até** $\bar{N_w} == N_w$

Figura 4.6: Algoritmo recursivo para o Cálculo do parâmetro de corte.

A escolha do algoritmo de corte é um parâmetro configurável do algoritmo de detecção, assim como o valor da constante C.

4.1.4 Geração de Alarmes

Após realizar a operação de corte (Subseção 4.1.3) o algoritmo detector de anomalias deve verificar os dados, identificando ataques DoS existentes na rede. Para realizar esta tarefa o algoritmo verifica as matrizes $detalhes\_medias[][]$, $medias\_detalhes[][]$, e $detalhes\_detalhes[][]$ geradas pela transformada wavelet bidimensional. Se a amostra mais recente de cada matriz for diferente de zero, isto é, maior que o valor de corte τ, em mais de uma sub-banda de detalhe, será disparado um alarme pelo algoritmo.

A Figura 4.7 apresenta o algoritmo que mostra o processo de disparada de alarmes. Para cada sub-banda de detalhes (detalhes dos detalhes, detalhes das médias e médias dos detalhes - linha 2) é verificado se existe algum coeficiente maior que 0, ou seja, maior que o valor de corte τ (linha 3), e se não é o fim de um ataque de longa duração (linha 3). Somente o

último valor de cada sub-banda de detalhes é utilizada, pois contém o valor mais recente dos sinais amostrados. Caso algum coeficiente seja diferente de zero é incrementado a variável $acumulador$ (linha 4) e é efetuada uma chamada ao método $media\_inicio\_ataque()$ (linha 5). Caso o valor do acumulador seja maior ou igual a dois é disparado um alarme (linha 8-9). O método $media\_inicio\_ataque()$ faz uma média com os valores dos coeficientes de escala e armazena para posterior análise. Para avaliar se é o final de um ataque de duração maior que a janela deslizante, o método $fim\_ataque()$ faz a média dos coeficientes de escala e compara com a média obtida no início do ataque (método $media\_inicio\_ataque()$). Se a média obtida pelo método $fim\_ataque()$ for pelo menos um percentual definido maior é detectado o final de um ataque de longa duração e não é gerado um novo alarme.

O tempo necessário para a detecção de ataques é dependente do intervalo de atualização das amostras de rede ($\triangle t$), pois o algoritmo é executado a cada intervalo de atualização. Isto é, se o valor de $\triangle t$ for igual a 30 segundos, o tempo para detectar um ataque pode ser de até 30 segundos.

```
   Entrada: coeficientes wavelets w
   Saída: alarmes
 1 acumulador ← 0
   /* Verifica as 3 sub-bandas de detalhes em busca de alarmes        */
 2 para i ← 0 até 2 faça
 3 |   se w[i][length(w[i])] ≠ 0 && ¬fim_ataque() então
 4 |   |   acumulador ← acumulador + 1
 5 |   |   media_inicio_ataque()
 6 |   fim
 7 fim
 8 se acumulador ⩾ 2 então
 9 |   Dispara Alarme
10 fim
```

Figura 4.7: Algoritmo que verifica pela existência de alarmes.

A Figura 4.8 apresenta os gráficos da matriz $medias\_detalhes[][]$ utilizando a estratégia adaptativa (a) e recursiva (b), que contém os coeficientes wavelets presentes na matriz onde exibe a detecção de um ataque DoS. Em Azul o parâmetro de corte calculado, e em vermelho os coeficientes wavelets antes do processo de corte.

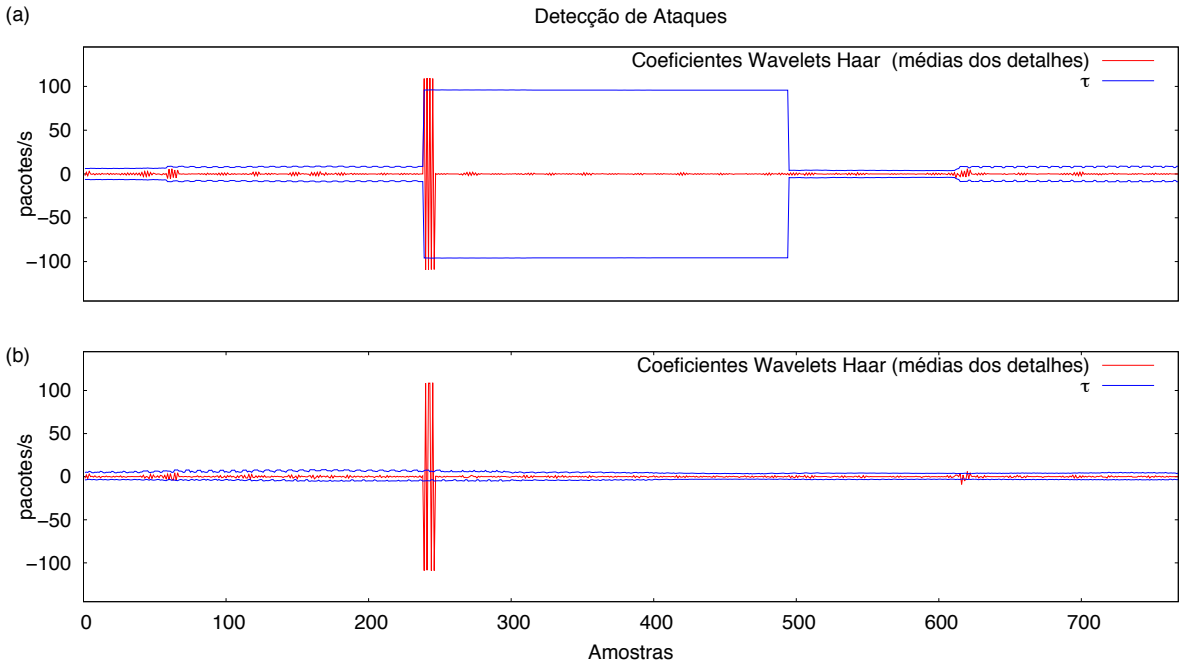

Figura 4.8: Exemplo de detecção de um ataque na matriz $medias\_detalhes[][]$ utilizando a estratégia de corte adaptativa (a) e recursiva (b).

O processo de geração de alarmes é dependente da wavelet utilizada. Caso seja utilizada a wavelet de Haar são disparados alarmes quando são encontradas anomalias em duas ou mais sub-bandas de detalhes, pois esta wavelet detecta pequenas perturbações. Caso seja utilizada a família de *Daubechies* Db2, Db4 ou Db8 são disparados alarmes quando é encontrada uma anomalia em uma ou mais sub-bandas de detalhes.

4.2 Sistema de detecção de intrusão em redes

Para a validação do algoritmo detector de anomalias proposto neste trabalho, foi desenvolvido um sistema de detecção de intrusão em redes (NIDS). Este NIDS permite capturar o tráfego de rede e gerar notificações sobre ataques. A arquitetura do NIDS foi construída de forma modular para permitir a personalização do detector de intrusão. O sistema proposto divide-se em 3 módulos: módulo de coleta, módulo de detecção e módulo de relatórios.

A Figura 4.9 apresenta o *workflow* do sistema proposto. O processo inicial é configurar as variáveis básicas do detector de intrusão. O próximo processo é instanciar os módulos: coleta, detecção e relatórios. Ao ser instanciado cada módulo deve ser escolhido pelo menos uma instância de cada módulo, como por exemplo: `SnifferSintético`, `Wavelet2DSQRT` (Wavelet 2D com normalização pela raiz quadrada) e `ReportConsole`. Após a instanciação dos módulos, o módulo de coleta captura os dados e envia para o módulo de detecção, este por sua

vez detecta os ataques e repassa para o módulo de relatórios os alarmes disparados. O módulo de relatórios finaliza o processo de detecção de anomalias, gerando um *log*, ou efetuando um bloqueio no *firewall*.

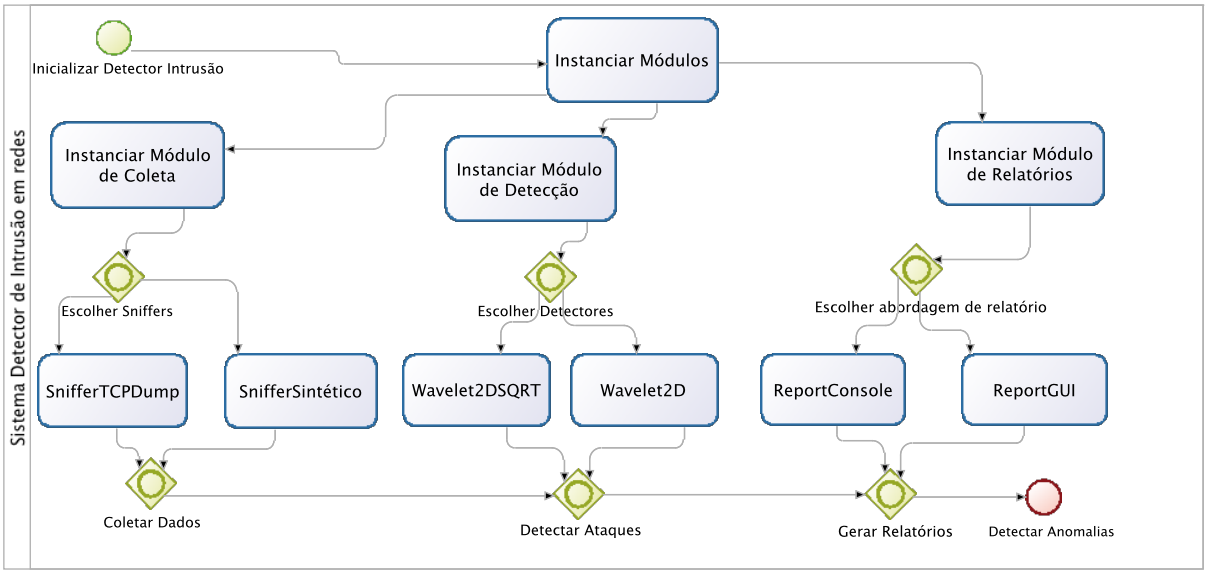

Figura 4.9: *Workflow* do sistema de detecção de intrusão proposto neste trabalho.

O NIDS proposto neste trabalho faz uso de *design patterns* para permitir a utilização do sistema em diferentes cenários, como por exemplo: testar diversos algoritmos de detecção, receber informações de diferentes coletores, adicionar regras no firewall de forma autônoma, emitir relatórios por email, etc.

O sistema de detecção de intrusão em redes foi desenvolvido na linguagem de programação Java 1.6[1] e C utilizando o ambiente de desenvolvimento integrado NetBeans[2].

A Subseção 4.2.1 apresenta o módulo responsável pela captura e disponibilização de informações do tráfego de rede. Na Subseção 4.2.2 é detalhado o módulo de detecção de anomalias, e por fim na Subseção 4.2.3 é apresentado o módulo de relatórios do *framework*.

4.2.1 Módulo de Coleta de dados

Para coletar informações sobre o tráfego de rede, fez-se necessário o desenvolvimento de um módulo para capturar informações no nível de transporte da rede. O módulo de coleta é responsável por capturar o tráfego de rede, que posteriormente será utilizado para detectar anomalias no módulo de Detecção. Este módulo utiliza o padrão de desenvolvimento *Abstract*

[1]http://www.java.com
[2]http://www.netbeans.com

Factory, podendo ser conectado em diversos tipos de sondas para captura de dados, como por exemplo: *sniffer* que coleta o tráfego de rede, base de dados sintética para avaliação *offline* e *NetFlow*[3]. Este módulo captura diversos descritores de rede para enviar as informações para o módulo de Coleta. Como exemplo deste descritores temos: IP, TCP, UDP, ICMP, TCP-SYN e Tamanho do *Payload*.

A ordem dos dados a serem disponibilizados na matriz de dados é importante no processo de detecção, devendo ser organizadas de acordo com a intensidade de cada sinal. No módulo de coleta proposto nesta subseção são utilizados os seguintes protocolos em ordem decrescente de intensidade: IP, TCP, UDP e ICMP. Caso os protocolos não sigam esta ordem, o número de falsos positivos é muito elevado pois há muita diferença entre as linhas da matriz, elevando consideravelmente o valor dos coeficientes wavelets, impactando no valor de corte τ.

A Figura 4.10 mostra o diagrama de classes do módulo de coleta, representando a interface `SnifferInterface` que define os métodos e variáveis obrigatórios nos coletores disponíveis. Foram desenvolvidos 2 coletores distintos neste trabalho: um coletor sintético (Classe `SnifferSintetico`) que é responsável por importar uma base de dados para verificação de ataques, e um *sniffer* (Classe `SnifferTCPDump`) que é responsável por capturar o tráfego de dados de uma determinada interface de rede. Na inicialização do sistema são configurados os parâmetros para o funcionamento do *sniffer*.

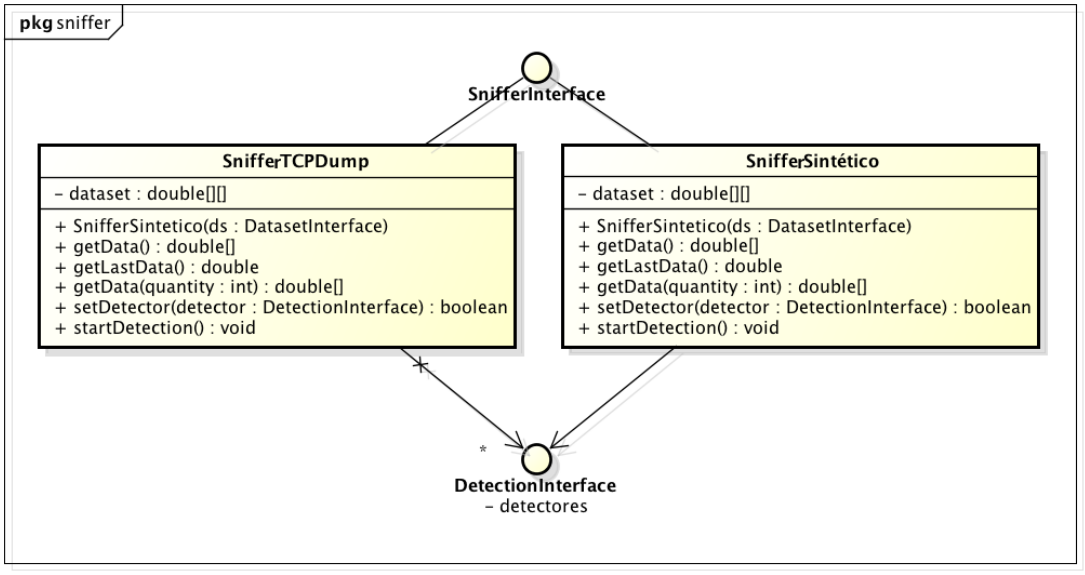

Figura 4.10: Diagrama UML do módulo responsável por coletar dados.

Para efetuar testes do algoritmo de detecção de anomalias, foi desenvolvido para o coletor

[3]*NetFlow* é um protocolo de rede desenvolvido pela Cisco para coleta de informações de tráfego IP.

sintético *parsers* para carregar base de dados distintas. A Figura 4.11 exibe o diagrama de classes destes *parsers*. Todos eles implementam a interface `DatasetInterface` permitindo a adição de novas bases de dados ao *framework*. Neste trabalho foram implementadas classes para trabalhar com dados das seguintes bases de dados: UFSM (Classe `UFSMDataset`), DARPA (Classe `DarpaDataset`), entre outras bases para testes (`AtaquesDataset, Stats,` etc).

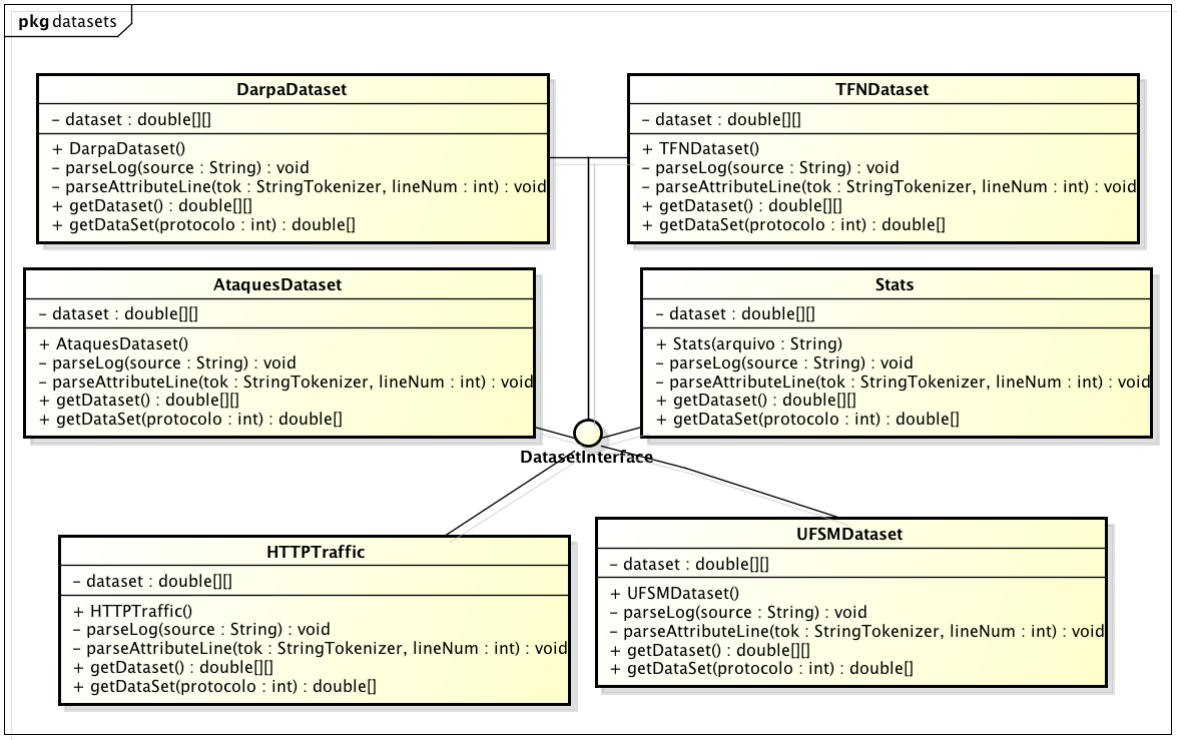

Figura 4.11: Diagrama UML do módulo responsável por disponibilizar dados ao Coletor.

Para a coleta de dados em redes de computadores foi implementado um coletor do tipo *sniffer*, desenvolvido em linguagem C, que utiliza a biblioteca *libpcap* [4] para a captura de pacotes. O coletor conecta-se ao módulo de coleta (desenvolvido em Java) através de uma conexão TCP. O *sniffer* captura informações que trafegam em uma determinada interface de rede[5], preferencialmente onde o tráfego da rede monitorada deve ser replicado. São geradas estatísticas a cada intervalo de tempo $\triangle t$, e estas são enviadas ao módulo de coleta de dados.

O coletor permite a parametrização de sua configuração, permitindo limitar o escopo dos dados capturados, podendo ser escolhido um segmento de rede, determinada porta, etc. Também pode ser configurado o intervalo de tempo $\triangle t$ no qual são geradas as estatísticas.

O *sniffer* foi desenvolvido com o objetivo de capturar todos os pacotes que trafegam na

[4] http://tcpdump.org

[5] Para a geração de estatísticas neste trabalho o *sniffer* foi conectado a uma placa de rede conectada em uma porta onde o roteador de borda da instituição de ensino replica todo o tráfego de rede.

rede, sendo esta uma atividade de alto custo computacional. Para permitir a captura do número máximo de pacotes presentes na rede, o *sniffer* ignora o *payload* de todos os pacotes de rede, somente capturando informações básicas do cabeçalho do pacote, como: endereço de origem, protocolo, endereço de destino, porta de origem, porta de destino, flags, e tamanho do pacote. O custo computacional necessário para analisar o cabeçalho permite o uso em redes com tráfego de milhares de pacotes por segundo[6].

A comunicação entre o *sniffer* e o módulo coletor (classe `SnifferTCPDump`) se dá através de comunicação via pacotes datagramas (UDP) utilizando a abordagem *push*, onde o sniffer envia a cada intervalo $\triangle t$ as estatísticas ao módulo coletor. O módulo coletor após receber as informações se resposabiliza por disponibilizar os dados para o módulo de detecção.

4.2.2 Módulo de Detecção de Intrusão

Este módulo é o responsável pela detecção de anomalias. O módulo de coleta repassa os descritores de rede para o módulo de detecção analisar as informações sobre a rede. Assim como no módulo de coleta, a implementação deste módulo utiliza o padrão de desenvolvimento *Abstract Factory*, permitindo a construção de diversas abordagens para a detecção de intrusão.

O algoritmo que implementar a interface do módulo de detecção precisa lidar com os dados sobre a rede, disponibilizados pelo coletor, e disparar notificações em caso de ataques através do módulo de relatório. O processamento necessário sobre os dados de rede podem ser distribuídos para diminuir o tempo de processamento, permitindo uma melhor escalabilidade deste módulo. Este módulo é o que mais consome recursos computacionais no sistema de detecção de intrusão em redes.

A Figura 4.12 exibe o diagrama de classes do módulo de detecção, que possuí duas classes de detecção implementando a interface `DetectionInterface`. Para o desenvolvimento de novos métodos de detecção é necessário somente implementar a interface, não necessitando de conhecimento prévio dos outros módulos do sistema. A classe `Wavelet2D` implementa o algoritmo da transformada wavelet 2D sem normalização dos coeficientes wavelets, enquanto que a classe `Wavelet2DSQRT` implementa a transformada 2D com normalização dos coeficientes wavelets através da raiz quadrada.

[6]Nas simulações realizadas neste trabalho foram capturados 40 mil pacotes por segundo, não tendo sido detectado nenhum descarte.

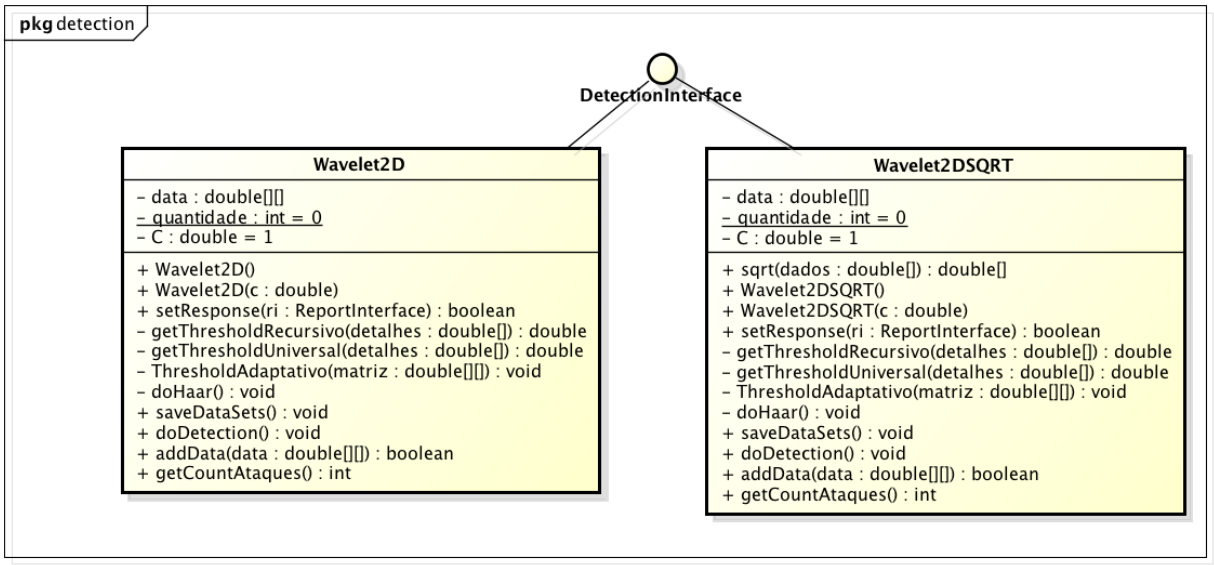

Figura 4.12: Diagrama de classes do módulo responsável pela detecção.

O algoritmo proposto na seção 4.1 foi implementado neste módulo como sendo parte da classe `Wavelet2D` para a detecção de ataques do tipo DoS. O módulo de detecção é o módulo mais importante do sistema para detecção de intrusão pois é nele que fica o algoritmo responsável por analisar os dados, e disparar alarmes.

Na inicialização do sistema devem ser configurados alguns parâmetros do detector de anomalias como: família wavelet, algoritmo de corte e tamanho da janela deslizante. As famílias disponíveis são: Haar, Daubechies Db2, Daubechies Db4 e Daubechies Db8. O algoritmo de corte pode ser adaptativo ou recursivo. Para instalar o NIDS proposto nesta seção é necessário uma máquina virtual Java. Caso seja escolhido para o módulo de coleta uma instância da classe `SnifferTCPDump` é requisito o sistema operacional Linux, com a biblioteca *libpcap* instalada e configurada.

4.2.3 Módulo de Relatórios

Para permitir a notificação do administrador de redes, ou disparar ações proativas foi desenvolvido o módulo de relatórios. Este é o módulo responsável por gerar notificações e/ou disparar ações do sistema.

A Figura 4.13 apresenta o diagrama de classes deste módulo. Neste trabalho foram gerados duas abordagens de relatório: modo texto (Classe `ReportConsole`), e modo gráfico (Classe `ReportGUI`). Estas classes armazenam os dados referentes ao processo de detecção, permitindo uma posterior análise do comportamento do algoritmo.

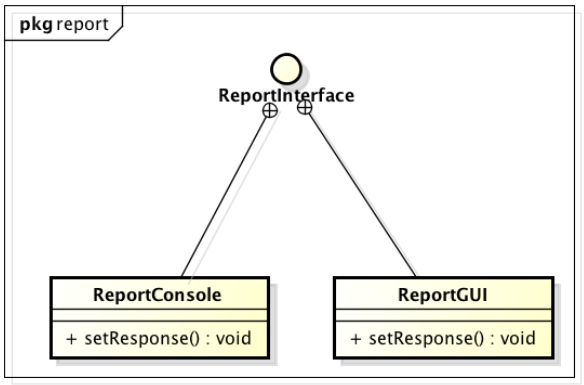

Figura 4.13: Diagrama de classes do módulo responsável pelo relatório.

Podem ser desenvolvidas novas estratégias de relatório implementando a interface genérica `ReportInterface` para por exemplo enviar logs e alertas por email para o administrador da rede. O módulo de detecção aciona então os métodos disponibilizados pela interface `ReportInterface` para alertar o administrador sobre ataques DoS na rede monitorada pelo módulo coletor.

Para permitir a integração com outros sistemas, é possível adicionar diversas instâncias deste módulo no sistema de detecção de intrusão. Pode ser implementada uma classe que tome ações proativas, e adicione regras no *firewall* de forma proativa, assim como um sistema para visualização dos dados gerados pelo NIDS, mantendo um histórico dos alarmes e comportamento de rede.

4.3 Trabalhos Relacionados

Para diminuir o número de falsos positivos Dainotti, Pescape, e Ventre (DAINOTTI; PESCAPE; VENTRE, 2006) desenvolveram um algoritmo para a detecção de anomalias baseado nos métodos CUSUM, EWMA e thresholds adaptativos utilizando a transformada Wavelet contínua. Os autores propõem uma arquitetura que utiliza duas camadas para a detecção de anomalias: a primeira mais simples, que busca de forma grosseira encontrar ataques, e repassa a informação a segunda camada, responsável por um processamento especialista (utiliza-se de uma biblioteca de ataques) para filtrar os dados, e diminuir o número de falsos positivos. A ferramenta proposta foi analisada em 3 bases de dados: DARPA 1999, UCLA e UNINA, sendo os ataques DoS injetados nas base de dados. Foram utilizadas duas ferramentas para geração de ataques: TFN2K e Stacheldraht e a geração através da modelagem pelo Matlab. Os autores utilizaram as métricas: taxa de acertos médios e taxa de erros médios.

São analisadas diferentes famílias Wavelets em (LU; TAVALLAEE; GHORBANI, 2008) para verificar o impacto de bases Wavelet no desempenho da aplicação, sendo proposta uma ferramenta para aplicação das diversas famílias Wavelets. A ferramenta utiliza uma arquitetura onde é executada a transformada Wavelet, e então é modelado o tráfego normal através da aplicação de um modelo de predição auto regressivo ARX (*Auto Regressive with eXogenous input*) que gera um modelo de tráfego normal, e então é analisado os resíduos da predição para a detecção de anomalias. Foram utilizados os dados de um dia da base de dados DARPA 1999, que foi convertida para fluxos de dados.

Para a detecção de ataques DDoS, reduzindo o número de falsos positivos Muhai e Ming (LI; LI, 2009) exploraram uma característica do comportamento do tráfego de rede, que se repete através do decorrer do tempo, e aplicaram a transformada Wavelet discreta para diminuir falsos positivos. Após a aplicação da transformada é efetuada a operação de *thresholding* e então é reconstruído o sinal para a detecção de anomalias. Foram utilizados os dados coletados na Universidade de Zaozhang com uma taxa de amostragem de 10 segundos, e a base Wavelet utilizada foi a de Daubechies, com um *threshold* fixo. Os ataques foram disparados pelos autores contra o servidor que estava sendo monitorado.

No trabalho de (KIM; REDDY, 2008) é aplicada a transformada Wavelet discreta e então o sinal é reconstruído somente com a informação dos coeficientes Wavelets. Após os dados serem reconstruídos pela transformada Wavelet inversa é calculada a correlação entre o fluxo de dados, e as portas para detectar anomalias no sinal. Foi utilizado para testes uma base de dados coletada na Universidade da Carolina do Sul. O processo de reconstrução do sinal através da transformada Wavelet inversa requer um processamento adicional, o que não ocorre no algoritmo proposto por este trabalho, pois a análise de alarmes é feita diretamente nos coeficientes Wavelets.

Em (HUANG; THAREJA; SHIN, 2006) é proposto um algoritmo baseado em um projeto *open source* de processamento de sinais, conhecido como LastWave, para analisar ataques da base de dados DARPA 1999 em diferentes famílias Wavelets. O mecanismo de detecção é baseado em duas métricas: proporção de desvio, que compara e verifica as características da análise, e entropia, que mede informações sobre sinais desarranjados. Foi utilizado para testes neste trabalho a base de dados DARPA 1999 e a base de dados EnetRegistry (HUANG; THAREJA; SHIN, 2006).

É proposto por (DALMAZO et al., 2009) um algoritmo para detecção de anomalias baseado

em séries temporais e Wavelets. Após a execução da série temporal os resultados são submetidos para a transformada wavelet visando diminuir o número de falsos positivos, e melhorar a taxa de acertos médios. Foi utilizado nos testes a base de dados DARPA 1999. O algoritmo proposto neste trabalho difere de (DALMAZO et al., 2009), pois a transformada Wavelet é aplicada diretamente nas variáveis de rede, não sendo necessário o processamento através de séries temporais no tráfego de rede.

O algoritmo proposto nesta dissertação difere dos trabalhos relacionados por propor uma abordagem baseada somente na transformada Wavelet 2D, sem a necessidade de reconstrução dos níveis. Outra diferença importante é que a ocorrência de ataques é verificada nas 3 subbandas de detalhes geradas pela aplicação da transformada bidimensional. Não é necessário a aplicação de nenhum algoritmo adicional para verificar a relação entre as diferentes variáveis disponíveis na rede, pois a transformada Wavelet 2D analisa essas informações ao ser aplicada na matriz dos dados de entrada. Assim a aplicação da transformada Wavelet nas colunas e linhas da matriz possibilita o reconhecimento de variações contidas no sinal e que são inerentes às diferentes variáveis do tráfego de rede.

4.4 Considerações Finais

Para a detecção de ataques DoS é proposto um algoritmo e o sistema para detecção de intrusão em redes. O algoritmo de detecção utiliza a transformada Wavelet 2D para analisar os descritores de rede e faz o uso de duas estratégias para a detecção de valores de corte: adaptativa e recursiva.

Com a aplicação da transformada wavelet 2D em diferentes descritores de rede são analisados os dados de cada protocolo, e a relação que existe entre os protocolos. Diferentemente do que ocorre nos trabalhos relacionados onde somente um descritor é analisado por vez. Podem ser utilizadas diversas famílias wavelets para a detecção dos ataques: *Haar*, *Daubechies* Db2, Db4 e Db8.

O processo de geração de alarmes é realizada através de duas técnicas de corte para os coeficientes wavelets: adaptativa e recursiva, podendo os coeficientes wavelets serem normalizados ou não.

Foi desenvolvido um *framework* para a detecção de anomalias, sendo implementado um sistema de detecção de intrusão baseado nele. Este *framework* utilizou *design patterns* para permitir uma modularização do módulo de coleta, detecção e relatórios. Na inicialização do

NIDS devem ser definidos diversos parâmetros, como: família wavelet, estratégia de corte, utilizado na operação de corte, intervalo de amostragem ($\triangle t$) e tamanho da janela deslizante.

No Capítulo 5 são apresentados os experimentos realizados com o algoritmo proposto em diferentes bases de dados e configurações.

5 EXPERIMENTOS

Neste capítulo são apresentados e discutidos os experimentos utilizados, para avaliar o sistema de detecção de ataques DoS baseado em Wavelet bidimensional.

Nos experimentos o algoritmo proposto é avaliado em diversos cenários, utilizando dados sintéticos e dados reais. Os experimentos visam validar a eficácia do algoritmo.

Na Seção 5.1 são apresentadas duas bases de dados utilizadas, expondo as características e preparação das mesmas para teste do sistema de detecção de intrusão de redes. Na Seção 5.2 são apresentados os experimentos realizados para validar o algoritmo de detecção de ataques DoS e os resultados. Por fim na Seção 5.3 são efetuadas as considerações finais e discutidos os trabalhos relacionados.

5.1 Bases de Dados

Para a realização dos experimentos que visam avaliar o algoritmo proposto neste trabalho, duas bases de dados foram utilizadas nos experimentos: base de dados sintética DARPA (1999) e uma base de dados coletada através de dados provenientes do tráfego de redes da Universidade Federal de Santa Maria (UFSM).

Esta seção está organizada da seguinte forma: na Subseção 5.1.1 são apresentados detalhes da base de dados da DARPA, e na Seção 5.1.2 são apresentados detalhes da base de dados capturada na UFSM.

5.1.1 Base de dados DARPA

A base de dados gerada pelo laboratório Lincoln do MIT (*Massachussetts Institute of Technology*) em conjunto com a DARPA (*Defense Advanced Research Projects Agency*) contém um conjunto de cinco semanas de tráfego de rede. Os dados desta base de dados foram gerados em uma rede controlada, com ataques gerados por *scripts* que implementam ataques conhecidos.

Foram capturados os dados de rede no período das 8 horas da manhã até as 6 horas da manhã do outro dia, contendo portanto 22 horas de tráfego capturado em cada dia. Foram coletados os dados durante cinco semanas, onde cada semana contém somente 5 dias de tráfego registrados. Os dados foram armazenados no formato de dados da ferramenta *tcpdump*[1], contendo aproximadamente 9 GB de dados.

[1] http://www.tcpdump.org

As três primeiras semanas de tráfego são destinadas a treinamento, pois possuem documentação do ataques existentes no período. A primeira e terceira semana não possuem ataques, sendo descartadas, pois neste trabalho não é necessário o processo de treinamento do algoritmo. A segunda semana de tráfego é utilizada na análise de eficácia do algoritmo proposto neste trabalho. A Tabela 5.1 apresenta a documentação dos ataques, contendo o identificador do ataque, a data e hora do início do ataque, o *host* de qual se originou o ataque e o nome do ataque. Em negrito estão os ataques DoS analisados nos experimentos. As duas últimas semanas de dados possuem diversos ataques não identificados e não documentados, o que não permite comparar e avaliar a eficácia do algoritmo.

Os dados disponibilizados pelo MIT[2] contém todas as informações dos pacotes, incluindo *headers* e *payload*. Neste trabalho são somente considerados as informações contidas no *header* dos pacotes, descartando qualquer informação presente no *payload*. Com o uso desta política o esforço computacional necessário para o processamento dos pacotes diminuí consideravelmente, e também permite manter a privacidade dos dados presentes no tráfego dos usuários.

A segunda semana da base de dados DARPA contém 43 ataques identificados (vide Tabela 5.1), sendo considerados para este trabalho somente os ataques de negação de serviço (8). Nesta base de dados existem 11 ataques do tipo DoS, porém somente 8 são identificáveis sem analisar o *payload* dos pacotes.

A Figura 5.1 mostra o gráfico do descritor IP do tráfego de rede da base de dados DARPA. São representadas 22 horas (79200 segundos) do tráfego de rede, onde não ocorre nenhum ataque. Este gráfico representa os pacotes capturados na primeira semana, de segunda-feira das 8 horas da manhã até as 6 horas da manhã de terça-feira. O número de pacotes durante o decorrer do dia varia de aproximadamente 40 a 200 pacotes por segundo, representando o comportamento de uso na época em que a base de dados foi construída.

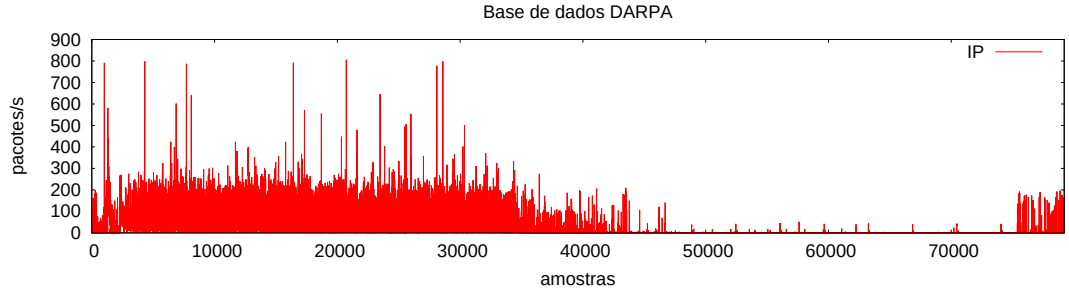

Figura 5.1: Comportamento do tráfego IP de um dia da base de dados DARPA

[2]http://www.ll.mit.edu/mission/communications/ist/corpora/ideval/data/1999data.html

Tabela 5.1: Lista com ataques presentes na segunda semana na base de dados DARPA. Fonte (DARPA, 1999).

ID	Data	Tempo	Origem	Nome
1	08/03/1999	08:01:01	hume.eyrie.af.mil	NTinfoscan
2	08/03/1999	08:50:15	zeno.eyrie.af.mil	**pod**
3	08/03/1999	09:39:16	marx.eyrie.af.mil	**back**
4	08/03/1999	12:09:18	pascal.eyrie.af.mil	httptunnel
5	08/03/1999	15:57:15	pascal.eyrie.af.mil	land
6	08/03/1999	17:27:13	marx.eyrie.af.mil	secret
7	08/03/1999	19:09:17	pascal.eyrie.af.mil	ps attack
8	09/03/1999	08:44:17	marx.eyrie.af.mil	portsweep
9	09/03/1999	09:43:51	pascal.eyrie.af.mil	eject
10	09/03/1999	10:06:43	marx.eyrie.af.mil	**back**
11	09/03/1999	10:54:19	zeno.eyrie.af.mil	loadmodule
12	09/03/1999	11:49:13	pascal.eyrie.af.mil	secret
13	09/03/1999	14:25:16	pascal.eyrie.af.mil	**mailbomb**
14	09/03/1999	13:05:10	172.016.112.001-114.254	ipsweep
15	09/03/1999	16:11:15	marx.eyrie.af.mil	phf
16	09/03/1999	18:06:17	pascal.eyrie.af.mil	httptunnel
17	10/03/1999	12:02:13	marx.eyrie.af.mil	satan
18	10/03/1999	13:44:18	pascal.eyrie.af.mil	**mailbomb**
19	10/03/1999	15:25:18	marx.eyrie.af.mil	perl (Failed)
20	10/03/1999	20:17:10	172.016.112.001-114.254	ipsweep
21	10/03/1999	23:23:00	pascal.eyrie.af.mil	eject (console)
22	10/03/1999	23:56:14	hume.eyrie.af.mil	**crashiis**
23	11/03/1999	08:04:17	hume.eyrie.af.mil	**crashiis**
24	11/03/1999	09:33:17	marx.eyrie.af.mil	satan
25	11/03/1999	10:50:11	marx.eyrie.af.mil	portsweep
26	11/03/1999	11:04:16	pigeon.eyrie.af.mil	**neptune**
27	11/03/1999	12:57:13	marx.eyrie.af.mil	secret
28	11/03/1999	14:25:17	marx.eyrie.af.mil	perl
29	11/03/1999	15:47:15	pascal.eyrie.af.mil	land
30	11/03/1999	16:36:10	172.016.112.001-254	ipsweep
31	11/03/1999	19:16:18	pascal.eyrie.af.mil	ftp-write
32	12/03/1999	08:07:17	marx.eyrie.af.mil	phf
33	12/03/1999	08:10:40	marx.eyrie.af.mil	perl (console)
34	12/03/1999	08:16:46	pascal.eyrie.af.mil	ps (console)
35	12/03/1999	09:18:15	duck.eyrie.af.mil	**pod**
36	12/03/1999	11:20:15	marx.eyrie.af.mil	**neptune**
37	12/03/1999	12:40:12	hume.eyrie.af.mil	**crashiis**
38	12/03/1999	13:12:17	zeno.eyrie.af.mil	loadmodule
39	12/03/1999	14:06:17	marx.eyrie.af.mil	perl (Failed)
40	12/03/1999	14:24:18	pascal.eyrie.af.mil	ps
41	12/03/1999	15:24:16	pascal.eyrie.af.mil	eject
42	12/03/1999	17:13:10	pascal.eyrie.af.mil	portsweep
43	12/03/1999	17:43:18	pascal.eyrie.af.mil	ftp-write

Para permitir a análise através do sistema de detecção de intrusão, os dados da base de dados DARPA foram pré-processados, pois encontram-se no formato do *tcpdump*. Para a análise foram coletadas informações da base de dados a cada 1 segundo, nos descritores: IP, TCP, ICMP, UDP, Não IP (ARP, BGP), e tamanho do *payload*. Foi escolhido o período de amostragem de 1 segundo, pois alguns ataques possuem curta duração, como por exemplo o *PoD*.

Foram considerados os seguintes tipos de ataques DoS: *mailbomb*, *back*, *PoD* e *neptune*, descritos na Seção 2.1. A Figura 5.2 mostra os ataques presentes na base de dados DARPA, com o retângulo tracejado em azul demarcando o momento do ataque. Note que nos gráficos (a), (c) e (d) houve um aumento no número de pacotes, variando de segundos a minutos dependendo do ataque.

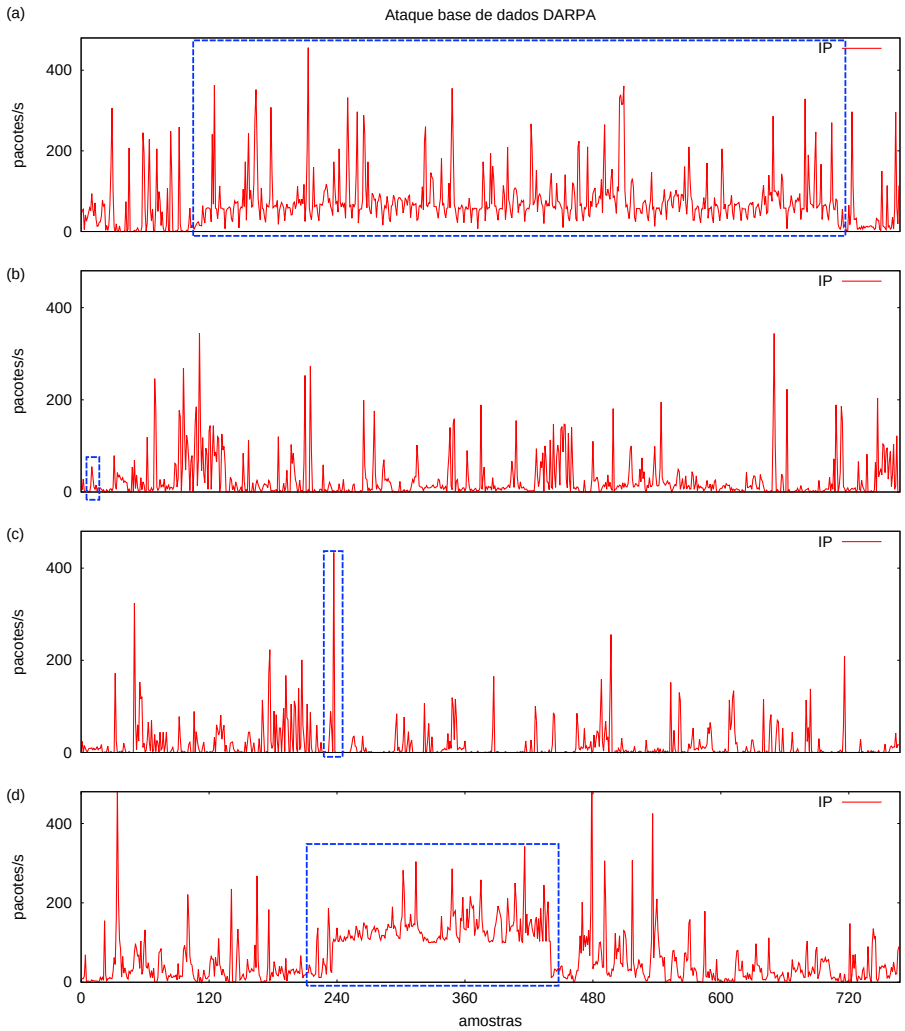

Figura 5.2: Ataques DoS presentes na base de dados DARPA: (a) *mailbomb*, (b) *back*, (c) *PoD*, e (d) *neptune*

Os dados disponibilizados pelo MIT e DARPA foram processados através da ferramenta

tcpstat[3] e armazenados em um arquivo que contém os dados. A Tabela 5.2 mostra os dados presentes no início da amostra representada no gráfico da Figura 5.1, que demonstra um dia da base de dados DARPA e contém além dos contadores dos protocolos IP, TCP, ICMP, UDP e Não IP (ARP, BGP), contém o somatório do tamanho do *payload* de todos os pacotes naquele *TimeStamp* e o instante de tempo em que foram capturados os pacotes (*TimeStamp*). Foram utilizados estes descritores pois os ataques de negação de serviço perturbam pelo menos um dos descritores utilizados.

Tabela 5.2: Exemplo do arquivo da base de dados DARPA, que contém os dados necessários para os testes do algoritmo de detecção de intrusão.

Timestamp	IP	TCP	ICMP	UDP	Não IP	Tam. *Payload*
920988846	8	8	0	0	0	39
920988847	100	98	2	0	0	479
920988848	20	19	1	0	0	88
920988849	6	6	0	0	0	21
920988850	5	5	0	0	0	23
920988851	77	74	2	1	0	275
...	...	...	...	...	...	...

5.1.2 Base de dados da UFSM

Para critério de comparação do algoritmo com uma base de dados atualizada e que contenha tráfego de dados reais, foi coletado uma base de dados na Universidade Federal de Santa Maria (UFSM). Este conjunto de dados foi coletado através de um *sniffer* que recebe todo o tráfego de rede espelhado.

O *sniffer* foi desenvolvido como um módulo de coleta do sistema de detecção de intrusão proposto neste trabalho, e captura todas as informações disponíveis nos cabeçalhos dos pacotes, conforme descrito na Subseção 4.2.1. Foram criados contadores com uma taxa de atualização de 1 segundo para permitir uma análise equivalente a da base de dados disponibilizada pelo MIT e DARPA. Os descritores escolhidos para análise foram: IP, TCP, ICMP, UDP, Não IP e tamanho do *payload*, também equivalente aos disponíveis na base de dados DARPA 99.

Foram capturadas informações somente dos cabeçalhos devido a privacidade dos usuários, e ao alto custo computacional para analisar o *payload* de todos os pacotes. A Figura 5.3 apresenta a arquitetura da rede utilizada pela instituição de ensino. O *sniffer* foi instalado em um servidor que recebe os dados e este conectado a uma porta do *switch* que clona todo o tráfego presente

[3]Disponível em: http://www.frenchfries.net/paul/tcpstat/

no ativo de rede.

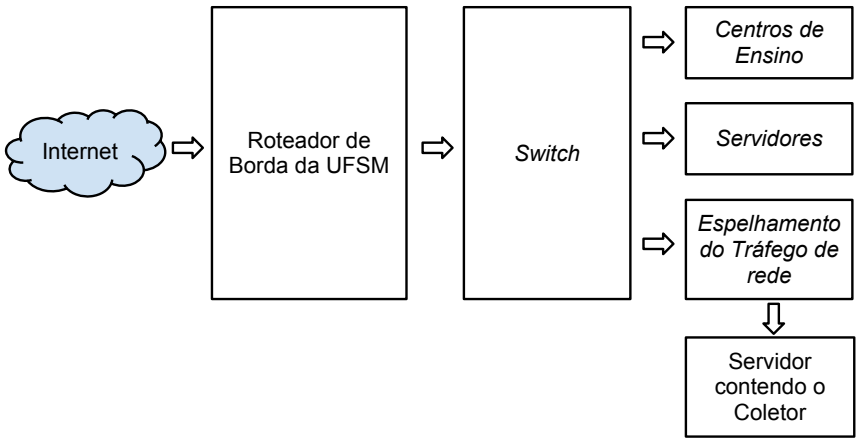

Figura 5.3: Arquitetura da Rede da Universidade Federal de Santa Maria.

A base de dados da DARPA apesar de servir como base de diferentes testes e estudos pela comunidade de pesquisa de sistemas de detecção de intrusão (LU; TAVALLAEE; GHORBANI, 2008) é uma base de dados criticada por não possuir uma metodologia adequada (MCHUGH, 2000) e que não equivale ao ambiente atual de uso (LU; TAVALLAEE; GHORBANI, 2008). A base de dados da UFSM foi capturada para suprir a necessidade de um conjunto de dados que contenha dados atuais e reais.

Foi capturado o tráfego de rede da primeira semana do mês de outubro de 2010. Foram analisadas as notificações recebidas através da RNP (Rede Nacional de Ensino e Pesquisa)[4] sobre incidentes de segurança que envolveram a rede da UFSM. Além disto foi efetuada uma análise manual de todo o tráfego de rede da semana utilizada nos experimentos, não existindo anomalias nos dados capturados.

Para permitir a análise de desempenho do algoritmo neste conjunto de dados foram inseridos 20 ataques DoS, modelados através da base de dados DARPA, e de ferramentas de ataque, como: TFN (*Tribe Flood Network*) (DITTRICH, 1999), mstream (DITTRICH et al., 2000) e hping (SANFILIPPO, 2011). Estas ferramentas são utilizados para ataques DoS e DDoS do tipo de esgotamento de largura de banda (PROLEXIC, 2011). Os ataques foram inseridos em posições randômicas na base de dados, não ocorrendo sobreposições dos ataques. Os testes efetuados com as ferramentas de ataques foram realizados em um ambiente controlado em um laboratório de pesquisa, sendo posteriormente adicionados no traço original da base de dados da UFSM.

[4]http://www.rnp.br

A Figura 5.4 exibe o gráfico de 24 horas (86400 segundos) do descritor de rede IP no tráfego de rede da base de dados da UFSM. O número de pacotes é substancialmente maior que na base de dados da DARPA (Figura 5.1), demonstrando a diferença no número de pacotes por segundo, já criticada por (MCHUGH, 2000). A diferença de pacotes por segundo entre as duas bases de dados é da ordem de 100 vezes, confirmando a mudança de comportamento que ocorreu durante o intervalo de captura dos dados da DARPA e da UFSM.

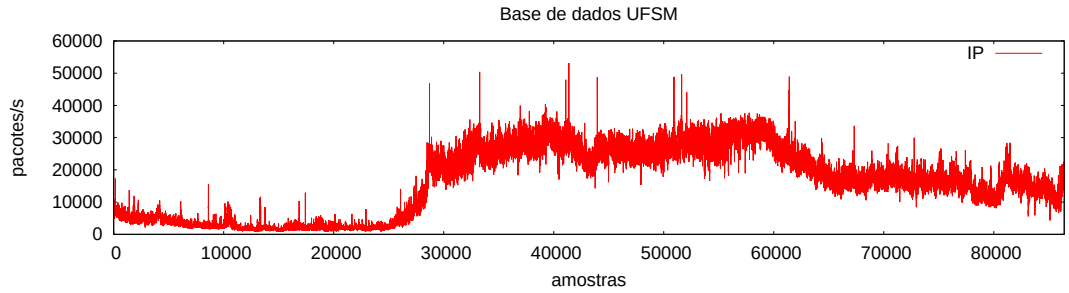

Figura 5.4: Comportamento do descritor IP no tráfego de rede da UFSM

Os dados coletados foram armazenados em um arquivo que contém as informações do tráfego de rede. A Tabela 5.3 exibe o início de um arquivo extraído da base de dados da UFSM (representado na Figura 5.4), que contém além dos contadores dos protocolos IP, TCP, ICMP, UDP e Não IP, o tamanho do *payload* e o instante de tempo em que foi capturado o pacote (*TimeStamp*).

Tabela 5.3: Exemplo do arquivo da base de dados da UFSM, que contém os dados necessários para os testes do algoritmo de detecção de intrusão.

TimeStamp	IP	TCP	ICMP	UDP	Não IP	Tam. *Payload*
1286161203	6998	6837	22	139	0	4323456
1286161204	6606	6485	6	115	0	3916607
1286161205	6450	6254	25	171	0	3592681
1286161206	6430	6176	19	234	1	3626368
1286161207	7437	7270	16	141	10	4359140
1286161208	7686	7451	80	150	5	4112007
...	...	...	...	...	...	...

5.2 Experimentos

Para avaliar a capacidade de detecção de ataques DoS do algoritmo proposto, foram realizados testes com o NIDS proposto na Seção 4.2. NIDS baseados em anomalias são capazes de detectar ataques conhecidos, assim como mutações e ataques desconhecidos. Para fins de

análise neste trabalho é avaliada a capacidade do NIDS detectar ataques conhecidos e documentados das bases de dados. Foram efetuados testes para validar o algoritmo de detecção de intrusão, onde foram utilizados os descritores disponíveis nas duas bases de dados discutidas nas Subseções 5.1.1, e 5.1.2.

Nesta seção são apresentados os experimentos realizados com o algoritmo de detecção de ataques DoS baseado em wavelet 2D. Para permitir a avaliação da capacidade de detecção foram utilizadas duas semanas de dados: uma da DARPA, e uma da UFSM, ambas contendo a localização de ataques de negação de serviço. A localização dos ataques se faz necessária para a correta avaliação do desempenho do algoritmo testado.

Na Subseção 5.2.1 é realizada uma bateria de testes utilizando diferentes famílias wavelets e a estratégia de corte recursiva na base de dados da DARPA. A estratégia de corte adaptativa, também avaliada utilizando diferentes famílias wavelets, é analisada nas subseções 5.2.2 e 5.2.3 utilizando as bases de dados DARPA e UFSM, respectivamente.

5.2.1 Estudo de Caso 1: DARPA com operação de corte recursiva

Neste experimento são utilizados os dados da base de dados DARPA e são executados testes utilizando as famílias wavelets de Haar e Daubechies (Db2, Db4 e Db8) com a estratégia de corte recursiva. Foram realizados testes com e sem a normalização dos coeficientes wavelets através da função raiz quadrada.

Para avaliar a detecção de ataques DoS foram utilizados os protocolos de rede que são perturbados na ocorrência de um ataque de negação de serviço. Os descritores perturbados por um ataque DoS na base de dados DARPA são: IP, TCP, ICMP, e UDP. A taxa de amostragem utilizada nos testes foi de 1 segundo, pois dentre o conjunto de ataques avaliados está o PoD, que possuí uma pequena duração.

Para tratar do problema de extensão das fronteiras da família wavelet de Daubechies Db2, Db4 e Db8, foi utilizada a repetição do último valor. Este problema não precisa ser tratado com a wavelet de Haar pois a mesma não possui problema de fronteira.

A Figura 5.5(a) mostra um recorte do sinal original formado pelos descritores de rede utilizados durante a ocorrência de um ataque do tipo *PoD*, com uma taxa de amostragem de 1 segundo. Os gráficos presentes na Figura 5.5(b),(c),(d) e (e) apresentam os coeficientes wavelets (detalhes) gerados pelas sub-bandas de detalhes (todas as sub-bandas juntas, médias dos detalhes, detalhes das médias e detalhes dos detalhes respectivamente) não normalizados, e os

valores de corte (τ) para cada sub-banda. Caso seja detectado um coeficiente maior que o valor de corte (τ) é gerado um alarme.

Conforme mostra a Figura 5.5(b),(c),(d) e (e) o ataque é detectado pela wavelet de Haar em todos os coeficientes de detalhes, conforme mostra o retângulo laranja. Também foram detectados pontos onde não existem ataques (falso positivo) e sim pequenas perturbações no tráfego de rede, enfatizados pelos retângulos verdes.

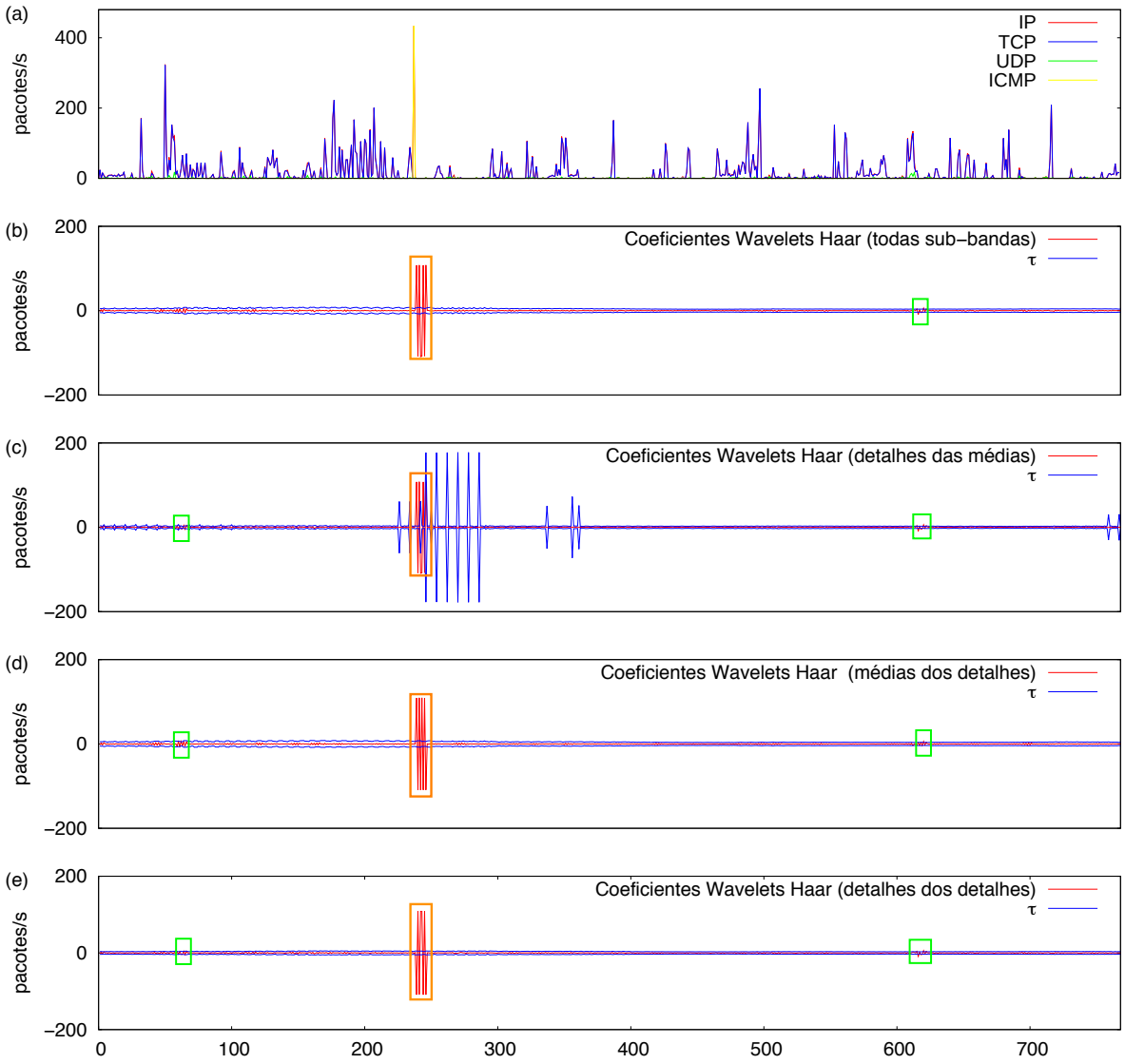

Figura 5.5: Uso da família wavelet de Haar e estratégia recursiva nos coeficientes wavelets durante um ataque PoD (a). (b) Todas as sub-bandas de coeficientes de detalhes. (c) Sub-banda médias dos detalhes dos coeficientes de detalhes. (d) Sub-banda detalhes das médias dos coeficientes de detalhes. (e) Sub-banda detalhes dos detalhes dos coeficientes de detalhes.

Os gráficos da Figura 5.6(b),(c),(d) e (e) exibem o comportamento da estratégia de corte recursiva utilizando a família wavelet de Daubechies Db2, durante um ataque *PoD*. As escalas

dos gráficos são ajustadas para apresentar de forma mais clara possível os gráficos, por este motivo as escalas variam nos diferentes testes. O ataque *PoD* é detectado corretamente pela estratégia de corte recursiva somente quando é levado em consideração todas as sub-bandas de detalhes, conforme mostra a Figura 5.6(b) no instante 250. Quando é utilizado somente os valores de cada sub-banda em separado o valor de corte é próximo ou igual a zero, detectando todos os pontos como sendo ataques, conforme mostram os gráficos da Figura 5.6(c), (d) e (e).

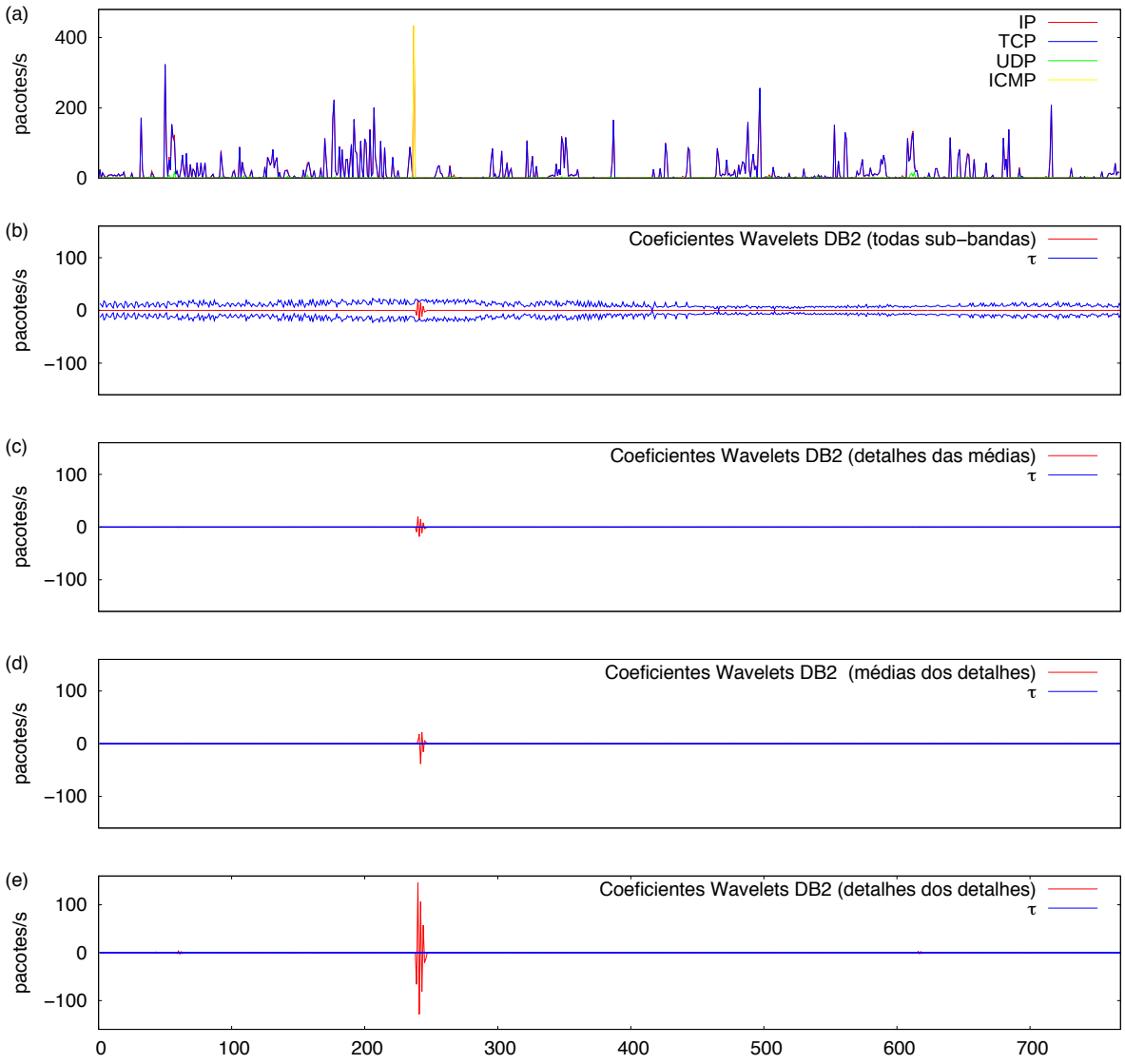

Figura 5.6: Uso da família wavelet de daubechies Db2 e estratégia recursiva nos coeficientes wavelets durante um ataque PoD. (a) Descritores de Rede. (b) Todas as sub-bandas de coeficientes de detalhes. (c) Sub-banda médias dos detalhes dos coeficientes de detalhes. (d) Sub-banda detalhes das médias dos coeficientes de detalhes. (e) Sub-banda detalhes dos detalhes dos coeficientes de detalhes.

Os testes com a família wavelet de *Daubechies Db4*, e *Daubechies Db8* apresentam as mesmas características do uso da wavelet de *Daubechies Db2*. Por este motivo os gráficos não

são apresentados.

Nas Figuras 5.7 e 5.8 são apresentados os gráficos utilizando a estratégia recursiva para corte, mas com a normalização dos coeficientes wavelets pelo uso da função raiz quadrada.

Nos gráficos da Figura 5.7(b), (d) e (e) que exibem os coeficientes wavelets da wavelet de Haar normalizados, o ataque é detectado corretamente no instante 250. Nestes gráficos não são detectados falsos positivos, diferentemente do teste mostrado na Figura 5.5 onde não são normalizados os coeficientes wavelets. Na Figura 5.7(c) é apresentado o gráfico dos coeficientes wavelets que representam os detalhes das médias e o ataque não é detectado, pois o parâmetro de corte τ é sempre maior que os coeficientes wavelets.

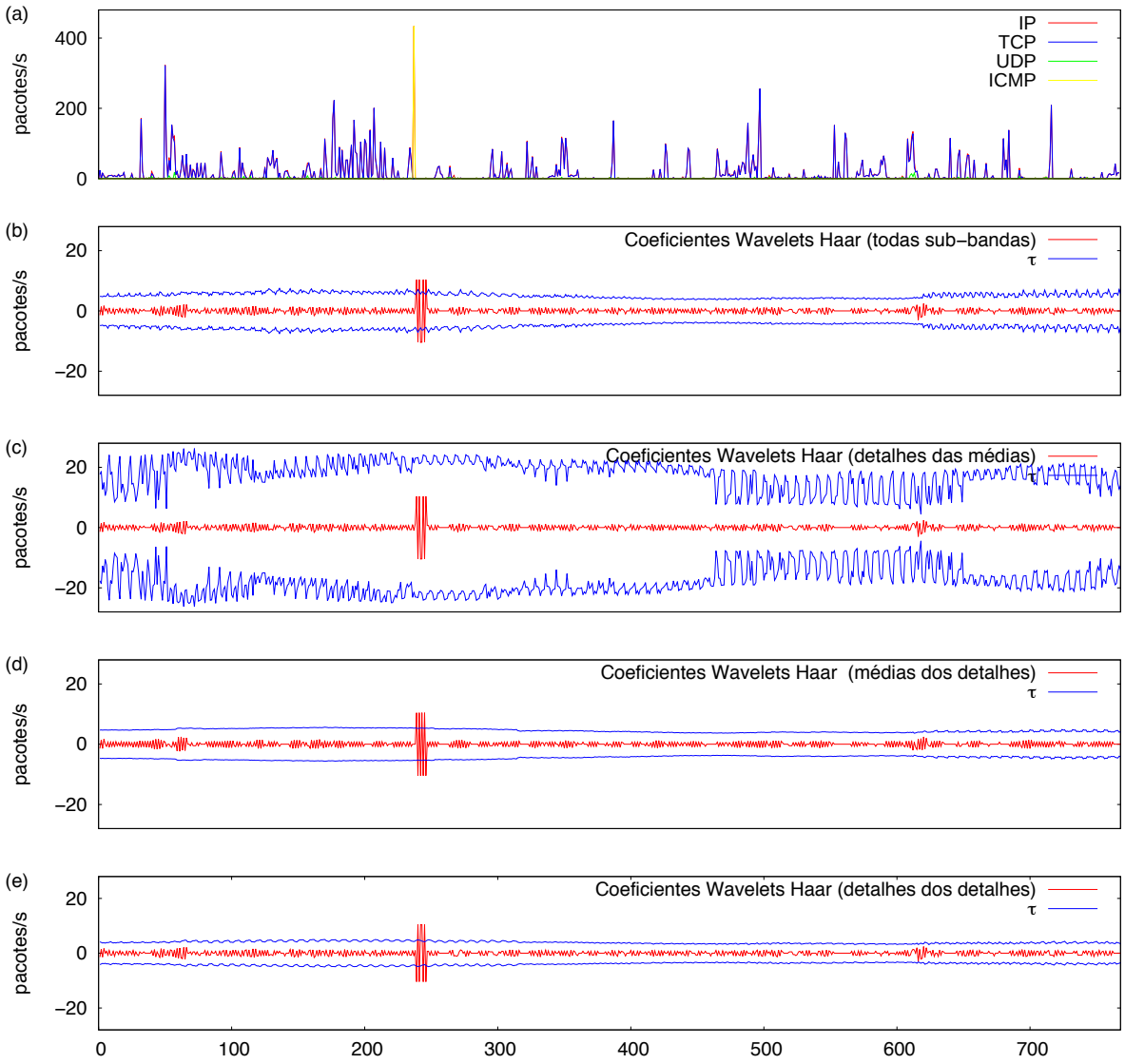

Figura 5.7: Uso da família wavelet de Haar e estratégia recursiva nos coeficientes wavelets normalizados durante um ataque PoD. (a) Gráfico do uso da rede e seus descritores. (b) Todas as sub-bandas de coeficientes de detalhes. (c) Sub-banda médias dos detalhes dos coeficientes de detalhes. (d) Sub-banda detalhes das médias dos coeficientes de detalhes. (e) Sub-banda detalhes dos detalhes dos coeficientes de detalhes.

Os gráficos dos coeficientes wavelets normalizados da wavelet de Daubechies Db2 representados na Figura 5.8 apresentam o mesmo problema de quando não se utiliza a normalização, isto é, o valor de corte τ tende a ficar igual ou próximo de zero. O benefício obtido com a normalização é a menor variabilidade dos coeficientes wavelets.

A transformada wavelet 2D de Daubechies Db4 e Db8 utilizando a normalização dos coeficientes wavelets apresenta o mesmo comportamento da Daubechies Db2.

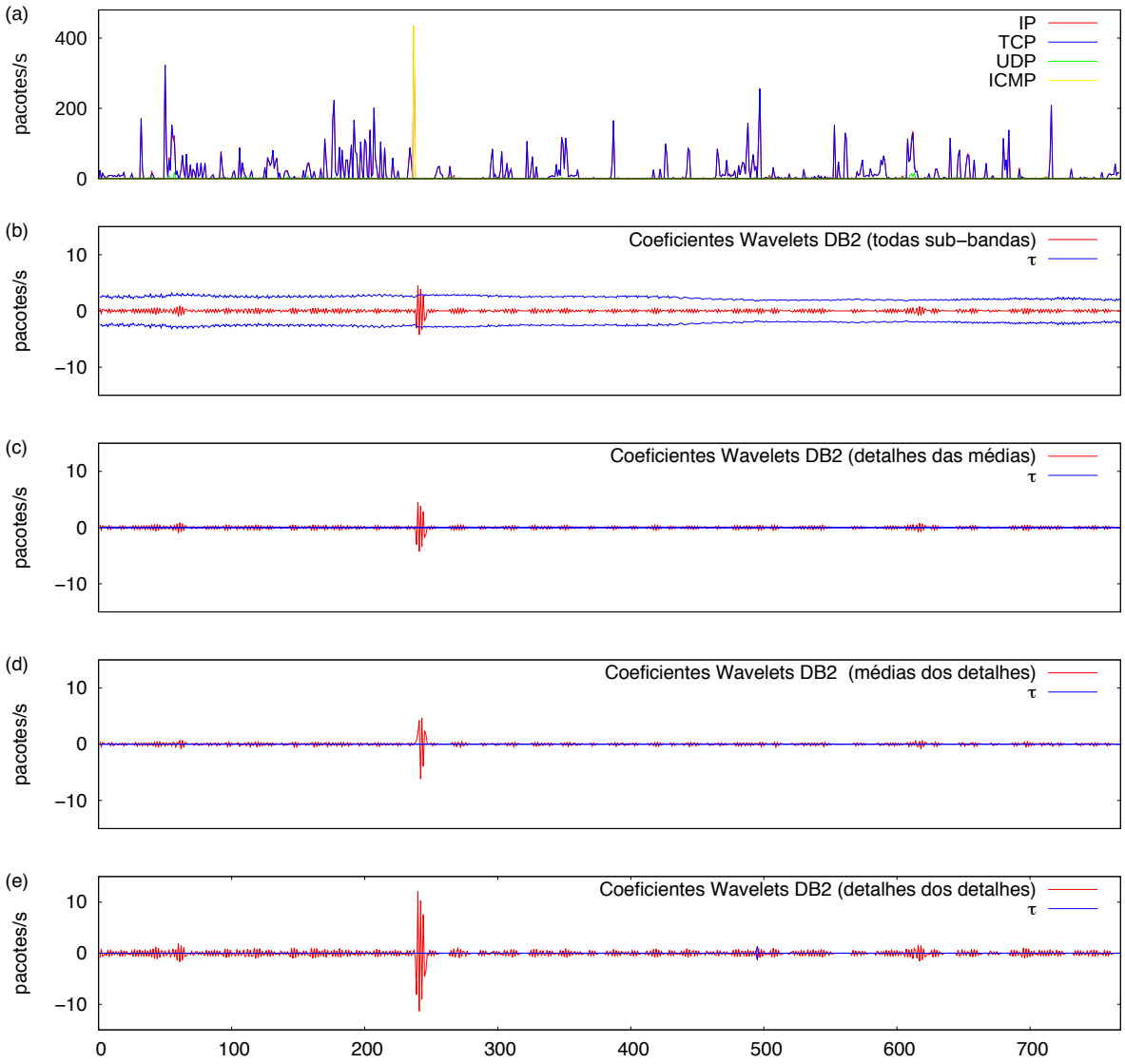

Figura 5.8: Uso da família wavelet de daubechies Db2 e estratégia recursiva nos coeficientes wavelets normalizados durante um ataque PoD. (a) Gráfico do uso da rede e seus descritores. (b) Todas as sub-bandas de coeficientes de detalhes. (c) Sub-banda médias dos detalhes dos coeficientes de detalhes. (d) Sub-banda detalhes das médias dos coeficientes de detalhes. (e) Sub-banda detalhes dos detalhes dos coeficientes de detalhes.

Para avaliar a capacidade de detecção de ataques DoS, os dados da segunda semana da base de dados DARPA foram submetidos ao sistema de detecção de intrusão. Para avaliar o algoritmo foram efetuados testes com as diferentes famílias wavelets, com e sem normalização dos coeficientes wavelets e variando o tamanho da janela deslizante.

Os gráficos presentes nas Figuras 5.9 e 5.10 exibem os resultados com as métricas de detecção do algoritmo recursivo utilizando as seguintes wavelets: Haar, Daubechies Db2, Daubechies Db4 e Daubechies Db8, e suas versões normalizadas. O gráfico da Figura 5.9 apresenta os verdadeiros positivos (VP), e a Figura 5.10 o número de falsos positivos (FP) encontrados

nos testes.

Conforme mostra o gráfico da Figura 5.9 contendo os resultados dos experimentos, todos os ataques DoS são detectados corretamente, mas isto ocorre pois a maioria das amostras são detectadas como anômalas. Com o uso da transformada wavelet 2D e estratégia de corte recursivo foram detectados a maioria dos pontos como falsos positivos, conforme pode ser visto no gráfico da Figura 5.10. O comportamento é o mesmo com os coeficientes wavelets normalizados ou não.

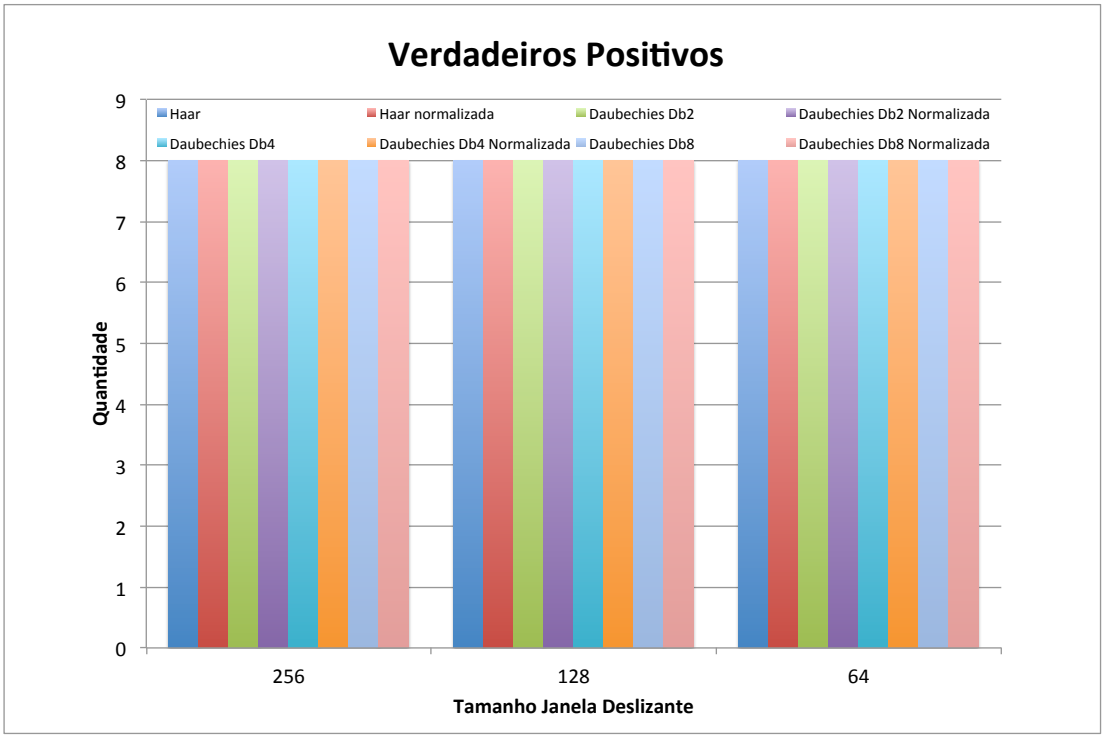

Figura 5.9: Gráfico com os Verdadeiros Positivos

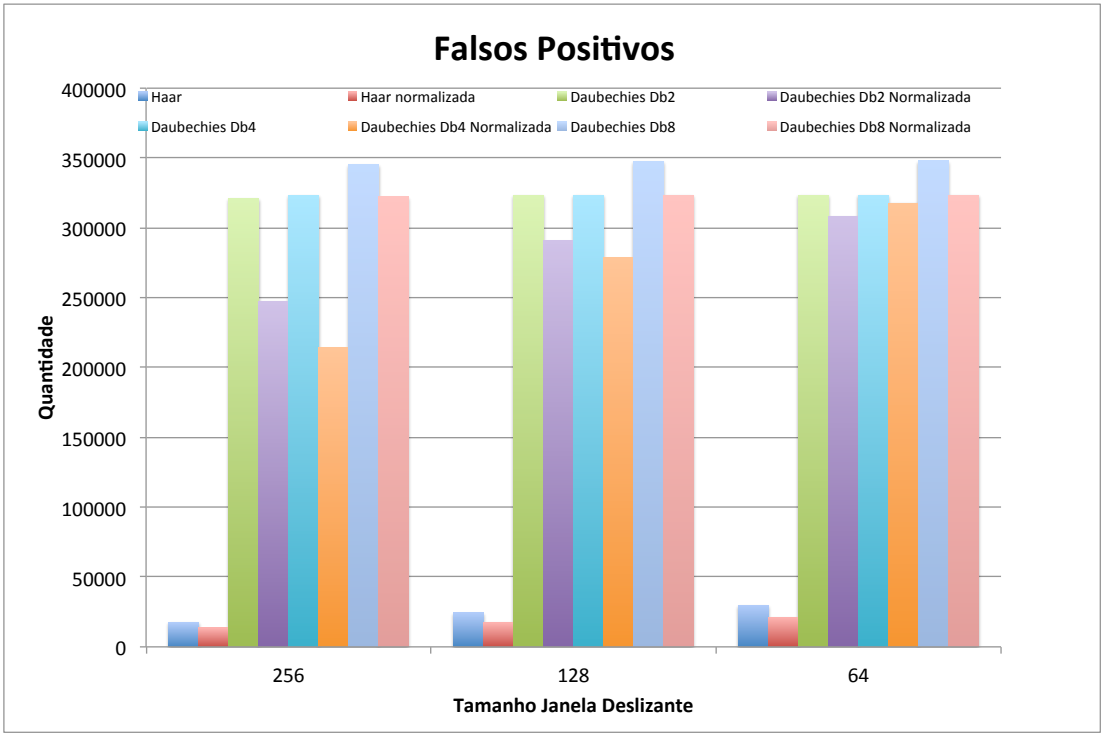

Figura 5.10: Gráfico com os Falsos Positivos

O melhor conjunto de parâmetros obtidos nos testes efetuados nesta subseção foram obtidos com a transformada wavelet de Haar normalizando os coeficientes wavelets, conforme mostra o gráfico da Figura 5.10. Em todos os testes executados pode-se observar que com uma janela deslizante de 256 amostras obteve-se uma menor quantidade de falsos positivos, mas o número de falsos positivos é muito elevado. O número de falsos positivos variou de 13723 quando utilizado a wavelet de *Haar* normalizada com uma janela deslizante de 256 amostras até 29858 falsos positivos quando utilizada a wavelet de *Haar* não normalizada com uma janela deslizante de 64 amostras.

5.2.2 Estudo de Caso 2: DARPA com operação de corte adaptativa

Neste experimento também é utilizada a base de dados DARPA, e são executados testes utilizando as famílias wavelets de Haar e Daubechies (Db2, Db4 e Db8). A estratégia de corte utilizada nestes testes foi a adaptativa, discutida na subseção 4.1.3. As fronteiras das wavelets de Daubechies Db2, Db4 e Db8 foram estendidas pela repetição do último valor. A wavelet de Haar não necessita de tratamento de fronteira.

Os protocolos de rede avaliados são os em que ocorrem perturbações na existência de um ataque de negação de serviço. Os descritores utilizados nestes testes são: IP, TCP, ICMP, e UDP. A taxa de amostragem utilizada nos testes foi de 1 segundo. Os testes foram efetuados

com e sem a normalização dos coeficientes wavelets.

A Figura 5.11(a) mostra o sinal de 768 amostras dos descritores de rede durante um ataque do tipo *PoD*, que ocorre no instante 250, com uma taxa de amostragem de 1 segundo. Conforme mostra a Figura 5.5(b),(c),(d) e (e) o ataque é corretamente detectado pela wavelet de Haar nas sub-bandas médias dos detalhes (d) e detalhes dos detalhes(e), não sendo detectados nas outras sub-bandas. Na sub-banda dos detalhes dos detalhes ele detecta um falso positivo aproximadamente no instante 650 (Figura 5.5(e)), mas como o algoritmo de geração de alarmes somente é disparado quando detecta perturbações em mais de uma sub-banda de coeficientes wavelets, o NIDS não gera um falso positivo.

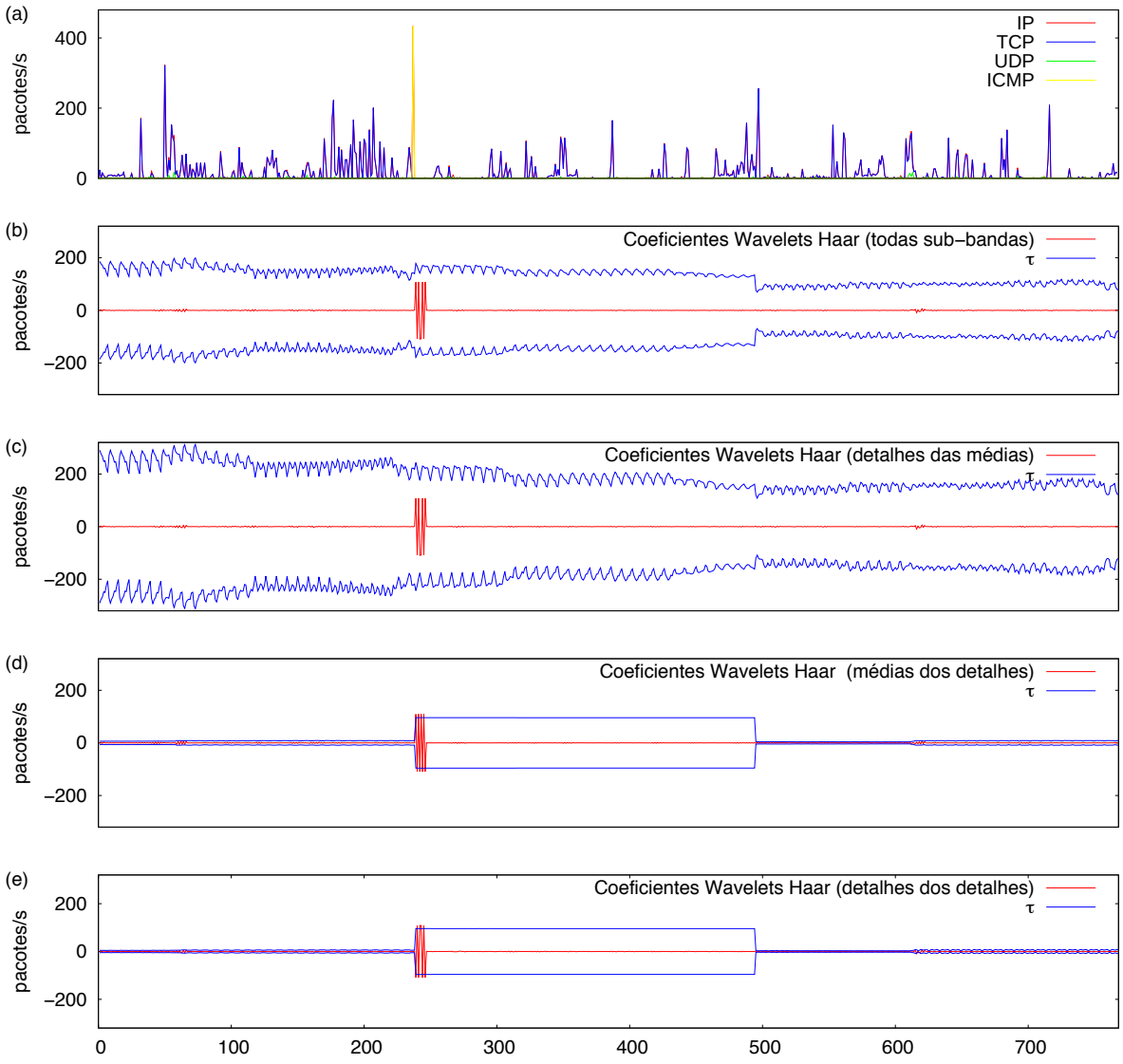

Figura 5.11: Uso da família wavelet de Haar e estratégia adaptativa nos coeficientes wavelets durante um ataque PoD. (a) Gráfico do uso da rede e seus descritores. (b) Todas as sub-bandas de coeficientes de detalhes. (c) Sub-banda médias dos detalhes dos coeficientes de detalhes. (d) Sub-banda detalhes das médias dos coeficientes de detalhes. (e) Sub-banda detalhes dos detalhes dos coeficientes de detalhes.

A Figura 5.12(a) exibe o comportamento dos descritores de rede durante um ataque *PoD*, que ocorre aproximadamente n. O gráfico 5.6(e) apresenta a sub-banda de coeficiente wavelet detalhes dos detalhes, na qual é detectado o ataque. Nas outras sub-bandas de detalhes (Figura 5.6(b), (c), (d)) não é detectado o ataque. Quando é utilizado a estratégia de corte adaptativa com a família wavelet de *Daubechies*, alarmes são gerados quando uma ou mais anomalias são detectadas. A wavelet de Haar detecta pequenas perturbações, e por este motivo são disparados alarmes quando são encontradas anomalias em duas ou mais sub-bandas de detalhes.

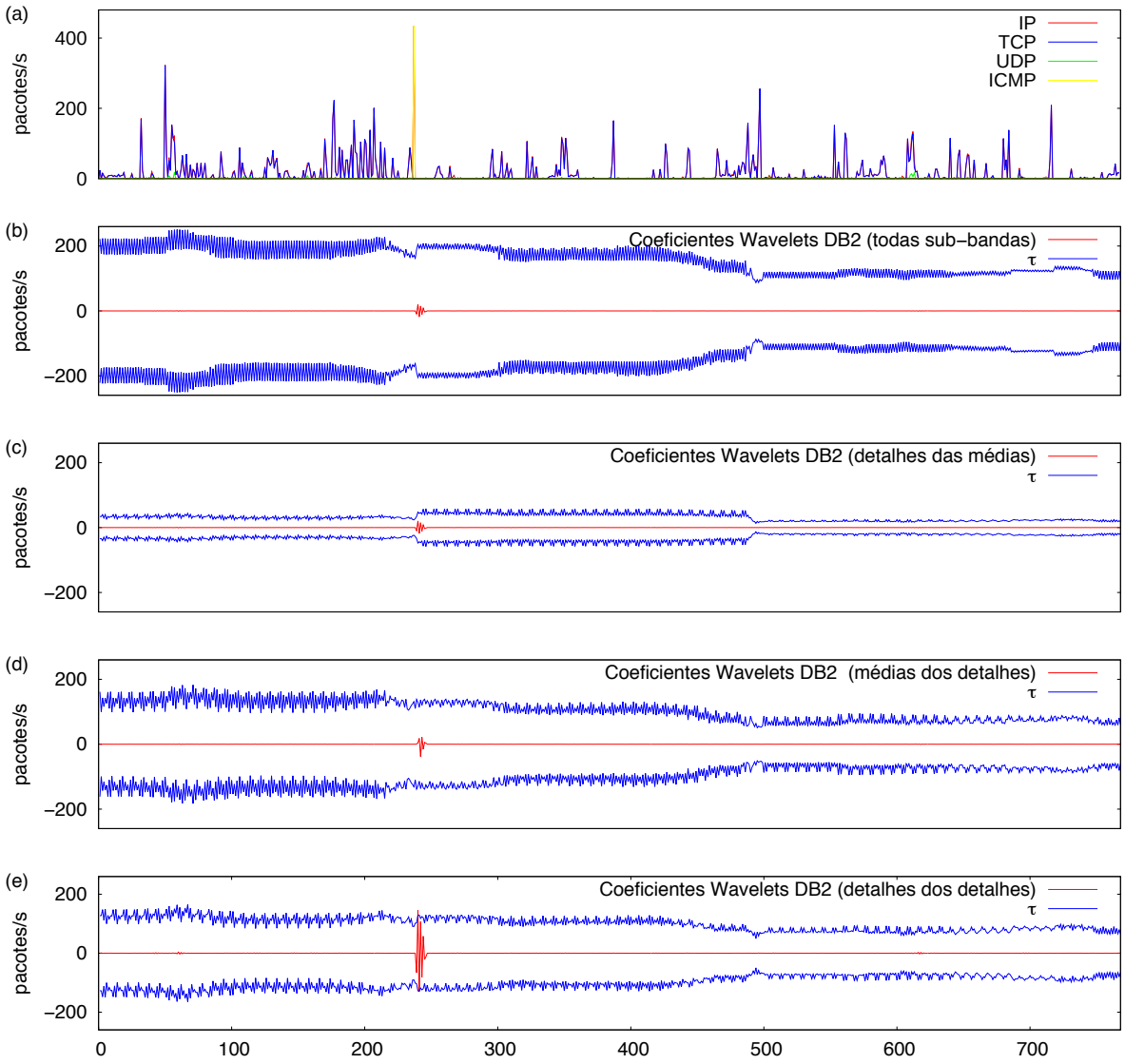

Figura 5.12: Uso da família wavelet de daubechies Db2 e estratégia adaptativa nos coeficientes wavelets durante um ataque PoD. (a) Gráfico do uso da rede e seus descritores. (b) Todas as sub-bandas de coeficientes de detalhes. (c) Sub-banda médias dos detalhes dos coeficientes de detalhes. (d) Sub-banda detalhes das médias dos coeficientes de detalhes. (e) Sub-banda detalhes dos detalhes dos coeficientes de detalhes.

Os gráficos da transformada wavelet de *Daubechies Db4* e *Daubechies Db8* não são apresentados pois o comportamento é similar ao da wavelet de *Daubechies Db2*. Ou seja, o ataque é detectado em pelo menos uma sub-banda de detalhes, sem a ocorrência de falsos positivos.

Nas Figuras 5.13 e 5.14 são apresentados os gráficos utilizando a estratégia adaptativa de corte, com a normalização dos coeficientes wavelets. Com o uso da normalização o ataque *PoD* é identificado nas mesmas sub-bandas em que ocorreu a detecção sem a normalização. Em outros tipos de ataques DoS a normalização melhora a eficácia do algoritmo, conforme mostram os gráficos das Figuras 5.15 e 5.16 que apresentam as métricas (VP e FP, respectivamente) do

algoritmo adaptativo utilizando as famílias wavelets com e sem normalização dos coeficientes wavelets.

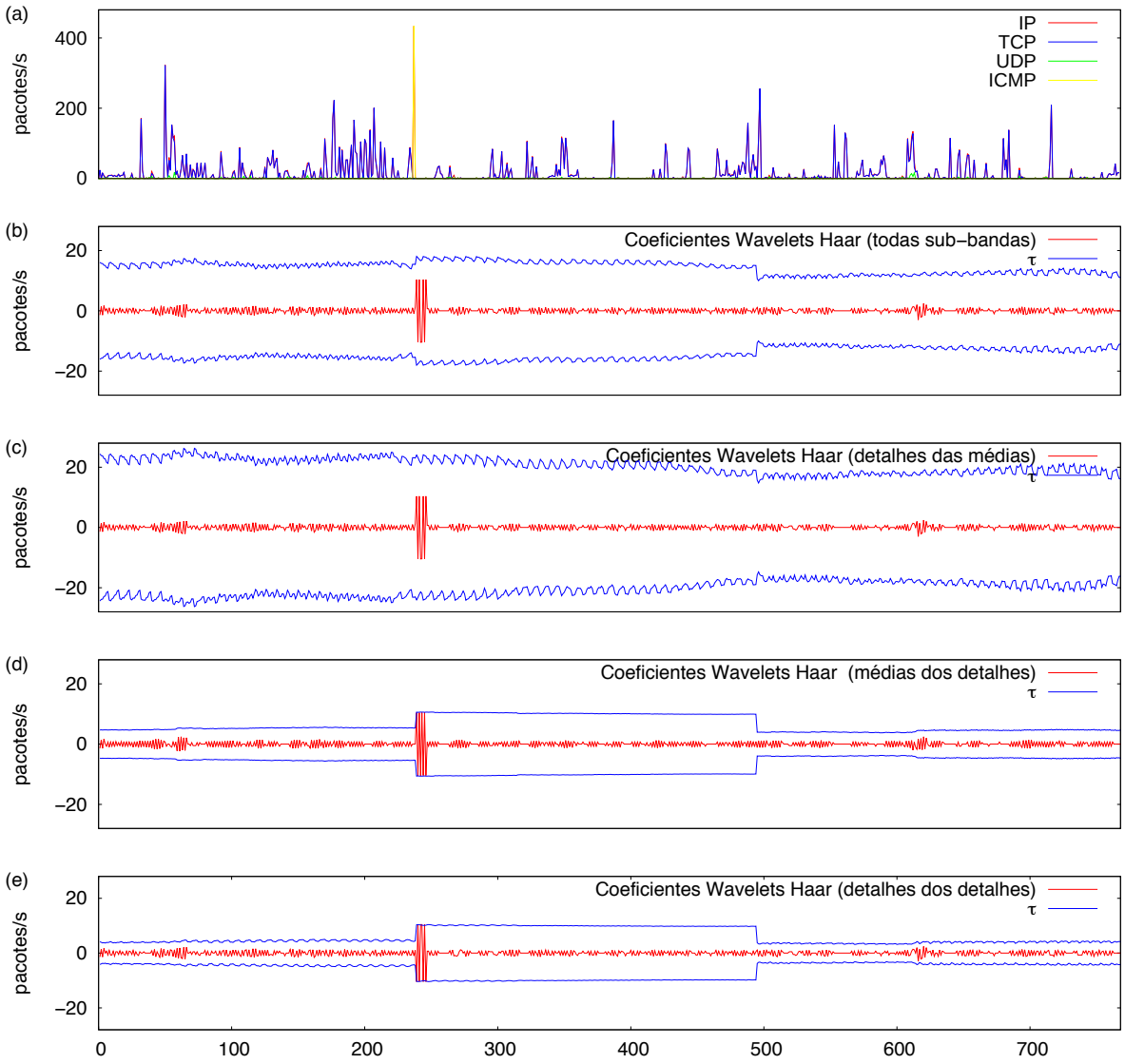

Figura 5.13: Uso da família wavelet de Haar e estratégia adaptativa nos coeficientes wavelets normalizados durante um ataque PoD. (a) Gráfico do uso da rede e seus descritores. (b) Todas as sub-bandas de coeficientes de detalhes. (c) Sub-banda médias dos detalhes dos coeficientes de detalhes. (d) Sub-banda detalhes das médias dos coeficientes de detalhes. (e) Sub-banda detalhes dos detalhes dos coeficientes de detalhes.

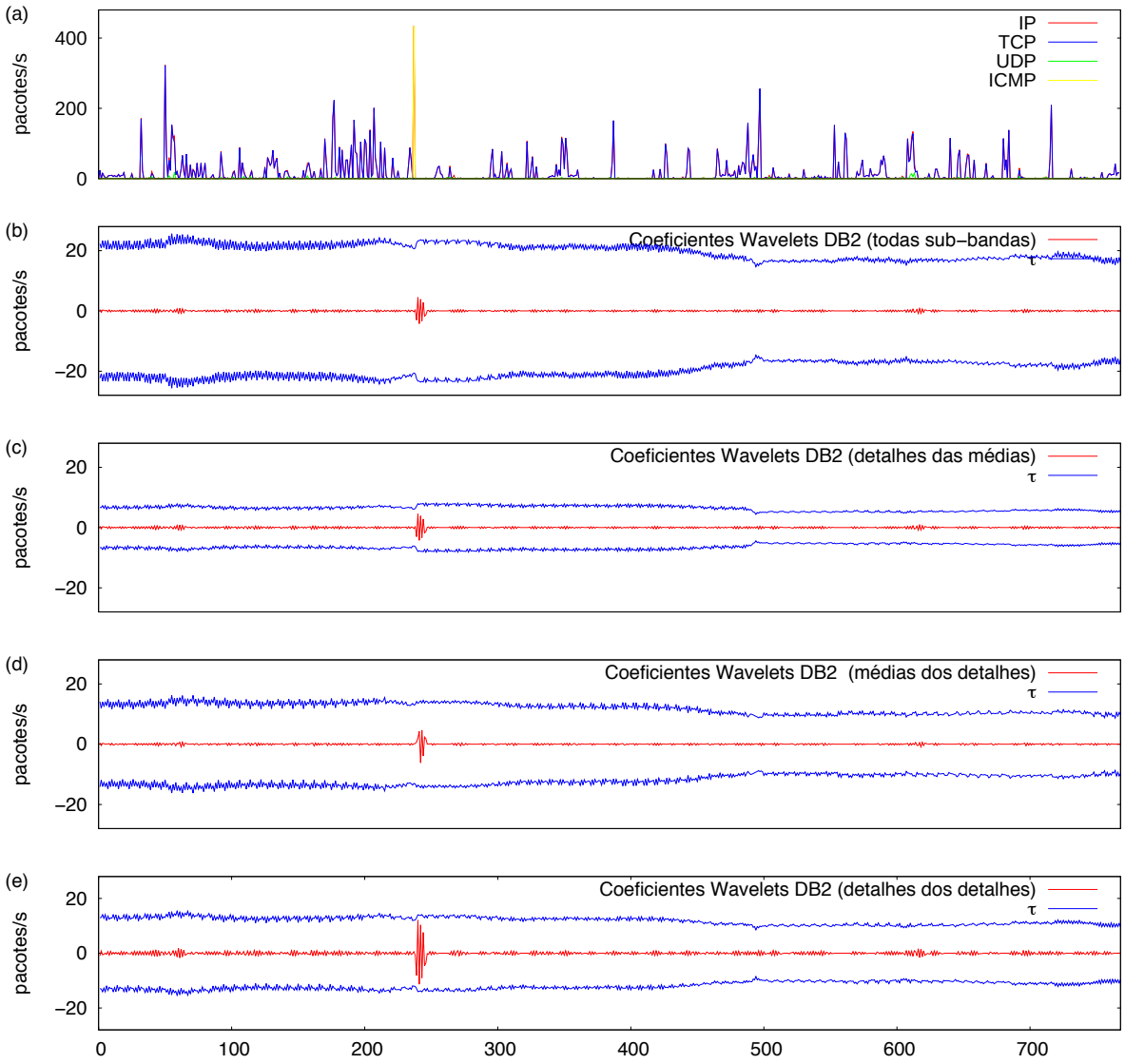

Figura 5.14: Uso da família wavelet de daubechies Db2 e estratégia adaptativa nos coeficientes wavelets normalizados durante um ataque PoD. (a) Gráfico do uso da rede e seus descritores. (b) Todas as sub-bandas de coeficientes de detalhes. (c) Sub-banda médias dos detalhes dos coeficientes de detalhes. (d) Sub-banda detalhes das médias dos coeficientes de detalhes. (e) Sub-banda detalhes dos detalhes dos coeficientes de detalhes.

Os gráficos da Figura 5.15 apresentam os verdadeiros positivos do NIDS utilizando a estratégia adaptativa, com diferentes famílias wavelets e valores para a janela deslizante, assim como coeficientes wavelets normalizados e não normalizados. Na Figura 5.16 são apresentados os resultados referentes a falsos positivos. Os testes efetuados com as diferentes famílias wavelets com e sem normalização dos coeficientes wavelets, pode-se notar que quanto maior o tamanho da janela deslizante, maior o número de verdadeiros positivos e menor o número de falsos positivos. Isto ocorre pois quanto mais pontos são disponibilizados para o cálculo da transformada wavelet, maior o conhecimento sobre o comportamento do tráfego de rede.

Diferentemente dos testes efetuados com o uso da estratégia de corte recursiva, a normalização dos coeficientes wavelets em conjunto com o uso da estratégia de corte adaptativa trouxe melhorias nos resultados, detectanto um menor número de FP. A taxa de detecção manteve o mesmo nível de quando os coeficientes wavelets não são normalizados, mas o número de falsos positivos diminuiu, melhorando a eficácia do algoritmo.

Através da normalização dos coeficientes wavelets ocorreu uma melhora na taxa de detecção, e diminuição no número de falsos positivos, tornando o uso da normalização recomendável.

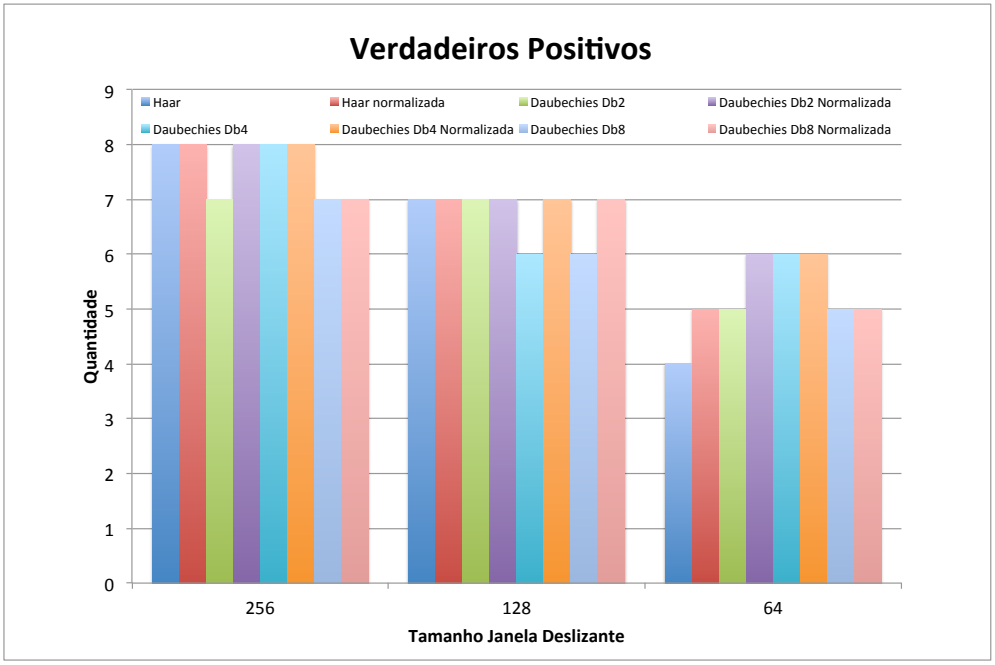

Figura 5.15: Gráfico com os Verdadeiros Positivos

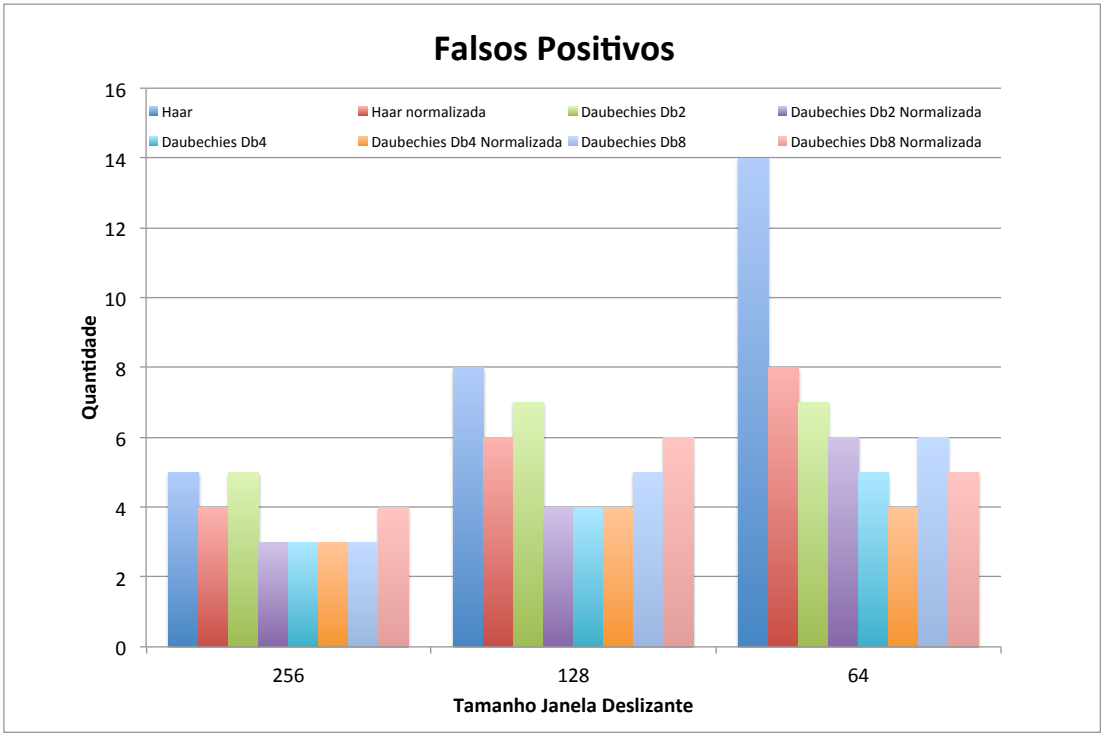

Figura 5.16: Gráfico com os Falsos Positivos

Nos resultados dos experimentos realizados a configuração que teve o melhor desempenho foi a que utiliza a transformada wavelet de *Daubechies Db4* com os coeficientes wavelets normalizados e uma janela deslizante de 256 amostras, conforme mostram os gráficos das Figuras 5.15 e 5.16. O uso da normalização dos coeficientes wavelets diminui o número de falsos positivos, e caso o *overhead* adicional ocasionado pelo seu cálculo não seja um problema, é recomendável a sua utilização.

5.2.3 Estudo de Caso 3: UFSM com operação de corte adaptativa

Para a realização deste experimento foram utilizados os dados da base de dados UFSM. São executados testes utilizando as famílias wavelets de Haar, e Daubechies Db2, Db4 e Db8. Após o uso da transformada wavelet foram executados testes com e sem normalização dos coeficientes wavelets.

A estratégia de corte adaptativa foi utilizada nestes testes. A estratégia de corte recursiva não foi utilizada nos experimentos com a base de dados da UFSM pois os resultados dos experimentos realizados na sub-seção 5.2.1 tornam o uso desta estratégia não recomendável.

Os ataques inseridos na base de dados da UFSM perturbam os protocolos de rede: IP, TCP, ICMP e UDP, sendo estes os descritores extraídos da base de dados UFSM, com uma taxa de amostragem de 1 segundo. Nos experimentos desta sub-seção as fronteiras da família wavelet

de Daubechies Db2, Db4 e Db8 foram estendidas pela repetição do último valor.

O gráfico presente na Figura 5.17 apresenta os resultados em relação ao número de verdadeiros positivos obtidos nos testes com a base de dados da UFSM, e o gráfico da Figura 5.17 apresenta os falsos positivos. Estes gráficos apresentam os testes utilizando diversas famílias wavelets com e sem normalização, o tamanho da janela deslizante, o número de verdadeiros positivos (VP) e o número de falsos positivos (FP).

A Figura 5.17 apresenta os resultados obtidos com a base de dados da UFSM. O tamanho da janela deslizante influencia diretamente na eficácia do algoritmo de detecção. Quando foram efetuados testes com uma janela deslizante de 256 amostras foi obtido o maior percentual da taxa de detecção.

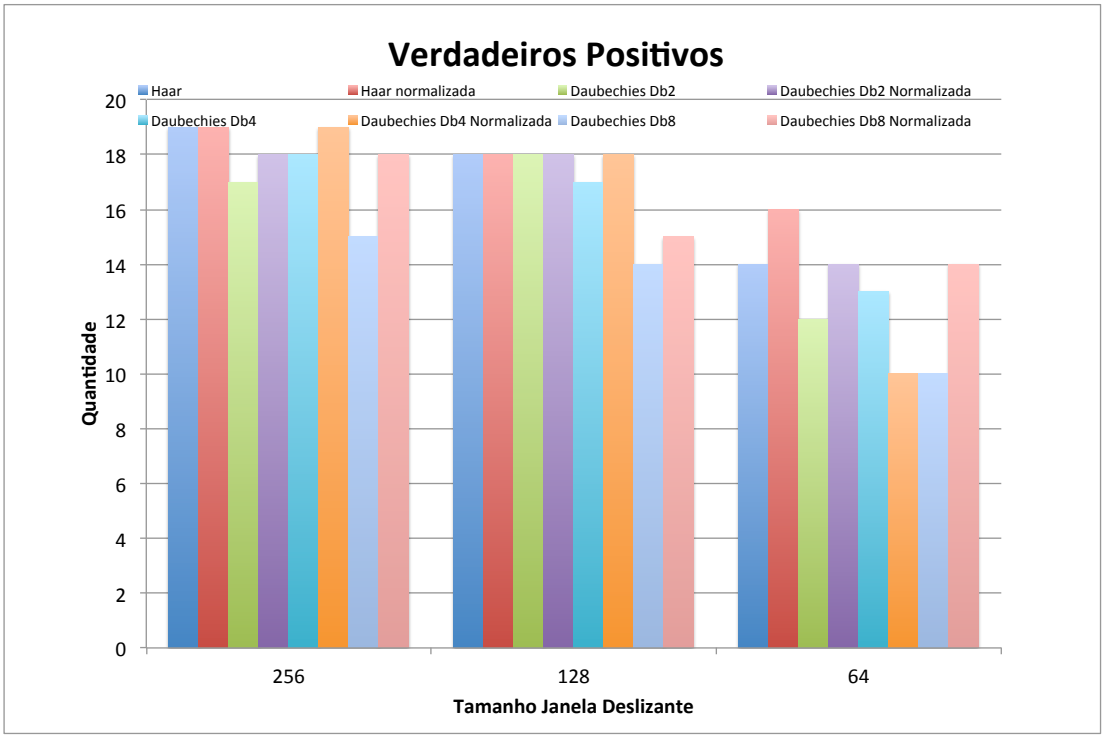

Figura 5.17: Gráfico com os Verdadeiros Positivos

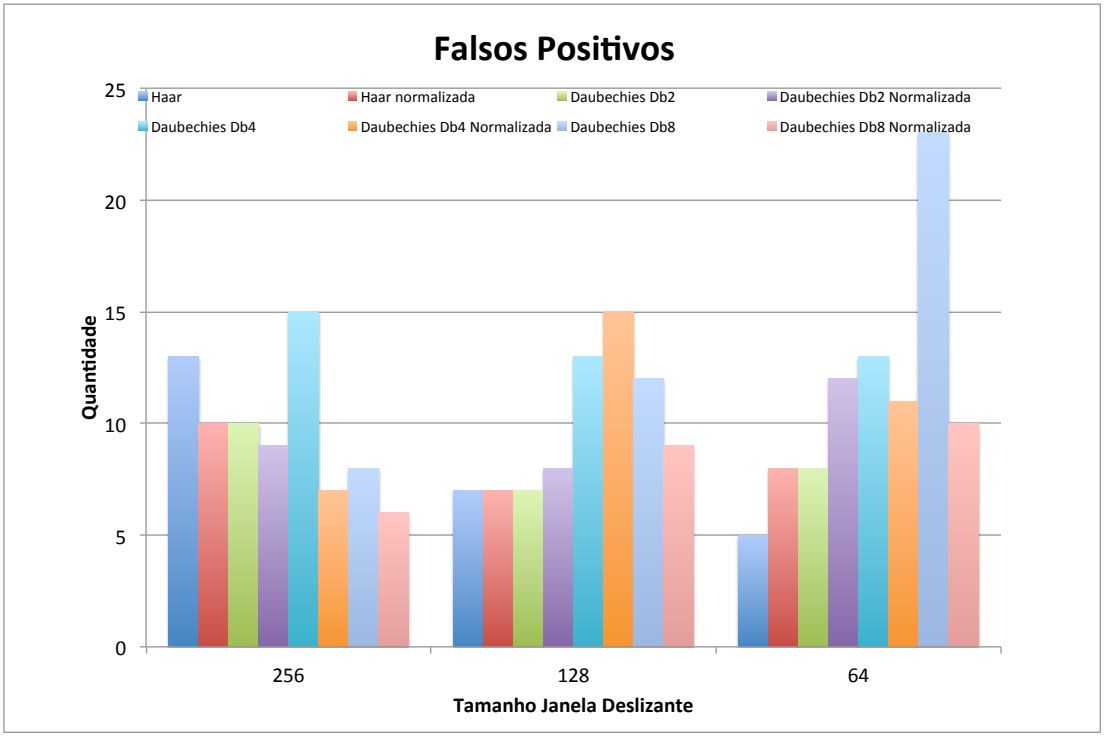

Figura 5.18: Gráfico com os Falsos Positivos

Nos testes efetuados com a transformada wavelet de *Daubechies Db8*, discretizados na Figura 5.18, o número de falsos positivos é menor quando utiliza-se uma janela deslizante de 256 amostras, diferentemente do que ocorre nos testes efetuados com outras famílias wavelets. Este fato ocorre devido a suavização do sinal obtida, com um número de filtros maior, e maior quantidade de dados.

Os experimentos realizados com a normalização dos coeficientes wavelets da transformada de Haar, *Daubechies* Db2, Db4 e Db8, melhoram a eficácia do algoritmo de detecção, pois o número de falsos positivos é menor, e o número de verdadeiros positivos é maior, conforme mostras as Figuras 5.17 e 5.18. Nestes testes a taxa de detecção melhorou em comparação com os coeficientes wavelets não normalizados, e houve diminuição no número de falsos positivos.

Analisando os testes efetuados com a base de dados da UFSM, a melhor configuração do algoritmo de detecção é quando utilizada a transformada wavelet de *Daubechies Db4*, com a normalização dos coeficientes wavelets e uma janela deslizante contendo 256 amostras.

A diferença nos resultados deve-se a diferença da família wavelet utilizada, assim como da suavização dos coeficientes wavelets ocasionada pela normalização.

5.2.4 Desempenho

Para avaliar o custo de processamento do algoritmo que detecta ataques *DoS* foram realizados testes para avaliar o tempo necessário para calcular cada janela deslizante, que é atualizada a cada $\triangle t$ intervalo de tempo. O tempo médio necessário para processar as informações é de 0,4125 milissegundos, tornando o algoritmo proposto uma opção rápida para utilização em NIDS. O intervalo de tempo $\triangle t$ deve ser maior que o tempo necessário para efetuar os cálculos. A Tabela 5.4 apresenta os tempos médios de execução de cada família wavelet e tamanho da janela deslizante. Quanto maior o número de filtros maior a complexidade computacional. Quando é utilizada a normalização dos coeficientes wavelets o custo computacional também é maior.

Tabela 5.4: Métricas obtidas na base de dados da UFSM com a utilização da transformada wavelet de *Daubechies Db8* sem a normalização dos coeficientes

Família wavelet	Tamanho Janela	Normalização	Tempo (ms)
Haar	256	Não	0,230
Haar	128	Não	0,110
Haar	64	Não	0,070
Haar	256	Sim	0,280
Haar	128	Sim	0,130
Haar	64	Sim	0,080
Daub Db2	256	Não	0,650
Daub Db2	128	Não	0,330
Daub Db2	64	Não	0,160
Daub Db2	256	Sim	0,670
Daub Db2	128	Sim	0,360
Daub Db2	64	Sim	0,170
Daub Db4	256	Não	0,730
Daub Db4	128	Não	0,440
Daub Db4	64	Não	0,200
Daub Db4	256	Sim	0,750
Daub Db4	128	Sim	0,430
Daub Db4	64	Sim	0,210
Daub Db8	256	Não	1,100
Daub Db8	128	Não	0,830
Daub Db8	64	Não	0,310
Daub Db8	256	Sim	0,900
Daub Db8	128	Sim	0,440
Daub Db8	64	Sim	0,320

Estes experimentos foram mensurados no seguinte ambiente:

- Computador com processador Intel Core i5 de 2.3 GHz de frequência de *clock*

- 8 GB de memória RAM DDR3 com frequência de 1333 MHz
- Sistema Operacional Mac OS X Lion 10.7.2 (11C74)
- Máquina virtual 1.6.0_29

5.3 Considerações Finais e Trabalhos Relacionados

Os experimentos apresentados neste capítulo foram projetados para avaliar a capacidade de detecção de ataques DoS do algoritmo proposto neste trabalho. Foram realizados experimentos nas duas base de dados: DARPA e UFSM.

Foram levados em consideração todos os protocolos de rede, onde ocorrem perturbações devidas a ataques de negação de serviço. Os protocolos utilizados foram: IP, TCP, ICMP e UDP.

As famílias Wavelets utilizadas nos testes foram a *Haar*, *Daubechies Db2*, *Daubechies Db4* e *Daubechies Db8*. Foram efetuados experimentos com e sem normalização dos coeficientes wavelets, e com janelas deslizantes de diferentes tamanhos. Foram também analisadas duas estratégias de corte: adaptativa e recursiva.

A matriz de dados de entrada foi gerada através de informações contidas nas bases de dados da UFSM e DARPA. A matriz de entrada é preenchida sequencialmente através de uma janela deslizante. Inicialmente foram efetuados experimentos com uma janela deslizante de 256 pontos para cada um dos descritores de rede, gerando a matriz de entrada. O tamanho da janela deslizante também foi testado com 128 e 64 pontos. A cada nova amostra disponível a última amostra de cada variável que compõem a matriz é descartada, sendo inserida a amostra nova. A transformada Wavelet é aplicada a cada atualização da janela deslizante.

Nos testes efetuados com a estratégia de corte recursiva foram obtidas taxas de detecção de 100%, mas com um número elevado de falsos positivos. O capacidade de um NIDS detectar ataques é avaliado tanto pela capacidade de detecção correta (VP), quanto pela classificação incorreta. Um NIDS que gera muitos falsos positivos não é utilizado pois classifica o tráfego correto como sendo ataque.

A estratégia de corte adaptativa obteve os melhores resultados nos experimentos, detectando corretamente a maioria dos ataques presentes nas bases de dados.

A taxa de detecção na base de dados da DARPA foi de até 100% com uma janela deslizante de 256 amostras utilizando a transforma wavelet de Haar e *Daubechies Db4* sem a normalização

dos coeficientes wavelets, e de 100% nas wavelets de Haar, *Daubechies Db2* e *Daubechies Db4* com a normalização dos coeficientes wavelets. Estes resultados foram obtidos com uma janela deslizante contendo 256 amostras.

Na base de dados da UFSM a taxa de detecção foi de até 95% dos ataques presentes na base de dados quando utilizada a transformada de Haar sem normalização, e transformada de Haar e *Daubechies Db4* quando os coeficientes wavelets são normalizados.

Nos experimentos os melhores resultados foram obtidos com a transformada wavelet de *Daubechies Db4* utilizando a normalização dos coeficientes wavelets.

A análise dos resultados com trabalhos relacionados é uma tarefa difícil, devido a utilização de diversas bases de dados e diferentes modos de extração de dados destas bases de dados. Por este motivo são efetuadas comparações com trabalhos que utilizaram a base de dados DARPA.

No trabalho proposto em (DAINOTTI; PESCAPE; VENTRE, 2006) os testes foram efetuados na base de dados DARPA, com a utilização de traços sem ataques, sendo inserido ataques modelados através de ferramentas como o TFN2K. A taxa de detecção foi de até 95,9%.

São analisadas diferentes famílias Wavelets em (LU; TAVALLAEE; GHORBANI, 2008) para verificar o impacto de bases Wavelet no desempenho da aplicação, com testes indicando uma taxa de detecção de 70% utilizando a wavelet de *Daubechies* e 60% com as wavelets de *Coiflets* e *Symlets*. Os dados foram convertidos de pacotes/s para fluxos/s. O percentual de detecção obtido foi através da análise de diversos tipos de fluxos (15 tipos ao total).

Nos testes efetuados por (DALMAZO et al., 2009) foi obtido um percentual de 77% na taxa de detecção dos ataques, com um pequeno número de falsos positivos.

Deste modo, pelos experimentos realizados, observou-se que a taxa de detecção de ataques DoS obtida pelo algoritmo proposto nesta dissertação é melhor do que nos trabalhos comparados, apontando o uso de wavelet 2D como promissora em IDS baseados em detecção de anomalias.

6 CONCLUSÕES

A Internet está no caminho para se tornar o meio de comunicação para todos os tipos de informações, desde a simples transferência de arquivos de computadores até a transmissão de voz, vídeo ou informações interativas em tempo real. A Internet está evoluindo de um serviço de melhor esforço simples para uma rede multi-serviços. Como consequência, a Internet é muito vulnerável a ataques, especialmente ataques DoS e DDoS.

Este trabalho propõe um algoritmo para detectar instrusões ocasionadas por ataques de negação de serviço através de uma abordagem baseada em detecção de anomalias. Para explorar a relação que existe entre os diferentes protocolos de rede, o algoritmo detecção de ataques DoS é baseado na transformada Wavelet 2D. Esta ferramenta matemática de processamento de sinais mostrou-se adequada para a análise e detecção de ataques DoS em redes de computadores.

Nos experimentos realizados foram utilizados três tamanhos de janelas deslizantes (64, 128 e 256 amostras) e para a geração de alarmes foi utilizada a estratégia de *hard threshold* para corte dos coeficientes wavelets.. Os testes foram efetuados com diferentes famílias wavelets e com e sem normalização dos coeficientes wavelets. Foram efetuados testes com um algoritmo recursivo e outro adaptativo para calcular o valor de corte. Os experimentos foram então efetuados em duas bases de dados (DARPA e UFSM). Como resultado, observou-se a possibilidade do uso da transformada Wavelet 2D como uma ferramenta para relacionar informações do tráfego de rede e detectar ataques do tipo DoS.

A taxa de detecção foi de até 95% com um baixo número de falsos positivos na base de dados da UFSM, utilizando a transformada wavelet de *Daubechies Db4* e normalização dos coeficientes wavelets. Na base de dados DARPA a taxa de detecção foi de até 100% com uma janela deslizante de 256 amostras, wavelet de *Daubechies Db4* e coeficientes wavelets normalizados. O custo computacional necessário para o processamento do algoritmo proposto permite que o intervalo entre as detecções seja no pior dos casos de 1,1 milissegundo.

Dos experimentos conclui-se finalmente que a utilização da transformada wavelet 2D na detecção de ataques de negação de serviço em redes de computadores é uma técnica viável e que pode compor o conjunto de algoritmos de um sistema de detecção de intrusão.

6.1 Trabalhos Futuros

A detecção de intrusão através de métodos baseados em processamentos de sinais é uma área que permite a classificação de anomalias sem a necessidade de conhecimento prévio do comportamento de rede.

Neste trabalho foram utilizadas a família ortonormal de *Daubechies*, podendo ser investigado testes com diferentes famílias Wavelets. O uso de wavelet *packets*, que é uma técnica de aplicação da transformada wavelet nos coeficientes de detalhes, também pode ser avaliada para analisar e verificar o impacto nas métricas de desempenho de IDS.

O uso de Wavelets 2D para detecção de outras categorias de ataques também pode ser explorado.

REFERÊNCIAS

AHMED, T.; ORESHKIN, B.; COATES, M. Machine Learning Approaches to Network Anomaly Detection. **Second Workshop on Tackling Computer Systems Problems with Machine Learning Techniques**, Cambridge, MA, p.7:1–7:6, 2007.

APACHE. The Apache HTTP Server Project. http://httpd.apache.org/ Acesso em Outubro de 2011.

AZEVEDO, R. P. de; KOZAKEVICIUS, A.; FOSTER, D.; SANTOS, E. dos. Transformada wavelet discreta no tratamento de imagens. **Anais do XXXIII Congresso de Matemática Aplicada e Computacional - CNMAC**, Águas de Lindóia, SP, p.918 – 919, 2010.

BADISHI, G.; KEIDAR, I.; SASSON, A. Exposing and Eliminating Vulnerabilities to Denial of Service Attacks in Secure Gossip-Based Multicast. **IEEE Transactions on Dependable and Secure Computing**, Los Alamitos, CA, USA, v.3, p.45–61, 2006.

BARFORD, P.; KLINE, J.; PLONKA, D.; RON, A. A signal analysis of network traffic anomalies. **Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurment**, New York, NY, USA, p.71–82, 2002.

CHANDOLA, V.; BANERJEE, A.; KUMAR, V. Anomaly detection: a survey. **ACM Computing Surveys**, v.41, n.3, p.1–58, 2009.

CISCO. Cisco Netranger Documentation. http://www.cisco.com/univercd/cc/td/doc/product/iaabu/netrangr/ Acesso em Outubro de 2011.

CORRÊA, J. L.; CANSIAN, A. M. Detecção de ataques de negativa de serviço por meio de fluxos de dados e sistemas inteligentes. **Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais (SBSEG 2007)**, Rio de Janeiro, RJ, p.125–141, 2007.

CSI/FBI. **Computer crime and security survey**. http://gocsi.com/survey. Acesso em Dezembro de 2011.

CSI/FBI. **Computer crime and security survey**. http://gocsi.com/survey. Acesso em Dezembro de 2011.

DAINOTTI, A.; PESCAPE, A.; VENTRE, G. NIS04-1: wavelet-based detection of dos attacks. **IEEE Global Telecommunications Conference, 2006. GLOBECOM '06.**, p.1 –6, nov. 2006.

DALMAZO, B. L.; PERLIN, T.; NUNES, R. C.; JESUS KOZAKEVICIUS, A. de. Filtros de Alarmes de Anomalias através de Wavelets. **Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais (SBSEG 2009)**, Campinas, SP, p.85–98, 09 2009.

DARPA. **MIT Lincoln Laboratory**: information systems technology. http://www.ll.mit.edu/mission/communications/ist/corpora/ideval/. Acesso em Dezembro de 2011.

DAUBECHIES, I. **Ten lectures on Wavelets**. Montpelier. Vermont. Capital City Press, 1992.

DITTRICH, D. **The Tribe Flood Network Distributed Denial of Service attack tool**. University of Washington, 1999. http://staff.washington.edu/dittrich/misc/tfn.analysis. Acesso em Dezembro de 2011.

DITTRICH, D.; WEAVER, G.; DIETRICH, S.; LONG, N. **The _mstream_ Distributed Denial of Service attack tool**. 2000. http://staff.washington.edu/dittrich/misc/mstream.analysis.txt. Acesso em Dezembro de 2011.

DONOHO, D.; I., J. Ideal spatial adaptation via wavelet shrinkage. **Biometrika**, n.85, p.425–455, Dec 1994.

DONOHO, D. L.; JOHNSTONE, I. M. Adapting to unknown smoothness via wavelet shrinkage. **Journal of the American Statistical Association**, p.1200–1224, 1995.

EDDY, W. TCP SYN Flooding Attacks and Common Mitigations. http://tools.ietf.org/html/rfc4987 Acesso em Outubro de 2011.

ERRAMILLI, A.; NARAYAN, O.; WILLINGER, W. Experimental queueing analysis with long-range dependent packet traffic. **Networking, IEEE/ACM Transactions on**, v.4, n.2, p.209 –223, apr 1996.

FARRAPOSO, S.; OWEZARSKI, P.; MONTEIRO, E. On the use of traffic monitoring and measurements for improving networking. **Proceedings of Telecommunications, advanced industrial conference on telecommunications/service assurance with partial and intermittent**

resources conference/e-learning on telecommunications workshop. aict/sapir/elete, p.416 – 421, July 2005.

GARCÍA-TEODORO, P.; DÍAZ-VERDEJO, J.; MACIÁ-FERNÁNDEZ, G.; VÁZQUEZ, E. Anomaly-based network intrusion detection: techniques, systems and challenges. **Computers & Security**, v.28, n.1-2, p.18–28, 2009.

HUANG, C.-T.; THAREJA, S.; SHIN, Y.-J. Wavelet-based Real Time Detection of Network Traffic Anomalies. **Securecomm and Workshops, 2006**, p.1 –7, Aug. 2006.

KIHONG, P.; WALTER, W. **Self-Similar Network Traffic and Performance Evaluation**. 1st.ed. New York, NY, USA, John Wiley & Sons, Inc., 2000.

KIM, S. S.; REDDY, A. Statistical Techniques for Detecting Traffic Anomalies Through Packet Header Data. **Networking, IEEE/ACM Transactions on**, v.16, n.3, p.562 –575, June 2008.

KUMAR, P.; SELVAKUMAR, S. Distributed Denial-of-Service (DDoS) Threat in Collaborative Environment - A Survey on DDoS Attack Tools and Traceback Mechanisms. **Advance Computing Conference, 2009. IACC 2009. IEEE International**, p.1275 –1280, Mar 2009.

LELAND, W. E.; TAQQU, M. S.; WILLINGER, W.; WILSON, D. V. On the self-similar nature of Ethernet traffic (extended version). **IEEE/ACM Trans. Netw.**, Piscataway, NJ, USA, v.2, p.1–15, February 1994.

LI, M.; LI, M. A New Approach for Detecting DDoS Attacks Based on Wavelet Analysis. **Proceedings of 2nd International Congress on Image and Signal Processing, 2009. CISP '09.**, p.1 –5, 17-19 2009.

LOUKAS, G.; ÖKE, G. Protection Against Denial of Service Attacks: a survey. **The Computer Journal Advance Access**, 2009.

LU, W.; TAVALLAEE, M.; GHORBANI, A. Detecting Network Anomalies Using Different Wavelet Basis Functions. **Communication Networks and Services Research Conference, 2008. CNSR 2008. 6th Annual**, Halifax, NS, p.149 –156, May 2008.

MAFRA, P. M.; SILVA FRAGA, J. da; MOLL, V.; SANTIN, A. O. POLVO-IIDS: um sistema de detecção de intrusão inteligente baseado em anomalias. **Anais do Simpósio Brasileiro em**

Segurança da Informação e de Sistemas Computacionais (SBSEG 2008), Gramado, RS, p.201–214, Set 2008.

MALLAT, S. **A wavelet tour of signal processing**. Academic Press, 1998.

MALLAT, S. G. A theory for multiresolution signal decomposition: the wavelet representation. **Pattern Analysis and Machine Intelligence, IEEE Transactions on**, v.11, n.7, p.674–693, 1989.

MCHUGH, J. Testing Intrusion detection systems: a critique of the 1998 and 1999 darpa intrusion detection system evaluations as performed by lincoln laboratory. **ACM Trans. Inf. Syst. Secur.**, New York, NY, USA, v.3, n.4, p.262–294, 2000.

MICROSOFT. The Oficial Microsoft IIS Site. http://www.iis.net/ Acesso em Outubro de 2011.

NIELSEN, O. Wavelets in Scientific Computing. **PhD thesis, Technical University of Denmark**, Department of Informatics and Mathematical Modelling, Technical University of Denmark, DTU, 1998.

OHSITA, Y.; ATA, S.; MURATA, M. Detecting distributed denial-of-service attacks by analyzing TCP SYN packets statistically. **Global Telecommunications Conference, 2004. GLOBECOM '04. IEEE**, v.4, p.2043 – 2049, Vol.4, Nov 2004.

ONUT, I.-V.; GHORBANI, A. Toward a feature classification scheme for network intrusion detection. **Proceedings of the 4th Annual Communication Networks and Services Research Conference, 2006. CNSR 2006.**, p.278–284, May 2006.

PATCHA, A.; PARK, J.-M. An overview of anomaly detection techniques: existing solutions and latest technological trends. **Computer Networks**, v.51, n.12, p.3448–3470, August 2007.

POSTEL, J. RFC 768 - User Datagram Protocol. http://www.faqs.org/rfcs/rfc768.html Acesso em Outubro de 2011.

POSTEL, J. RFC 792 - Internet Control Message Protocol. http://www.faqs.org/rfcs/rfc792.html Acesso em Outubro de 2011.

PROLEXIC. **Prolexic Attack Report. Q3 2011**. www.prolexic.com/attackreports. Acesso em Dezembro de 2011.

ROESCH, M. Snort - Lightweight Intrusion Detection for Networks. **Proceedings of the 13th USENIX conference on System administration, LISA '99**, Berkeley, CA, USA, p.229–238, 1999.

SAMAAN, N.; KARMOUCH, A. Network anomaly diagnosis via statistical analysis and evidential reasoning. **IEEE Transactions on Network and Service Management,**, v.5, n.2, p.65–77, June 2008.

SANFILIPPO, S. **Hping - Active Network Security Tool**. http://www.hping.org/. Acesso em Dezembro de 2011., 2011.

SCHERRER, A.; LARRIEU, N.; OWEZARSKI, P.; BORGNAT, P.; ABRY, P. Non-Gaussian and Long Memory Statistical Characterizations for Internet Traffic with Anomalies. **IEEE Transactions on Dependable and Secure Computing,**, v.4, n.1, p.56 –70, Jan.-Mar 2007.

SERPRO, S. F. d. P. d. D. S. **Serpro faz balanço de medidas de segurança em resposta a ataques virtuais**. http://www.serpro.gov.br/noticiasSERPRO/2011/junho/serpro-faz-balanco-de-medidas-de-seguranca-em-resposta-a-ataques-virtuais/. Acesso em Dezembro de 2011.

SOULE, A.; SALAMATIAN, K.; TAFT, N. Combining filtering and statistical methods for anomaly detection. **Proceedings of the 5th ACM SIGCOMM conference on Internet Measurement**, Berkeley, CA, USA, p.31–31, 2005.

SPECHT, S. M. Distributed denial of service: taxonomies of attacks, tools and countermeasures. **Proceedings of the International Workshop on Security in Parallel and Distributed Systems, 2004**, Cambridge, Massachusetts, USA, p.543–550, 2004.

STOLLNITZ, E.; DEROSE, A.; SALESIN, D. Wavelets for computer graphics: a primer 1. **IEEE Computer Graphics and Applications**, v.15, n.3, p.76–84, May 1995.

WANG, J. **Computer network security**: theory and practice. Higher Education Press, 2009.

ZHANG, J.; REXFORD, J.; FEIGENBAUM, J. Learning-Based Anomaly Detection in BGP Updates. **ACM SIGCOMM MineNet workshop**, New York, NY, USA, p.219–220, 2005.