IDENTIFICAÇÃO DE ANOMALIAS EM FLUXOS DE REDE UTILIZANDO PREVISÕES EM SÉRIES TEMPORAIS PELO MÉTODO DE HOLT-WINTERS

Vander Luiz Proença da Silva

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Engenharia de Sistemas e Computação, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia de Sistemas e Computação.

Orientador: Luís Felipe Magalhães de Moraes

Rio de Janeiro
Setembro de 2015

IDENTIFICAÇÃO DE ANOMALIAS EM FLUXOS DE REDE UTILIZANDO PREVISÕES EM SÉRIES TEMPORAIS PELO MÉTODO DE HOLT-WINTERS

Vander Luiz Proença da Silva

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA (COPPE) DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO.

Examinada por:

Prof. Luís Felipe Magalhães de Moraes, Ph.D.

Prof. Nilton Alves Junior, D.Sc.

Prof. Felipe Maia Galvão França, Ph.D.

Prof. Jorge Lopes de Souza Leão, Dr.Ing.

RIO DE JANEIRO, RJ – BRASIL
SETEMBRO DE 2015

Silva, Vander Luiz Proença da

Identificação de Anomalias em Fluxos de Rede Utilizando Previsões em Séries Temporais pelo Método de Holt-Winters/Vander Luiz Proença da Silva. – Rio de Janeiro: UFRJ/COPPE, 2015.

XIV, 76 p.: il.; 29, 7cm.

Orientador: Luís Felipe Magalhães de Moraes

Dissertação (mestrado) – UFRJ/COPPE/Programa de Engenharia de Sistemas e Computação, 2015.

Referências Bibliográficas: p. 66 – 71.

1. Network. 2. Redes de Computador. 3. Anomaly. 4. Anomalias. 5. Anomaly Detection. 6. Detecção de Anomalias. 7. Time Series. 8. Séries Temporais. 9. Holt-Winters. 10. Traffic Partition. 11. Classificação de Tráfego. 12. Partição de Tráfego. 13. Flows. 14. Fluxos de rede. 15. NetFlow. 16. NFSen. I. Moraes, Luís Felipe Magalhães de. II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia de Sistemas e Computação. III. Título.

Aos meus pais.

Agradecimentos

Ao meu Deus pelo sustento e pelas condições de chegar até aqui. À minha família pelo apoio, amor, carinho e pela imensa paciência que tiveram comigo. Ao meu pai Luiz Acácio que em todas as ocasiões estendeu a mão e me ajudou em diversas situações. À minha mãe Elizabeth pelo apoio emocional nos momentos mais turbulentos. À minha irmã Raquel pelo incentivo e companheirismo.

À minha namorada Danielle pela compreensão durante a elaboração deste trabalho, pela força e pelas noites de sábado passadas na frente de um computador ao meu lado.

Aos meus amigos do laboratório Ravel, Evandro, Felipe, Renato, Marcos e José pelos interessantes debates e pelas trocas de ideias sobre o tema deste trabalho. Ressalto minha profunda gratidão ao Evandro e Felipe que ajudaram diretamente no planejamento e montagem das ferramentas utilizadas para a validação do método.

Ao meu orientador Luís Felipe pelo apoio em todo o mestrado, especialmente durante a elaboração deste trabalho. Aos integrantes da banca Nilton Alves Junior, Felipe Maia Galvão França e Jorge Lopes de Souza Leão pela valiosa contribuição dada a este trabalho.

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

IDENTIFICAÇÃO DE ANOMALIAS EM FLUXOS DE REDE UTILIZANDO PREVISÕES EM SÉRIES TEMPORAIS PELO MÉTODO DE HOLT-WINTERS

Vander Luiz Proença da Silva

Setembro/2015

Orientador: Luís Felipe Magalhães de Moraes

Programa: Engenharia de Sistemas e Computação

A área de segurança de redes de computadores apresenta atualmente diversos desafios. Com o avanço das tecnologias de telecomunicações e a popularização da Internet, vem crescendo rapidamente a quantidade de tráfego gerado a partir de aplicações mau intencionadas, ameaçando o funcionamento das redes envolvidas como um todo e colocando em risco os usuários. Em princípio, através da observação dos padrões de tráfego na rede, é possível identificar anomalias geradas por fluxos relacionados a objetivos maliciosos. Com o intuito de auxiliar os gerentes das redes a prover uma maior segurança aos seus usuários, esta dissertação apresenta uma proposta de técnica para a detecção e classificação de anomalias em fluxos de rede, com suas respectivas implementação e avaliação. A metodologia proposta consiste da coleta dos fluxos que passam pela borda da rede, para classifica-los em quatro categorias. A partir dessa classificação, são definidas métricas relacionadas ao comportamento da rede. Através de medidas empíricas das métricas definidas, a técnica de análise utilizada consiste na construção de séries temporais, e da comparação dos valores medidos e de previsões obtidas com o modelo de Holt-Winters. Uma implementação desta metodologia foi construída e testada num ambiente real, utilizando dados coletados a partir do tráfego passando pelos roteadores de borda da Rede-Rio de Computadores. Através dessa coleta, o método aqui proposto foi capaz de detectar anomalias relacionadas a varreduras de porta e ataques de negação de serviço mais populares. Os resultados obtidos demonstram a viabilidade de utilização da proposta apresentada para a detecção de anomalias relacionadas a fluxos maliciosos em redes. A técnica proposta foi implementada no servidor IPTRAF, localizado do laboratório Ravel e vem sendo testada como ferramenta de apoio às tarefas de gerência da Rede-Rio.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

ANOMALY DETECTION IN NETWORK FLOWS USING HOLT-WINTERS TIME SERIES FORECASTING METHOD

Vander Luiz Proença da Silva

September/2015

Advisor: Luís Felipe Magalhães de Moraes

Department: Systems Engineering and Computer Science

The area of computer networks security presents several challenges currently. With the advance of the telecommunication technologies and the popularity of the Internet, the amount of traffic originated from bad intentioned applications has been rapidly growing, threatening operation of the related networks as a whole and placing a risk situation to the users. In principle, through the observation of network traffic patterns, it is possible to identify anomalies generated by flows related to malicious objectives. In order to help network managers to provide better security to their users, this dissertation presentes a technique proposal for the detection and classification of anomalies in network flows, with their respective implementation and evaluation. The proposed methodology consists of collecting the flows passing through the border of the network, classifying them into four categories. From such a classification, metrics related to network the behavior are defined, in order to identify possible anomalies in the traffic patterns. Through empirical measurements of the defined metrics, the analysis technique utilised consists of the construction of time series, for comparing the values acquired from measurements with predictions obtained by using the Holt-Winters model. An implementation of this methodology has been built and tested in a real environment, using collected data captured from the traffic passing through the edge routers of *Rede-Rio de Computadores*. Through this capture, the proposed method was able to detect anomalies related to the most popular port scans and denial of service attacks. The obtained results have demonstrated the viability of the suggested methodology for the detection of anomalies related to malicious flows in computer networks. The proposed technique was implemented in the IPTRAF server, located in the Ravel laboratory and has been tested as a supporting toll to the management tasks of *Rede-Rio*.

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Símbolos

I_t	Sazonalidade no instante t, p. 18
L_t	Nível médio no instante t, p. 17
T_t	Tendência no instante t, p. 17
X_t	Valor da série temporal X no instante t, p. 17
α	Constante de suavização de nível médio, p. 17
γ	Constante de suavização de tendência, p. 17
σ	Constante de suavização de sazonalidade, p. 18
$\widehat{X}_t(h)$	Previsão feita no instante t para o valor da série temporal X no instante $t+h$, p. 17
d_t	Desvio da série temporal no instante t, p. 35
GB	*GigaByte*, p. 27
GHz	*GigaHertz*, p. 27
W	*Watt*, p. 27

Lista de Siglas

AWS	Amazon Web Services, p. 41
DDR	*Double Data Rate* - Taxa de dados dobrada, p. 27
DoS	*Denial of Service*, ou Negação de Serviço, p. 14
ERM	Erro Relativo Médio, p. 37
EWMA	*Exponential Weigthted Moving Averages*, p. 17
HTTP	*Hypertext Transfer Protocol* - Protocolo de Transferência de Hipertexto, p. 58
IETF	*Internet Engineering Task Force*, p. 5
IP	*Internet Protocol*, p. 6
KB	Kilobytes, p. 73
MB	MegaBytes, p. 44
Mbps	Mega bits por segundo, p. 41
NIST	*National Institute of Standards and Technology*, p. 19
RAID	*Redundant Array of Independent Disks* - Matriz redundante de discos independentes, p. 27
RFC	*Request for Comments*, p. 5
SDI	Sistema de Detecção de Intrusão, p. 19
TCP	*Transmission Control Protocol*, p. 5
UDP	*User Datagram Protocol*, p. 7

Capítulo 1

Introdução

A *Internet* tem se tornado cada vez mais presente na vida das pessoas. Ela é utilizada nos momentos de trabalho e de lazer, e tem por muitas vezes substituído as formas tradicionais de acesso a livros, filmes, músicas, notícias e de comunicações, fazendo assim parte das atividades corriqueiras de nossa vida.

E é nesse ambiente cada vez mais populado e movimentado pelo cidadão comum que temos percebido a presença de organizações privadas e estatais o transformando num belicoso campo de batalha, onde até mesmo governos como o da China e dos EUA tem criado ferramentas pra transformar os dispositivos de rede das pessoas e a infraestrutura de telecomunicações do país em armas cibernéticas, como por exemplo as iniciativas *Great Cannon* [1] da China e *Quantum* [2] utilizada pela agência de segurança nacional dos EUA. O ambiente virtual tratado com tamanha agressividade por estas entidades e com tamanha despreocupação pelo cidadão comum faz com que estejam em risco nossa privacidade e até mesmo nossa liberdade de expressão.

Em geral, o usuário comum da *Internet* não é preocupado com segurança, não utiliza criptografia, não utiliza *firewall* e não utiliza ferramentas específicas de monitoração de seus próprios recursos o que o torna suscetível a ser vitimado por organizações ou indivíduos maliciosos.

Da mesma maneira que um condomínio provê alguma segurança a seus condôminos através de câmeras, porteiros e seguranças, o método aqui proposto implementado e avaliado permite ao gerente de rede prover alguma segurança aos usuários de sua rede.

Este trabalho de forma alguma tem a pretensão de resolver todos os problemas de segurança e nem de garantir proteção total contra estas imensas ameaças, pois uma solução completa para este problema ao meu ver vai além de uma questão apenas tecnológica e passa por questões políticas onde o cidadão comum deve pressionar os seus governos para que eles o repeitem e de boicotes à empresas que violem seus dados com contratos abusivos e utilizem de maneira indevida de seus dados e dispositivos. Porém, o trabalho de dissertação aqui apresentado é capaz de em

alguns casos evitar tais problemas, amenizá-los de alguma forma e de até mesmo ajudar o gerente de rede a informar seus usuários que seus dispositivos podem estar sendo utilizados de maneira maliciosa.

O objetivo deste trabalho de dissertação é de propôr, implementar e avaliar uma ferramenta capaz de identificar eventos maliciosos na rede a partir do monitoramento da borda da rede. O monitoramento feito na rede deve ser capaz de capturar os dados referentes aos fluxos de rede que passam pela borda, o que faz com que o método não seja útil na detecção de ameaças de rede com origem e destino dentro da própria rede. A identificação de ameaças é realizada ao se identificar padrões anômalos em séries temporais, séries essas que são montadas com valores de métricas específicas obtidas através da observação de características dos fluxos monitorados na borda.

1.1 Contribuições

A primeira contribuição deste trabalho de dissertação é propôr, implementar e avaliar uma metodologia de detecção de anomalias em redes de computadores capaz de alertar de maneira automatizada um gerente de rede sobre a ocorrência de algum evento anômalo, bem como inferir se tal evento é malicioso e classificá-lo.

Para a detecção de anomalias este método propõe a utilização de um ferramental matemático que quando adicionado ao modelo de previsão em séries temporais de Holt-Winters consegue trazer estabilidade ao método, dado que o mesmo possui uma perda de estabilidade e exibe um comportamento sazonal distorcido quando são inseridos valores discrepantes em sua série de entrada. Esse ferramental é a segunda contribuição aqui prestada.

1.2 Organização do texto

No capítulo 2, é feita uma revisão bibliográfica sobre o modelo *TCP/IP*, fluxos, anomalias, séries temporais também serão apresentados os principais trabalhos relacionados que serviram de base para este trabalho.

No capítulo 3, onde é descrito o método proposto será apresentada e detalhada a maneira pela qual o método realiza a coleta e classificação dos fluxos de rede, bem como a contabilização das métricas a serem analisadas e a técnica utilizada para detecção de anomalias.

No capítulo 4, serão feitas discussões e análises referentes a características relevantes do método. Será também apresentada uma série de validações do método através da indução de ataques contra a rede a fim de avaliar a capacidade de detecção do método.

No capítulo 5, serão feitas as considerações finais e levantadas algumas perspectivas referentes a trabalhos futuros.

Capítulo 2

Fundamentação Teórica e Trabalhos Relacionados

Este capítulo apresenta a base teórica sobre a qual o trabalho é fundamentado. A primeira parte do capítulo tem por objetivo descrever o modelo de comunicação utilizado na *Internet*, bem como características de seus principais protocolos. A segunda e terceira partes deste capítulo definem fluxos e anomalias. Ainda na terceira parte deste capítulo se encontram algumas descrições e comentários sobre os tipos de anomalias e como elas podem ser observadas na rede. A quarta parte deste capítulo introduz alguns conceitos sobre séries temporais. A quinta parte deste capítulo apresenta alguns dos principais trabalhos relacionados à construção de métodos de detecção de anomalias, com destaque aos trabalhos que serviram de base para esta dissertação.

2.1 Modelo *TCP/IP*

De acordo com Kurose [3] , "*Internet* é uma rede de computadores que interconecta milhares de dispositivos computacionais ao redor do mundo.". A comunicação através da *Internet* só é possível porque todos os dispositivos a ela conectados seguem as mesmas regras para troca de informações. Esses conjuntos de regras são chamados de protocolos e são seguidos pelos dispositivos da rede com o objetivo de que haja entendimento das informações trocadas por todos os dispositivos interconectados. Kurose [3] , afirma que "Um protocolo define o formato e a ordem das mensagens trocadas entre duas ou mais entidades comunicantes bem como as ações realizadas na transmissão e/ou no recebimento de uma mensagem ou outro evento."

A instituição responsável pela padronização e especificação desses protocolos é o *Internet Engineering Task Force* (*IETF*) [4], que é uma grande comunidade aberta de projetistas de rede, operadores, fornecedores e pesquisadores preocupados com a

evolução da arquitetura da *Internet* e do bom funcionamento da mesma. Através de documentos *RFC* (*Request for Comments*) [5], são elaboradas as diretivas de desenvolvimento dos protocolos usados na *Internet*.

As tecnologias e arquiteturas dos dispositivos que se conectam à *Internet* são muito diversificadas, e incluem desde telefones celulares, *tablets*, computadores pessoais e até supercomputadores. Muitos são os fabricantes dos dispositivos que integram a rede, o que naturalmente traz algumas dificuldades no que tange a interoperabilidade e padronização, de acordo com Tanenbaum [6], a solução para essas dificuldades partem da redução da complexidade dos processo através da utilização de um modelo de camadas de protocolos, onde o propósito de cada camada é prover serviços para as camadas mais altas de maneira transparente. O modelo de camadas atualmente utilizado na *Internet* é o modelo *TCP/IP*, que é ilustrado na Figura 2.1.

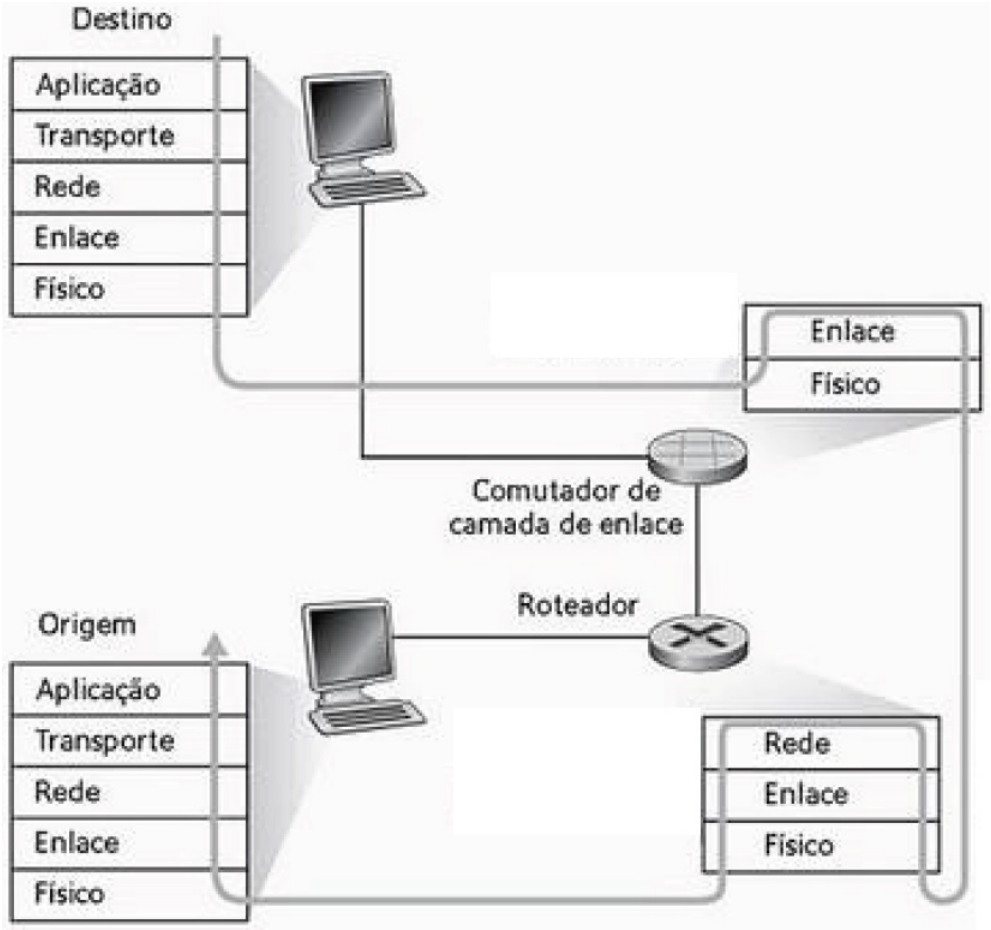

Figura 2.1: Camadas do Protocolo *TCP/IP* - Retirado de [3]

Tanenbaum [6], afirma que "A cola que mantém toda a *Internet* funcional é o protocolo de camada de rede, o *IP*(*Internet Protocol*)"e ao contrário do que acontece em outras camadas, o *IP* é o único protocolo utilizado na camada de rede pelo modelo *TCP/IP*. O *IP* foi projetado para conectar redes de redes trazendo assim uma ótima escalabilidade ao modelo *TCP/IP*, escalabilidade essa que é alcançada

devido a maneira como o *IP* endereça os elementos que fazem parte da rede e pela maneira que repassa os pacotes que trafegam pela rede.

Um endereço *IP* consiste em um número binário de quatro octetos, ou seja, 32 *bits*, sendo possível assim endereçar $2^{32} = 4.294.967.296$ elementos de rede. A notação mais comum usada para um endereço *IP* é do tipo *a.b.c.d*, onde *a*, *b*, *c* e *d* podem ser entendidos como números inteiros decimais de 0 a 255 de maneira que cada um desses números representa um dos octetos que fazem parte do endereço *IP*.

Um pacote *IP* pode ser dividido em duas partes: dados e metadados. A parte de dados contém informações que são repassadas à camada superior e a parte de metadados que é localizada no cabeçalho do pacote contém informações necessárias ao bom funcionamento do protocolo. As informações que fazem parte do cabeçalho são [3]:

- Versão - Esse campo possui quatro *bits* e especifica a versão do protocolo *IP* utilizada na transmissão do pacote. Essa informação é utilizada para interpretar o restante do pacote.

- Comprimento do cabeçalho - Como o cabeçalho do pacote *IP* pode conter um tamanho variável, seu tamanho é especificado neste campo.

- Tipo de Serviço - Este campo foi incluído no pacote *IP* com o objetivo de diferenciar os tipos de pacotes *IP*, como por exemplo, pacotes de telefonia, de aplicações de tempo real ou de aplicações comuns. Este campo geralmente é utilizado apenas pelo administrador da infraestrutura de rede.

- Comprimento do pacote - Neste campo é especificado o tamanho total do pacote *IP*.

- Identificador do pacote - O identificador de pacote serve para determinar a qual pacote um determinado fragmento pertence.

- *Flags* - Este campo possui três *bits*, o primeiro serve para indicar se um pacote pode ou não ser fragmentado, o segundo para indicar se há mais fragmentos de um determinado pacote e o terceiro não possui utilidade.

- Deslocamento de fragmentação - Este campo serve para informar a ordem dos fragmentos num determinado pacote.

- Tempo de vida - Este campo limita o número de saltos que um determinado pacote *IP* pode percorrer na rede.

- Protocolo de camada superior - Este campo identifica qual o protocolo de camada de transporte é utilizado na camada superior.

- Soma de verificação de cabeçalho - Este campo é utilizado para que o protocolo realize uma verificação de erros de *bit* no pacote.
- Endereço *IP* da fonte - Neste campo é indicado o remetente do pacote.
- Endereço *IP* de destino - Neste campo é indicado o destino do pacote.
- Opções - O campo de opções possui um tamanho variável e foi criado com o intuito de ampliar o cabeçalho do protocolo para que sejam possíveis a realização de atividades de administração de rede e desenvolvimento não consideradas durante o projeto do protocolo.

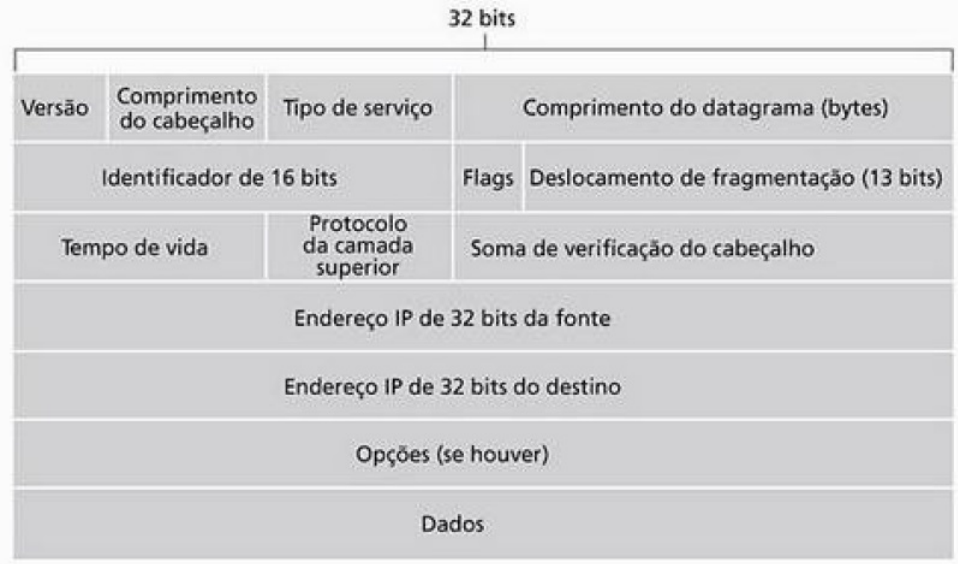

Figura 2.2: Pacote *IP* - Retirado de [3]

Usualmente, os protocolos de camada de transporte utilizados no modelo *TCP/IP* são os protocolos *UDP* e *TCP*. O protocolo *UDP* é mais simples, não realiza um estabelecimento de conexão prévia e não garante a entrega de mensagens. O *TCP* porém, é um protocolo mais complexo, que realiza um estabelecimento de conexão antes do início da transmissão dos dados e garante a entrega dos dados que lhe são entregues pela camada de aplicação para que sejam transmitidos.

Os pacotes *UDP* tem o formato explicitado na Figura 2.3 e possui os seguintes campos em seu cabeçalho:

- Porta da Fonte - Que especifica de qual porta do remetente foi enviado o pacote. As portas no protocolo *UDP* servem para endereçar a qual processo de camada de aplicação os dados devem ser entregues.
- Porta do destino - Que especifica qual porta do destinatário irá receber o pacote.

- Comprimento - Que especifica o tamanho do pacote *UDP*. O protocolo *UDP* utiliza apenas um campo para indicar o tamanho de seu pacote, diferentemente do protocolo *IP*. Isso se dá devido ao fato do protocolo *UDP* possuir um tamanho fixo de cabeçalho e o protocolo *IP* um cabeçalho de tamanho variável.
- Soma de verificação - Este campo é utilizado pelo protocolo para realizar verificação de erros.

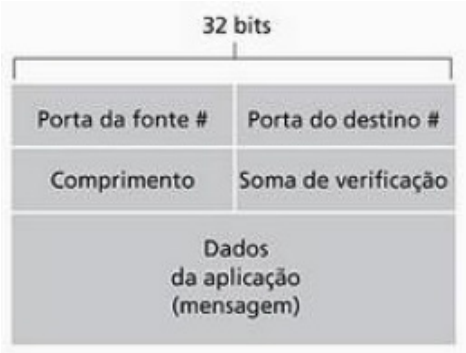

Figura 2.3: Pacote *UDP* - Retirado de [3]

O pacote *TCP* possui um cabeçalho maior do que o cabeçalho do pacote *UDP*, e isto acontece devido ao fato de o protocolo *TCP*, ao contrário do *UDP*, executar várias funções e possuir vários mecanismos de controle da transmissão, como por exemplo a confirmação de pacotes, retransmissão de pacotes, controle de fluxos e controle de congestionamento.

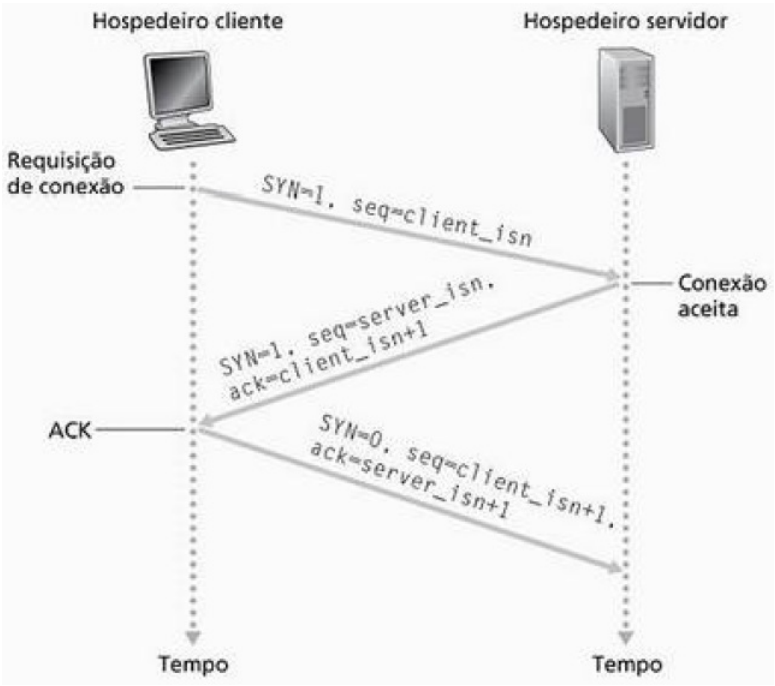

Figura 2.4: Estabelecimento de Conexão *TCP* - Retirado de [3]

O processo de estabelecimento de conexão, que também é chamado de *three way handshake* ou apresentação de três vias é um processo que tem por objetivo reservar recursos e sincronizar o protocolo para a realização da transmissão. O estabelecimento de conexão possui três etapas [3], conforme mostrado na Figura 2.4. Na primeira etapa o elemento de rede que inicia a transmissão envia uma requisição de conexão que é sinalizada através da ativação da *flag SYN* no cabeçalho do pacote *TCP*. Na segunda etapa, o elemento de rede destinatário da requisição inicial responde com um pacote que possui a confirmação do recebimento do pacote anterior e a sinalização de estabelecimento de conexão ativa. A terceira e última etapa, consiste na confirmação do pacote enviado da segunda etapa, realizado pelo elemento de rede que iniciou a transmissão. É importante notar que nesta última etapa a *flag SYN* tem seu valor marcado como zero.

Um pacote *TCP* pode ser dividido em duas partes. A primeira é o cabeçalho, que possui os metadados que são utilizados pelo protocolo para o correto funcionamento de todos os seus mecanismos, e a segunda parte é o campo de dados que é entregue ao protocolo de camada de aplicação que utiliza os serviços do *TCP*. Os campos [3] do cabeçalho *TCP* são:

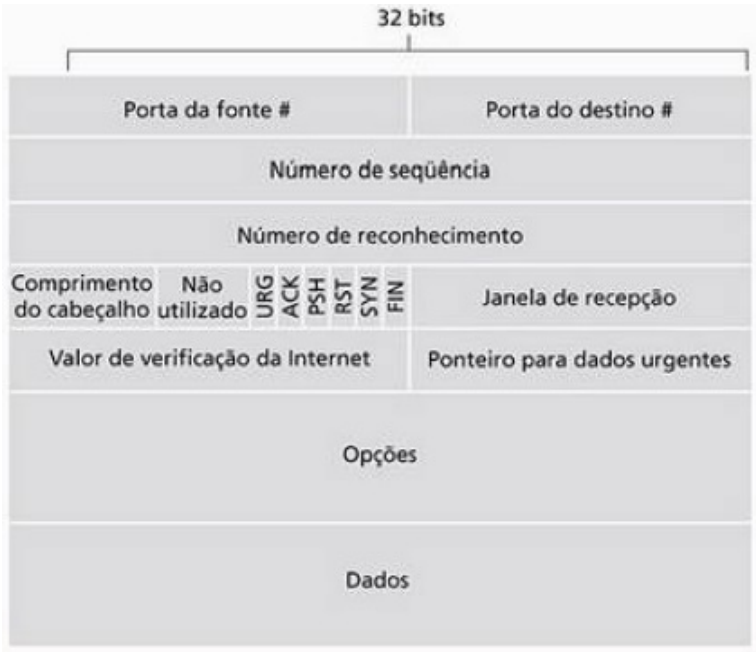

Figura 2.5: Pacote *TCP*, retirado de [3]

- Porta de fonte e destino - Os números de porta fonte e destino são utilizados para endereçar os pacotes aos seus respectivos processos na camada de aplicação.
- Número de sequência e Número de reconhecimento - São dois valores de 32 *bits* cada que são utilizados para implementação do serviço de transferência

confiável do *TCP*. O número de sequência identifica os pacotes e o de reconhecimento os identifica como recebidos.

- Comprimento do cabeçalho - Este campo serve para indicar o tamanho do cabeçalho do pacote, dado que o campo de Opções é opcional e de tamanho variável.

- Janela de recepção - Esse campo é utilizado no controle de fluxo do *TCP* e serve para indicar a quantidade de *bytes* que o receptor está disposto a aceitar.

- Valor de verificação da *Internet* - Este valor é utilizado para uma verificação de erros do pacote *TCP*, semelhante a verificação do protocolo *UDP*.

- Ponteiro para dados urgentes - Este campo serve para indicar a existência e duração da transmissão de alguns dados urgentes e são utilizados apenas quando a *flag URG* está ativa [6].

- Opções - Campo opcional e de tamanho variável, utilizado apenas em algumas aplicações específicas.

- *Flags* - O campo de *Flags* contém seis *bits*, sendo um para cada *flag* (também chamado de indicador, ou sinalizador). O primeiro *bit* é o *URG* que é utilizado para sinalizar que existem dados nesse pacote que o protocolo de camada superior marcou como urgentes. O segundo *bit* é o *ACK* que é utilizado para indicar a existência do reconhecimento do recebimento de algum pacote no campo reservado para tal no pacote atual. O terceiro *bit* é o *PSH*, que serve para indicar que o destinatário deve repassar imediatamente os dados para a camada superior. O quarto *bit* é o *RST* que serve para reinicializar uma conexão em que ocorreu algum erro. O *RST* também é utilizado para rejeitar algum pacote inválido ou recusar a tentativa de abertura de uma conexão. O quinto *bit* é o *SYN* que é utilizado para indicar que existe um estabelecimento de conexão ocorrendo como já foi citado anteriormente. o sexto e último *bit* é o *FIN* que é utilizado para indicar um pedido de encerramento de conexão.

2.2 Fluxos

Um fluxo de pacotes de rede em uma conexão *TCP/IP*, pode ser entendido como o conjunto de pacotes enviados numa transmissão desde sua abertura até seu encerramento.

Com o protocolo *TCP* um fluxo é o conjunto de todos os pacotes enviados entre um determinado par de endereços *IP* e portas *TCP* e que são enviados entre um

pacote de estabelecimento de conexão (pacote marcado com a *flag SYN*) e um pacote de fechamento de conexão (pacote marcado com a *flag FIN*).

Com o protocolo *UDP* um fluxo é o conjunto de todos os pacotes enviados entre um determinado par de endereços *IP* e portas *UDP* e que são enviados num determinado período de tempo determinado pela aplicação coletora.

Diversas são as aplicações coletoras de fluxos existentes, entre as mais conhecidas estão a *J-Flow* [7] que é uma solução proprietária da Juniper, *NetFlow* [8] que é uma solução proprietária da Cisco e *sFlow* [9] que é uma solução proprietária de um conjunto de empresas, entre elas a *Alcatel-Lucent*. Na elaboração deste trabalho foi usada a tecnologia *NetFlow*. Os motivos pelos quais esta ferramenta foi escolhida e seu funcionamento são detalhados na Seção 3.1.

2.3 Anomalias

Anomalias podem ser definidas como qualquer tipo de evento que venha a se afastar do padrão observado de um determinado objeto de estudo. De igual forma, em redes de computadores pode-se dizer que existe uma anomalia quando ocorre um evento que se diferencia do comportamento padrão de determinada característica da rede [10].

Quando é abordado o tema de anomalias em redes de computadores, pode-se estar tratando de uma definição com o escopo muito amplo. Por exemplo, a queda de energia numa localidade com um roteador importante de um determinado provedor pode causar uma anomalia na rede deste provedor, bem como um rompimento dos meios de transmissão da rede, um erro de protocolo de roteamento ou até mesmo um ataque feito por usuários mal-intencionados ou infectados por algum tipo de vírus de computador.

Ao se tratar de tráfego, pode-se caracterizá-lo de duas maneiras [10]. A primeira delas é caracterizar o tráfego como comum e assim não gerar nenhum alarme ou aviso. A segunda maneira de caracterizar o tráfego é considerá-lo como anômalo, levantando assim suspeitas e motivos para uma análise mais profunda de sua ocorrência.

A caracterização do tráfego é relacionada diretamente à maneira de observá-lo, dado que a definição de tráfego anômalo é o tráfego que possui alguma anomalia, ou seja, o tráfego que de alguma maneira foge ao padrão observado da rede. E isto leva a uma reflexão sobre a maneira pela qual as características da rede são observadas e a quais características observar.

Uma das maneiras de se observar a rede é utilizando-se de diagramas de dispersão [10] conforme mostra a Figura 2.6, onde é exibido no eixo y as portas de destino do protocolo *TCP* de pacotes que chegam a um determinado servidor de um *site* e no

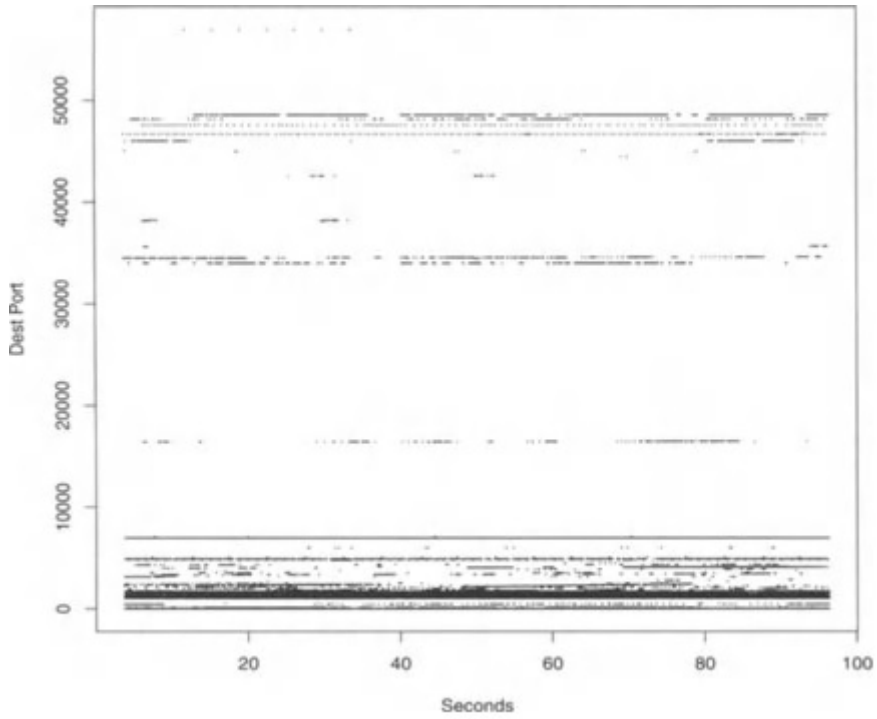

Figura 2.6: Diagrama de Dispersão - Portas de Destino X segundos - Retirado de [10]

eixo x o tempo do experimento.

Provavelmente a maneira mais popular de se entender o tráfego é através da observação do total de *bits* que passam por determinado enlace. Na Figura 2.7 pode-se observar um exemplo retirado de um dos enlaces de borda da RedeRio de computadores [11], onde é mostrado no eixo y o valor médio de bits por segundo durante trinta minutos, e no eixo x pode ser observado o tempo no qual foi feita a medida.

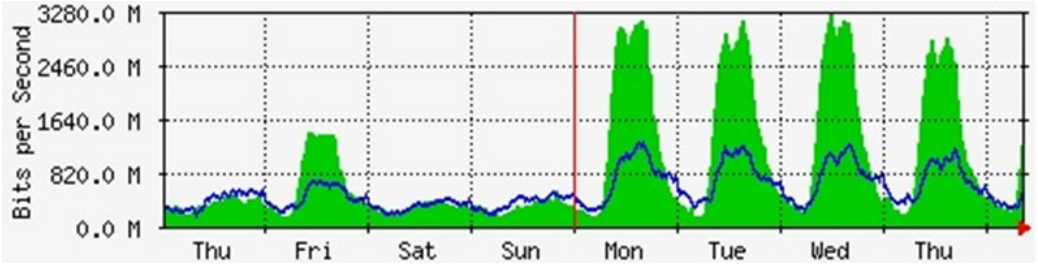

Figura 2.7: Exemplo de observação de tráfego total - Retirado de [11]

Também é possível observar o tráfego da rede considerando os tipos de fluxos que são comutados através de um determinado enlace. Através da ferramenta IPTRAF [13], os fluxos que passam nos roteadores de borda da RedeRio de computadores [11] são coletados e exibidos. Uma dessas exibições é mostrada na Figura 2.8, onde observa-se no eixo x o tempo onde o dado foi coletado e no eixo y a média de bits por

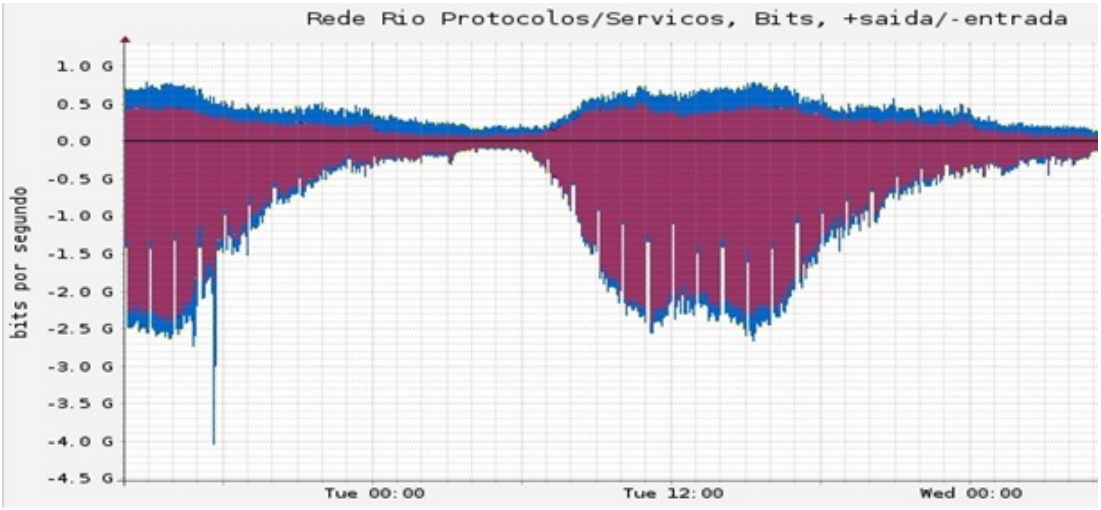

Figura 2.8: Exemplo de observação de fluxos do tráfego total - Retirado de [12]

segundo feita a cada cinco minutos. Note que o eixo y é dividido em duas partes, onde a parte superior significa o valor da média obtida analisando-se os fluxos que saem da rede e a parte inferior é obtida analisando-se os fluxos que entram na rede. As cores do gráficos representam os protocolos de camada de transporte utilizados pelos fluxos, sendo a cor azul representante do *UDP* e o roxo sendo a cor representante do *TCP*.

Foi proposta uma visualização do tráfego por Cláudia de Abreu no seu trabalho de dissertação [14], onde objetiva-se a detecção de anomalia através de uma visualização gráfica de características da rede. A Figura 2.9 mostra um exemplo da visualização proposta. Os dados exibidos na Figura 2.9 também tem origem na coleta dos roteadores de borda da RedeRio realizada pelo servidor IPTRAF. Este gráfico é gerado a cada cinco minutos e considera apenas os fluxos coletados nos cinco minutos anteriores ao momento em que foi gerado. Na Figura 2.9 o eixo y representa a porta de destino para qual um determinado fluxo foi enviado, o eixo x representa o *IP* de origem a partir do qual um determinado fluxo foi enviado e no eixo z é mostrado o *IP* de destino para qual um determinado fluxo foi enviado. Os símbolos que são exibidos no gráfico também tem um significado importante mas serão explicitados na Seção 2.5.1.

2.3.1 Ataques

Nessa seção será feita uma descrição de técnicas e categorias de ataques a elementos de redes. Junto com suas descrições serão citados alguns dos mais comuns exemplos de ataques, sendo impossível citar todos, pois novas e diferentes técnicas de ameaças vem surgindo a cada dia.

O ato de mascarar um *IP*, também chamado de *IP Spoofing*, consiste em um

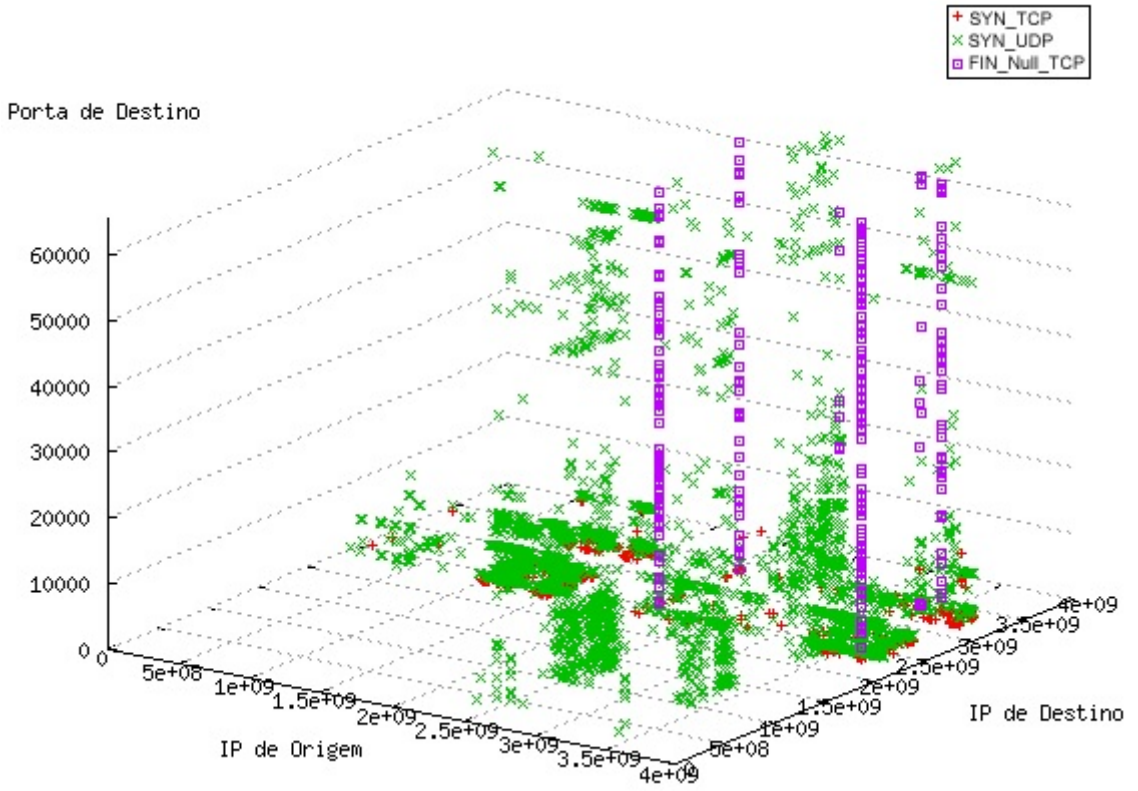

Figura 2.9: Exemplo de observação multi-dimensional de fluxos - Retirado de [15]

determinado elemento da rede atribuir qualquer endereço *IP* ao campo *IP* de fonte, que não seja realmente o seu endereço *IP*. Em outras palavras, mascarar um *IP* consiste em falsificar a "identidade"de remetente do pacote. Esta é uma prática comum ao se realizar ataques e tem o intuito de dificultar que mecanismos de defesas da vítima venham a interromper ou evitar o ataque. Qualquer ataque que se utilize dessa técnica não pode necessitar do recebimento de algum pacote, ou estabelecimento de alguma conexão com a vítima, visto que qualquer resposta dada pela vítima jamais chegará ao atacante.

Negação de Serviço ou *DoS* (*Denial of Service*), é um tipo de ataque, onde um atacante tenta tornar indisponíveis os recursos de um sistema, através de uma demanda forjada acima da capacidade do sistema[16]. Quando esse ataque é feito por mais de um atacante é chamado de *DDoS* (*Distributed Denial of Service*). Muitas técnicas são utilizadas para a realização deste tipo de ataque [17], dentre elas:

- *Smurf Attack* – Esse tipo de ataque tem por objetivo consumir os recursos de rede da vítima, e consiste em enviar requisições *broadcast* do protocolo de rede *ICMP* com o endereço *IP* de fonte mascarado.

- *TCP SYN Flood* – Esse ataque tem por objetivo consumir os recursos de memória e de processamento da vítima. O método consiste no envio de várias requisições de conexão para a vítima, que reserva espaço de memória para

a troca de pacotes. Neste método, o atacante não completa a abertura das conexões requisitadas de maneira a maximizar o uso de recursos por parte da vítima e minimizar o uso de recursos utilizados pelo atacante. Como não é necessário que haja resposta às requisições, por vezes, o *IP* do atacante se encontra mascarado.

- *TCP Connect Flood* – Este ataque é semelhante ao *TCP SYN Flood*, porém as conexões são estabelecidas de maneira completa e em alguns casos são mantidas abertas por longos períodos de tempo. Este ataque tende a exigir mais recursos por parte do atacante, dado que é necessária a alocação de memória para cada abertura de conexão e respostas aos pacotes de abertura de conexão, porém se mostra mais difícil de mitigar, pois suas conexões aparentam ser legítimas, especialmente quando as conexões são mantidas abertas. Neste tipo de ataque é impossível que o *IP* do atacante seja mascarado, pois caso contrário seria impossível que a conexão fosse completada. Devido a estas características, esse tipo de ataque de negação de serviço geralmente é realizado por máquinas de indivíduos sem objetivos maliciosos sendo mobilizadas através de algum tipo de vírus de computador.

- *UDP floods* – Esse ataque tem por objetivo consumir os recursos de rede da vítima, e consiste no envio em larga escala de pacotes *UDP* que podem ter tamanho de até 65.000 *bytes*.

Escaneamento de porta ou *Port Scanning*, é a primeira fase de um ataque planejado, fase esta, onde é feito o reconhecimento das vulnerabilidades em um determinado sistema. Várias técnicas [18] [19] são utilizadas nesse tipo de ataque e algumas serão citadas abaixo.

- *TCP SYN Scan* – Neste escaneamento, um determinado atacante envia vários pacotes de conexão *SYN*, que são pacotes usados para o estabelecimento de conexão em várias portas. Quando a conexão é respondida pelo alvo com um SYN/ACK, significa que a porta está aberta.

- *TCP Connect Scan* – Muito parecido com o *SYN Scan* citado anteriormente, mas, nas portas onde o atacante identifica uma porta aberta, ele estabelece a conexão completamente e depois a finaliza. O atacante tem essa abordagem justamente para dificultar a identificação do ataque.

- *FIN Scan* – Neste tipo de escaneamento, um determinado dispositivo mal-intencionado, envia vários pacotes com o identificador *FIN* ativado, ou seja, pacotes de finalização de conexão. Por padrão, elementos de rede seguindo o protocolo *TCP* devem responder com um pacote possuindo o indicador *RST*

ativado quando a porta está fechada e simplesmente ignorar quando a porta está aberta. Através desse método o atacante pode inferir se existe algum elemento em um determinado endereço *IP* e quais portas estão abertas.

2.4 Séries Temporais

Uma série temporal consiste [20] em um conjunto de valores, onde cada valor tem a si atrelado um indicador do tempo onde esse valor ocorreu ou foi observado. Bons exemplos de séries temporais seriam a velocidade de uma determinada partícula, onde o valor observado é a distância percorrida num determinado índice de tempo, ou o valor de mercado de uma determinada empresa, onde o valor observado é o preço de suas ações.

As séries temporais podem de maneira geral ser classificadas como contínuas ou discretas [20]. Séries temporais contínuas são utilizadas em abordagem mais teóricas, pois para que se utilize intervalos de tempo infinitesimais é necessário que se utilize infinitas medidas de tempo a fim de se obter valores num intervalo de tempo qualquer [21], portanto neste trabalho serão consideradas as séries temporais discretas. As séries temporais discretas podem ser descritas de três maneiras possíveis:

- Por amostragem de uma série temporal de natureza contínua. Por exemplo, a temperatura de um ambiente medida em intervalos de 15 minutos.
- Por agregação num período de tempo. Por exemplo, o total de vendas realizadas por uma empresa medido mensalmente.
- Séries de natureza discreta. Por exemplo, dividendos pagos por uma empresa em anos sucessivos.

Muitas são as aplicações práticas dos estudos de séries temporais [20], entre elas está a descrição do comportamento de uma determinada variável através de métodos gráficos e da obtenção de valores estatísticos. Com um bom entendimento do comportamento pode-se realizar uma modelagem da série que pode ser univariada, ou seja, que leva em consideração apenas os valores passados da série, ou multivariada, que leva em considerações o valor de outras séries ou variáveis. Pode-se também fazer previsões sobre o futuro comportamento ou valor de uma determinada série. Através de bons modelos e previsões é possível também fazer controle de um determinado processo, observando uma determinada série temporal e interagindo com mecanismos que se relacionam com a série com a intenção de se atingir determinado objetivo.

2.4.1 Suavização Exponencial Simples

A suavização exponencial, ou *EWMA* (*Exponential Weigthted Moving Averages*) é um dos mais conhecidos métodos de previsão e a partir do qual veio a surgir muitas variações, algumas das quais serão abordadas na próximas seções.

A suavização exponencial simples consiste na ideia de se obter uma média ponderada, onde os valores recentes tem um peso maior em relação aos mais antigos e isso se dá através de uma relação exponencial [21]. A suavização exponencial simples pode ser calculada através da Equação 2.1 abaixo:

$$\overline{X}_t = \alpha X_t + \alpha(1-\alpha)X_{t-1} + \alpha(1-\alpha)^2 X_{t-2} + ... + \alpha(1-\alpha)^i X_{t-i} + ... \qquad (2.1)$$

Onde X_t é o valor da série temporal X no instante t, $\overline{X}_t$ é denominado de valor exponencialmente suavizado e α é a constante de suavização. Vale notar que $0 < \alpha < 1$.

Na suavização exponencial simples, é considerado como valor previsto o último valor suavizado exponencialmente, ou seja: $\widehat{X}_t(h) = X_t$, sendo $\widehat{X}_t(h)$ a notação utilizada para os valores previstos da série X realizada no instante t para h períodos a frente. Substituindo na Equação 2.1 tem-se:

$$\widehat{X}_t(h) = \alpha X_t + (1-\alpha)\widehat{X}_{t-1}(h) \qquad (2.2)$$

Observe que na Equação 2.2 acima não se utiliza todos os valores da série temporal desde o início da observação até o instante t. Essa equação é chamada de equação de atualização de previsão [21] e permite que seja usado apenas o valor previsto anteriormente, o valor medido da série e o parâmetro de suavização utilizado na previsão.

2.4.2 Tendência Linear de Holt

O método de suavização exponencial simples quando aplicado a uma série que apresenta uma tendência linear crescente ou decrescente, irá fornecer previsões que subestimam ou superestimam continuamente os valores observados [21]. Então é proposto o método de suavização exponencial com tendência linear de Holt, que consiste em obter uma estimativa da tendência da série, utilizando também uma suavização exponencial [20]. A tendência é obtida segundo a Equação 2.3 abaixo.

$$T_t = \gamma(L_t - L_{t-1}) + (1-\gamma)T_{t-1} \qquad (2.3)$$

Onde, T_t representa a tendência da série temporal no instante t, L_t representa

o nível médio da série temporal no instante t e γ é a constante de suavização utilizada para cálculo da tendência. O nível médio é calculado de maneira semelhante ao método de suavização exponencial simples, mas levando em conta o fator de tendência. O nível médio é calculado conforme mostra a Equação 2.4 abaixo.

$$L_t = \alpha X_t + (1 - \alpha)(L_{t-1} + T_{t-1}) \tag{2.4}$$

O valor previsto para a série temporal no instante t, buscando-se h períodos de tempo adiante se dá pelo nível médio no instante t acrescido da tendência calculada no instante t multiplicada pela quantidade de períodos k que deseja-se prever, conforme mostra a Equação 2.5 abaixo:

$$\widehat{X}_t(h) = L_t + hT_t \tag{2.5}$$

2.4.3 Método de Holt-Winters

A suavização exponencial pode ser também utilizado em séries temporais que possuem uma variação sazonal. Para tais séries pode-se utilizar o método de Holt-Winters que utiliza do índice de sazonalidade para adaptar o método a eventuais variações periódicas [20]. De acordo com Morettin [21], o método de Holt-Winters é mais adequado à análise de séries mais gerais apesar de trazer dificuldades para determinar as constantes de suavização e de estudo das propriedades estatísticas.

O método de Holt-Winters assume duas formas de utilização do índice de sazonalidade, que pode ser multiplicativo ou aditivo [21], se fazendo útil a observação do comportamento da série para a escolha da abordagem. A notação escolhida aqui para o índice de sazonalidade é I_t.

O método de Holt-Winters com abordagem multiplicativa consiste em:

$$\widehat{X_t}(h) = (L_t + hT_t)I_{t-p+h} \tag{2.6}$$

$$L_t = \alpha(\frac{X_t}{I_{t-p}}) + (1 - \alpha)(L_{t-1} + T_{t-1}) \tag{2.7}$$

$$T_t = \gamma(L_t - L_{t-1}) + (1 - \gamma)T_{t-1} \tag{2.8}$$

$$I_t = \sigma(\frac{X_t}{L_t}) + (1 - \sigma)I_{t-p} \tag{2.9}$$

O método de Holt-Winters com abordagem aditiva consiste em:

$$\widehat{X}_t(h) = L_t + hT_t + I_{t-p+h} \tag{2.10}$$

$$L_t = \alpha(X_t - I_{t-p}) + (1-\alpha)(Lt-1+Tt-1) \tag{2.11}$$

$$T_t = \gamma(L_t - L_{t-1}) + (1-\gamma)(T_{t-1}) \tag{2.12}$$

$$I_t = \sigma(X_t - L_t) + (1-\sigma)I_{t-p} \tag{2.13}$$

Onde p é o valor que define quantos intervalos de tempo possui o período de sazonalidade, σ é a constante de suavização referente à sazonalidade e $(t-p+h)$ se refere ao índice de sazonalidade no instante de tempo exatamente um período anterior ao instante de tempo que é o alvo da previsão. Note que para se realizar previsões para instantes de tempo maiores que o período de sazonalidade, deve-se utilizar $(t-(cp)+h)$, sendo c o menor numero inteiro onde, $t > (t-(cp)+h)$ [21].

Os valores iniciais do método de Holt-Winters com a abordagem multiplicativa são [21]:

$$L_p = \frac{\sum_{i=0}^{p} X_i}{p} \tag{2.14}$$

$$T_p = 0 \tag{2.15}$$

$$I_i = \frac{X_i}{L_p}, i = 1, 2, ..., p \tag{2.16}$$

Os valores iniciais do método de Holt-Winters com a abordagem aditiva são calculados de igual forma do método com abordagem multiplicativa, alterando-se apenas a maneira que se calcula os valores iniciais da sazonalidade, que não serão calculados segundo a Equação 2.15, mas sim segundo a Equação 2.17 abaixo [22]:

$$I_i = X_i - L_p, i = 1, 2, ..., p \tag{2.17}$$

2.5 Trabalhos Relacionados

Sistemas de Detecção de Intrusão, ou *SDI,* são ferramentes muito usadas na área de segurança de redes. Existem muitos tipos de *SDI's* com metodologias diferentes e diversas nomenclaturas lhes são propostas [23] [24] [10]. Neste trabalho serão adotadas as nomenclaturas propostas pelo *NIST* em [25] e por Mukherjee *et al.* em [26].

A nomenclatura proposta por [26] definem dois tipos de arquiteturas para modelagem dos *SDI's*, que são:

- Baseados em Rede - Que monitoram a rede ou algum segmento da rede através de um ou mais dispositivos, geralmente num *firewall* ou roteador de borda.

- Baseados em Dispositivos - Que monitoram apenas informações pertencentes ao dispositivo e referentes a ele, como *logs* dos sistemas operacionais e o tráfego de rede.

A nomenclatura proposta por [25] define três tipos de metodologias de detecção usadas pelos *SDI's*, que são:

- Detecção baseada em assinatura - Nesse tipo de detecção, compara-se algum padrão observado na rede com uma biblioteca de padrões de ameaças conhecidas. *SDI's* com detecção baseada em assinatura tendem a ser muito eficientes para evitar ameaças já conhecidas, porém ineficientes para evitar novas ameaças.

- Detecção baseada em anomalia - Ao utilizar esse tipo de detecção, o *SDI* compara definições de atividades consideradas normais com os eventos observados com o objetivo de identificar desvios significativos. Os perfis considerados normais nesse tipo de detecção são geralmente definidos de maneira dinâmica, observando determinadas atividades, protocolos ou aplicações por um período de tempo ou até mesmo de maneira contínua. O maior benefício de metodologias de detecção baseadas em anomalia é a maior capacidade de detecção de ameaças ainda não conhecidas se comparadas com metodologias não baseadas em anomalias. Uma das maiores dificuldades encontradas por metodologias desse tipo é garantir que fiquem de fora do padrão de normalidade da rede quaisquer atividades maliciosas que possam ocorrer durante o período de treino do método.

- Protocolos de análise de estados - Esta metodologia consiste em comparar predeterminados padrões que geralmente são aceitos como atividade comum de um protocolo e comparam com o padrão observado no momento. Diferentemente dos *SDI's* baseados em detecção de anomalias que dependem apenas de informações coletadas, seja do dispositivo ou da rede, os *SDI's* baseados em análise de estados dependem fortemente de informações e perfis dos protocolos ou aplicações, bem como detalhes de implementação e particularidades dos fabricantes. Os protocolos de análise de estado são capazes de entender e rastrear o funcionamento da rede e das aplicações das quais ele foi preparado para proteger.

Existe uma grande variedade de *SDI's* propostos na literatura e disponíveis no mercado, usando diversas técnicas e metodologias. Neste trabalho o foco será dado nos *SDI's* baseados em rede e com detecção baseada em anomalia, pelo fato da metodologia proposta nesse trabalho se tratar de um *SDI* pertencente a estas categorias.

Considerando os *SDI's* com detecção baseada em anomalias pode-se encontrar métodos que utilizam de diversas abordagens. O trabalho de Mahoney [27] consiste em cinco propostas de detecção de ataques em redes de computadores utilizando detecção de anomalias através de técnicas de *Machine Learning*, onde cada uma dos algoritmos utiliza diferentes técnicas de aprendizados e os compara em ambientes onde são gerados uma série de ataques.

Os trabalhos [28] e [29] avaliam e exploram as características de auto-similaridade observadas na rede para realizar a detecção de anomalias. O trabalho [28] considera o tráfego da rede a fim de estimar o parâmetro de Hurst que mede o grau de auto-similaridade e através de sua variações busca encontrar anomalias periódicas na rede. O trabalho [29] utiliza além do parâmetro de Hurst, um método de priorização de tráfego com o objetivo de diminuir a característica de rajadas da rede e assim facilitar a detecção de anomalias.

Através da utilização de um filtro estatístico de agregação de dados e de uma rede neural, o trabalho [30] realiza a detecção e classificação de anomalias. O filtro de agregação estatístico proposto em [30] provê a entrada de 22 características relevantes que são apresentadas à rede neural de duas camadas, com 22 entradas e uma saída. Analisando-se o valor desta saída pode-se inferir se o fluxo é ou não malicioso.

O trabalho [31] reúne um conjunto de seis informações retiradas dos fluxos de rede, que é enviado para um módulo de previsão de comportamento e para um módulo de detecção de anomalia. Através de análises estatísticas que inferem o distanciamento do comportamento padrão levando-se em conta o desvio padrão, o método proposto em [31] realiza a detecção de anomalias.

Os trabalhos de Wang *et.al* [32] e De Moura [33] realizam a detecção de anomalias através da variação da entropia de informações retiradas de fluxos de rede. No trabalho de Wang *et.al* [32] a entropia é calculada tendo-se como base 4 métricas formadas a partir de informações coletadas dos fluxos que tem ligação direta com eventos maliciosos, como por exemplo, a não chegada de pacotes de reconhecimento, o número de pacotes com a *flag RST* ativa, entre outros. De Moura [33] utiliza uma abordagem diferente, avaliando a entropia obtida a partir dos parâmetros de *IP* e porta de destino e de origem dos fluxos que trafegam pela rede. Em seguida, De Moura [33] utiliza o método de séries temporais de Holt-Winters para avaliar quais as variações dos valores da entropia podem ser consideradas anomalias.

Nas seções subsequentes serão descritos de maneira mais detalhada três trabalhos que serviram diretamente de base para a construção do método que será aqui proposto.

2.5.1 Detecção de Anomalias Através de uma Visualização Tridimensional dos Fluxos de Rede

Este trabalho [14] foi realizado por Cláudia de Abreu Silva e propõe uma metodologia para a elaboração de uma ferramenta para detecção de anomalias [15]. No início do trabalho ela introduz características dos ataques, *malwares* e *worms* mais comuns e o seu reflexo no padrão de comportamento do tráfego na rede. Em seguida é apresentada uma proposta de classificação e filtro dos fluxos de pacotes de rede, classificação esta que serve de base para a classificação apresentada neste trabalho.

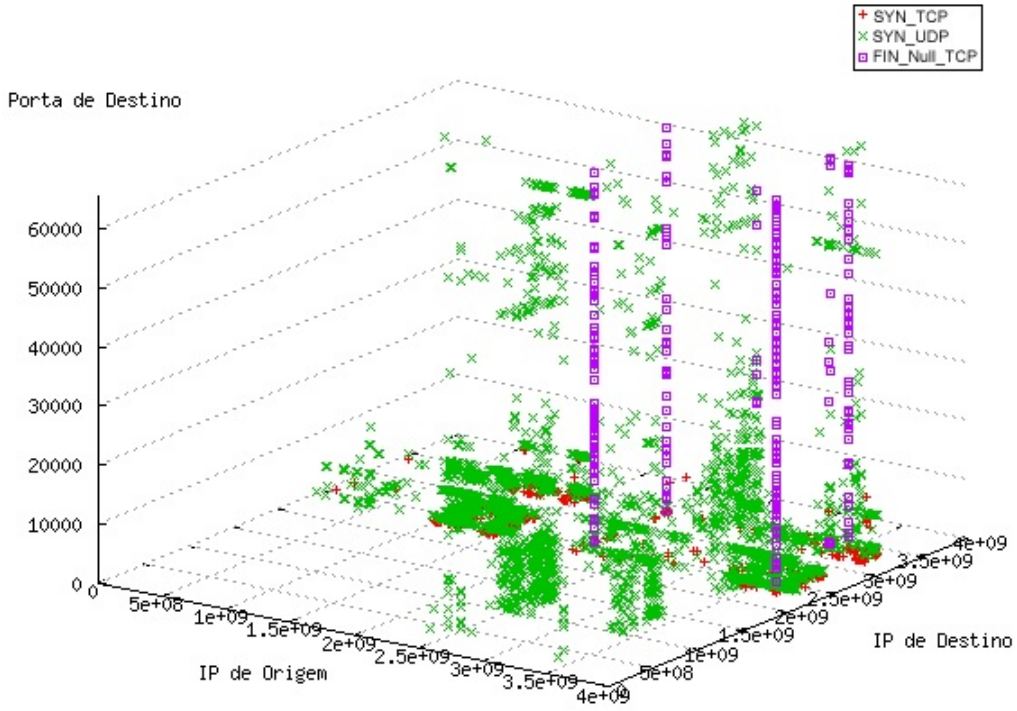

Figura 2.10: Exemplo de de detecção de anomalias por exibição proposta por [14]

A classificação proposta em [14] tem por objetivo isolar algumas anomalias geradas por ataques, *malwares* e *worms*. A proposta consiste em definir os seguintes conjuntos:

- *SYN TCP* - Um fluxo é dito pertencente a esta classificação se a porta de origem for maior que 1023, a porta de destino estiver na lista de portas suspeitas construída pelo método e possuir a *Flag TCP SYN* ativa.
- *SYN UDP* - Um fluxo é dito pertencente a esta classificação se a porta de origem for maior que 1023, a porta de destino estiver na lista de portas suspeitas e o protocolo usado na transmissão for o protocolo *UDP*.
- *SYN Half Open* - Um fluxo é dito pertencente a esta classificação se a porta de origem for maior que 1023, e possuir as *Flags TCP RST* e *TCP ACK* ativas.

- *FIN Null TCP* - Um fluxo é dito pertencente a esta classificação se o protocolo utilizado for o *TCP* e possuir a *Flag TCP RST* ativa.

- *SYN TCP* Porta Baixa - Um fluxo é dito pertencente a esta classificação se o protocolo utilizado for o *TCP* e possuir as *flags TCP RST* e *TCP ACK* ativas.

Para a detecção de anomalias ela propõe uma metodologia baseada em uma visualização do tráfego que é gerada pelo método. Um exemplo dessa visualização é mostrado na Figura 2.10. Em seu trabalho ela explica como identificar anomalias utilizando essa visualização e também disserta sobre análises feitas do método e de sua classificação.

2.5.2 Detecção de Anomalias Através da Análise da Variação do Tráfego

O trabalho de Brutlag [34] foi realizado com o objetivo de detectar comportamentos anormais em redes de computadores. A metodologia utilizada consiste em prever valores de tráfego para um período de tempo no futuro e comparar com o valor medido, caso esses valores apresentem um desvio muito grande é considerado que houve uma anomalia. O método de previsão em séries temporais utilizado é Holt-Winters.

Neste trabalho foi desenvolvida uma métrica de desvio esperado, onde o autor propõe a realização de uma suavização exponencial nos desvios medidos ao longo do tempo, utilizando a mesma constante de suavização utilizada no cálculo da sazonalidade, conforme mostrado na Equação 2.18 abaixo:

$$d_t = \sigma \mid X_t - \widehat{X} \mid + (1 - \sigma) d_{t-p} \tag{2.18}$$

Logo em seguida ele define o intervalo $(\widehat{X}_t(1) - \delta d_{t-p}, \widehat{X}_t(1) + \delta d_{t-p})$, com δ sendo um fator de escala usualmente entre 2 e 3. Caso algum valor observado X_t caia fora do intervalo é considerado como anomalia.

O autor elabora uma ferramenta para validação de seu método fazendo uso do pacote RRDtool [35], onde mostra a eficácia de sua detecção com alguns elementos gráficos e análises.

2.5.3 Detecção de Anomalias Através da Observação do Comportamento em Quatro Métricas

O trabalho de Nguyen et. al [36] consiste numa proposta de detecção de anomalias utilizando monitoramento de fluxos. Nesta proposta são extraídas quatro

métricas dos fluxos que são capturados da rede. Cada métrica tem por objetivo levantar informações sobre características comuns a anomalias geradas por ataques. As métricas definidas tem por principal foco ataques de *DoS* e *PortScan*, e são definidas por:

- *TOTAL BYTES* - Volume total dos fluxos capturados em *bytes*.
- *TOTAL PACKETS* - Quantidade total de pacotes observados nos fluxos capturados.
- *DSocket* - Quantidade de fluxos com volumes similares, mesmo endereço *IP* de destino e mesma porta de destino.
- *DPort* - Quantidade de fluxos com volumes similares, mesmo endereço *IP* de origem e destino, podendo ser endereçados a portas diferentes.

Logo após as métricas serem capturadas é elaborada uma sequência numérica para cada métrica com os valores obtidos, como essa sequência é indexada pelo tempo, formam-se quatro séries temporais. O autor utiliza o método de previsão de valores em séries temporais Holt-Winters de maneira a determinar se o valor capturado é alto ou baixo pro padrão que se espera observar de uma determinada série temporal na rede. Comparando o comportamento observado e valores tabelados, pode-se afirmar se está ou não ocorrendo alguma anomalia na rede e qual seria o tipo de anomalia.

Capítulo 3

Método Proposto

O método proposto neste trabalho, consiste em um *SDI* baseado em redes e com detecção baseada em anomalias que tem como principal objetivo a detecção de ataques de negação de serviço e de escaneamento de porta, mas mantendo a capacidade de identificação de anomalias mais genéricas.

Pode-se entender o método como um processo de quatro etapas, que são a coleta, a classificação, a obtenção de métricas e a análise. As quatro etapas serão abordadas com mais detalhes nas seções subsequentes.

3.1 Coleta

A coleta dos dados utilizados para a realização desse projeto foi realizada em parceria com a RedeRio de Computadores. "A RedeRio é uma rede de computadores, integrada por universidades e centros de pesquisa localizados no Estado do Rio de Janeiro"[11] fundada em 1992 no Estado do Rio de Janeiro com objetivos de impulsionar as atividades ligadas à ciência, tecnologia e educação através do provimento de infraestruturas relacionadas a área de telecomunicações. Entre suas muitas atividades, a RedeRio dá apoio à trabalhos de cunho acadêmico nas áreas relacionadas a sua atuação. Nesse ambiente de colaboração nos foi cedido acesso aos fluxos que trafegam o roteador de borda da RedeRio.

Vale a pena salientar que os fluxos consistem apenas de metadados dos pacotes que passam pelos roteadores de borda da RedeRio, ou seja, o conteúdo dos pacotes não é de forma alguma monitorado, observado ou disponibilizado e com isso mantendo preservada a privacidade dos usuários da RedeRio.

Os roteadores de borda da RedeRio são do modelo *Cisco ASR9000* [37] que possui o pacote de ferramentas de instrumentação *NetFlow* [8] que é capaz de disponibilizar o acesso aos metadados dos fluxos que trafegam por enlaces do roteador. No contexto da RedeRio, esses roteadores estão configurados para realizar a exportação de todos

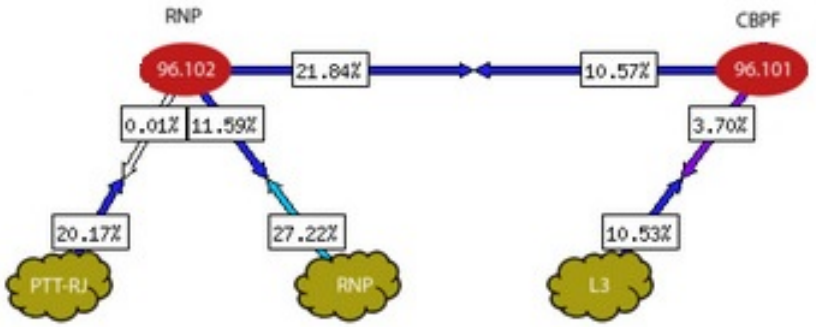

Figura 3.1: Conexões de Borda da RedeRio - Cedido pela RedeRio [11]

os fluxos que passam pelos enlaces de borda da rede, ou seja, todos os fluxos que entram e saem da rede são monitorados.

O *NetFlow* [8] é uma solução proprietária da Cisco [38] que é uma grande empresa do ramo de telecomunicações.

O *NetFlow* define um fluxo utilizando informações coletadas do cabeçalho dos pacotes, essas informações [39] são:

- *IP* da fonte.
- *IP* de destino.
- Porta da Fonte.
- Porta de Destino.
- Protocolo de rede.
- Classe de Serviço.
- Roteador ou *Switch*.

Todos os pacotes com o mesmo endereço *IP* de fonte/destino, com as mesmas portas de fonte/destino, com a mesma classe de serviço e passando pelo mesmo equipamento de rede são agrupados em um fluxo. O roteador possui uma espaço de memória reservado para a alocação dos fluxos, e quando um pacote é recebido pelo roteador ele tem seus metadados adicionados ao espaço de memória correspondente ao seu fluxo. Apesar de serem consideradas apenas estas sete características para a definição de um fluxo, os metadados capturados são todas as informações contidas nos cabeçalhos dos protocolos de camada de enlace, rede e transporte.

Devido a alta taxa de transmissão dos enlaces de borda da RedeRio, não é possível que se faça a coleta de todo o tráfego. A tentativa de realizar a coleta de todo o tráfego foi feita, porém notou-se que havia uma queda do número total de fluxos passando pela rede nos horários de pico de utilização conforme mostrado

a Figura 3.2. O motivo de tal evento se dá pois o roteador ASR9000 [40] nega recursos de processamento ao *NetFlow* quando o mesmo atinge 50% de utilização do processador da placa onde se encontra a interface monitorada. O *NetFlow* vem preparado para realizar amostragem de pacotes a serem analisados pela ferramenta. A amostragem utilizada para a realização deste trabalho é de um pacote a cada 3 pacotes.

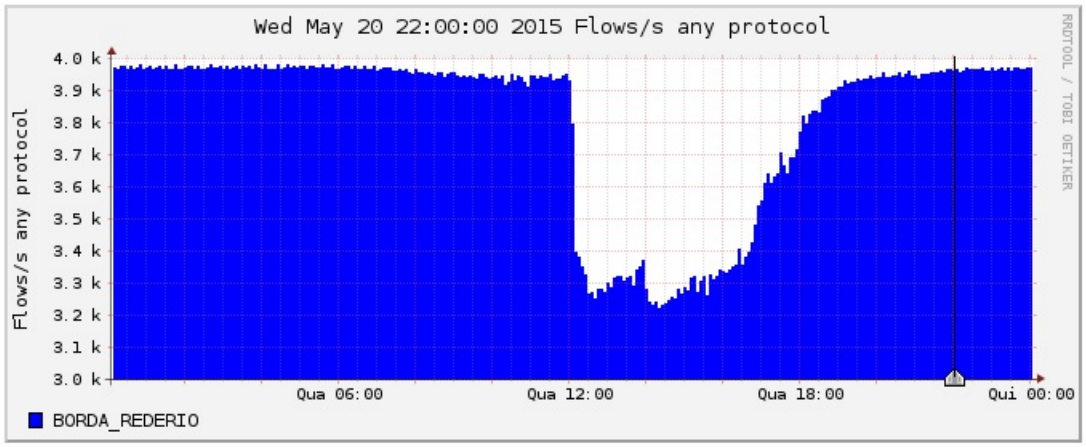

Figura 3.2: Queda da quantidade de fluxos monitorados em horário de pico

O *NetFlow* realiza a exportação dos fluxos de acordo com algumas circunstâncias, dentre elas, o fato de um determinado fluxo estar ativo ou inativo por um determinado período de tempo, quando algum recurso físico do roteador está prestes a se exaurir ou quando é explicitamente forçado pelo usuário a exportar.

O período de tempo padrão [40] pelo qual o *NetFlow* considera um fluxo como inativo e o exporta é de quinze segundos sem receber nenhum pacote. De maneira análoga, um fluxo é considerado ativo se não sofre interrupção de quinze segundos em nenhum momento em sua transmissão, e para tais fluxos o tempo padrão de exportação que o *NetFlow* utiliza é de trinta minutos. Na coleta realizada por este trabalho o tempo de exportação de fluxos ativos foi alterado para um minuto.

A exportação feita pelo roteador utiliza-se do protocolo *UDP* para transporte dos fluxos até o servidor de coleta e usualmente cada pacote *UDP* contém entre trinta e cinquenta fluxos[8]. O servidor de coleta utilizado na elaboração deste trabalho é o servidor IPTRAF localizado no laboratório RAVEL/COPPE.

O servidor IPTRAF consiste em um gabinete Dell T-620 com dois processadores Intel Xeon E5-2630 v2 de 2,6GHz tendo cada um com seis núcleos de processamento, 32GB de memória RAM do tipo DDR3, sete discos rígidos de 600GB, fonte redundante de 750W e uma placa de rede *Gigabit Ethernet*. O sistema de arquivos utiliza a tecnologia RAID 5 de paridade e recuperação de erros e possui uma partição de armazenamento exclusiva para os fluxos, pois a alta rotatividade com que os arquivos são escritos e apagados pode gerar instabilidade caso venha a ser colocado

na mesma partição que o sistema operacional. O sistema operacional utilizado é a distribuição *linux* Ubuntu *Server* 14 [41].

No servidor IPTRAF os fluxos chegam comprimidos e codificados de acordo com o padrão do *NetFlow*, se fazendo necessária a decodificação. Para a realização da decodificação foram utilizados os pacotes de ferramentas NfSen [42] e NfDump [43]. NfSen que é uma abreviação de *NetFlow Sensor* é uma ferramenta de código aberto feita para executar consultas de diversas informações relevantes e exibir de maneira gráfica os fluxos do *NetFlow*. O NfDump é responsável pelo recebimento e decodificação dos fluxos do *NetFlow* e é através dele que repassamos os fluxos para a aplicação principal do método.

Na Figura 3.3 é mostrado um exemplo de exibição construída utilizando o NfSen na máquina coletora IPTRAF, onde é mostrado todo o tráfego passando pelos roteadores de borda da RedeRio. Na Figura 3.4 é mostrado todo o tráfego passando pelos roteadores de borda, mas já considerando a direção pela qual os fluxos passam e qual a interface por onde eles são comutados.

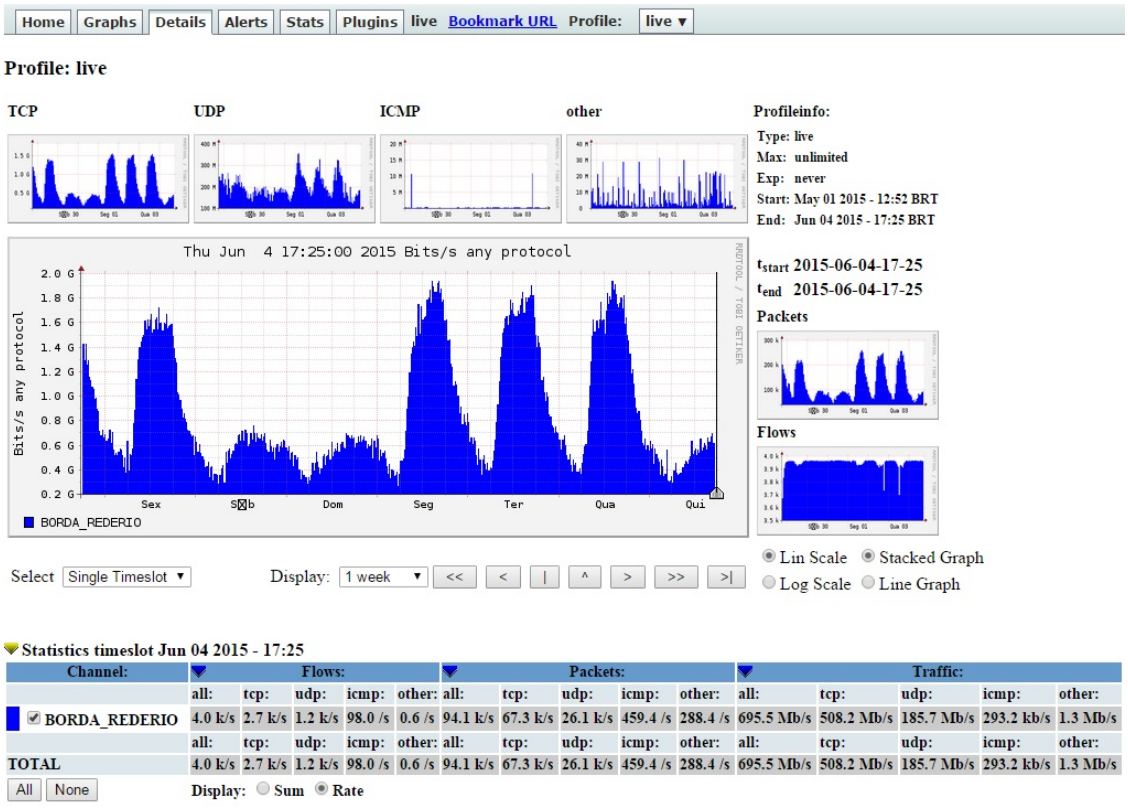

Figura 3.3: Exemplo de exibição do NfSen feita pelo IPTRAF

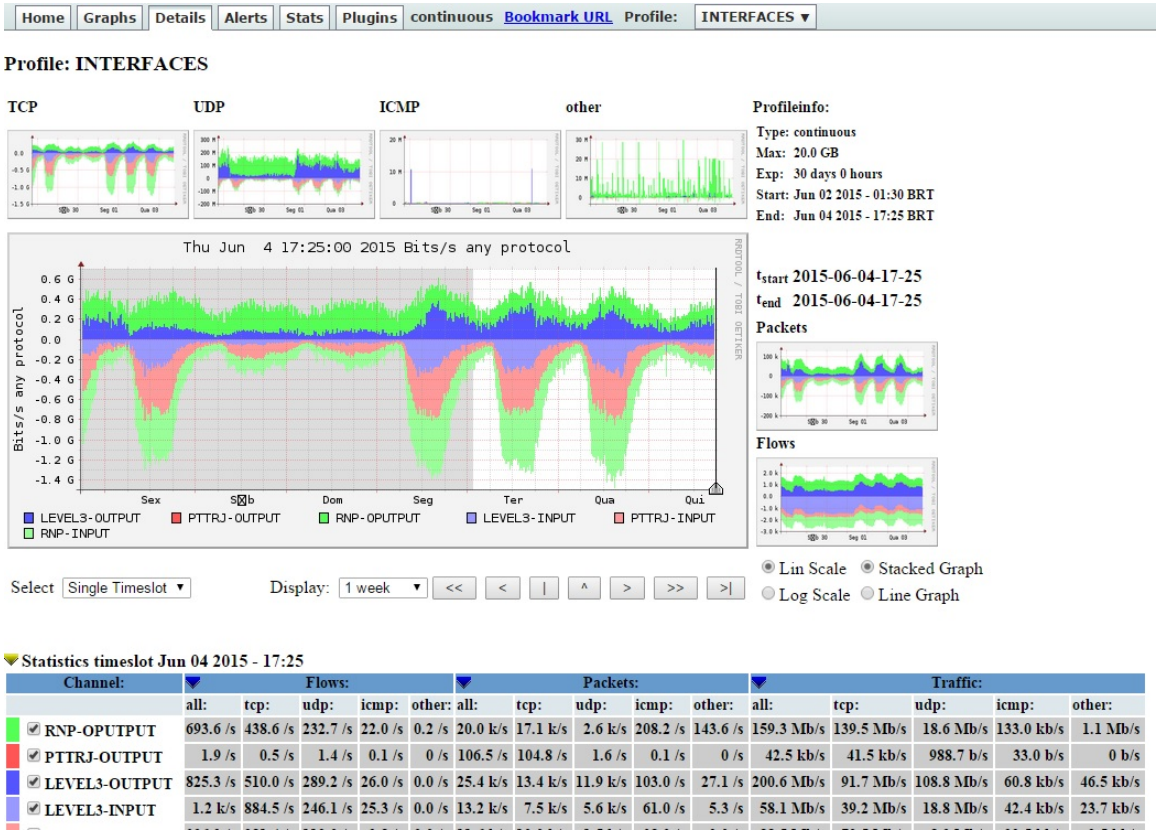

Figura 3.4: Exemplo de exibição por interface do NfSen feita pelo IPTRAF

3.2 Classificação

Logo após a etapa de coleta o método inicia sua etapa de classificação. A classificação aqui realizada tem por objetivo facilitar a detecção de anomalias através do "isolamento"dos eventos anômalos. A suposição que foi feita ao desenvolver a classificação é de que seria possível realizar uma partição do conjunto total de fluxos coletados de tal forma que uma determinada anomalia tenha todos os seus fluxos contidos dentro de apenas um dos conjuntos da partição. Sendo possível realizar tal partição haveria um ganho na capacidade de detecção de anomalias, visto que o impacto estatístico de um evento anômalo não seria tão diluído no comportamento típico de rajadas comumente observado nas redes de computadores. Sendo assim, a classificação aqui proposta tem o claro objetivo de destacar as anomalias através de seu isolamento num conjunto de fluxos menor do que o conjunto total de fluxos coletados.

Em busca da melhor forma de realizar tal classificação, foi observado o trabalho da Cláudia de Abreu Silva [14] que se tornou a principal referência utilizada na busca pela classificação ideal. O trabalho da Cláudia de Abreu Silva [14] foi comentado com mais detalhes na Seção 2.5.1 e consiste numa proposta de análise de tráfego para *Backbones* utilizando uma classificação dos fluxos, classificação esta que orientou a

proposta realizada neste capítulo.

A classificação proposta nesse método constitui uma partição do tráfego em quatro conjuntos distintos que são:

- *TCP* - Esse conjunto contém todos os fluxos cujos pacotes foram enviados utilizando o protocolo *TCP* numa conexão bem sucedida, ou seja, iniciando com um pacote de conexão (*SYN*) e finalizado por um pacote de encerramento (*FIN*) devidamente reconhecidos (*ACK*) de acordo com o protocolo. As informações referentes a esse fluxo são enviadas pelo roteador logo após o envio de um pacote de encerramento de conexão(*FYN*), ou se estiver ativo a mais de um minuto ou ainda se estiver inativo a quinze segundo consecutivos.

- *SEMI-ABERTOS* - Esse conjunto contém todos os fluxos cujos pacotes foram enviados utilizando o protocolo *TCP*, mas, que fazem parte de fluxo cujo estabelecimento da conexão foi mal sucedido ou o encerramento da conexão não foi realizado. As informações referentes a esse fluxo são enviadas pelo roteador logo após quinze segundos de inatividade.

- *NULOS* - Esse conjunto contém todos os fluxos cujos pacotes foram enviados a destinos inexistentes ou inválidos e pacotes utilizando o protocolo *TCP* cuja conexão não foi aberta.

- *UDP* - Esse conjunto contém todos os pacotes cujos fluxos não são enviadas utilizando o protocolo *TCP*, ou seja, que usam o protocolo UDP ou algum pacote de controle ligado ao *IP*. Como esses fluxos não possuem pacotes de abertura ou finalização de transmissão, os pacotes são considerados encerrados após um silêncio de quinze segundos ou um período de atividade contínua de um minuto.

Os dados são classificados fazendo uso do valor de dois indicadores enviados pelo *NetFlow*, sendo o primeiro o indicador o campo *protocol* contido no cabeçalho do protocolo *IP* capturado no fluxo. Conforme citado na Seção 2.1 este campo do cabeçalho *IP* serve para indicar o protocolo de camada de transporte utilizado pelo pacote. O segundo indicador é o campo *flags* do cabeçalho do protocolo *TCP*. Conforme citado na Seção 2.1 este campo do cabeçalho *TCP* é usado para identificar algumas características da conexão *TCP*.

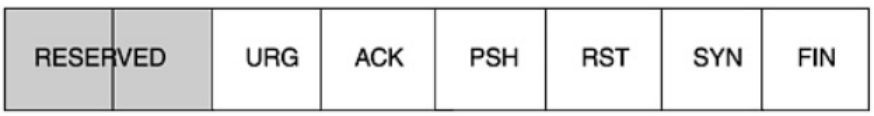

Figura 3.5: *Flags* do Protocolo *TCP* - Retirado de [23]

Para a implementação da classificação foram utilizadas as linguagens *Perl* [44] e *shell script* [45], que são linguagens que permitem a execução de tarefas implementadas diretamente pelo sistema operacional Linux. A classificação consiste em analisar os seguintes critérios:

```
SE(PROTOCOLO = TCP)ENTÃO{
    SE(FLAGS = RST + ACK)ENTÃO{
        CLASSIFICAÇÃO <- SEMI-ABERTO;
    }SENÃO{
    SE(FLAGS = RST)ENTÃO{
        CLASSIFICAÇÃO <- NULO;
    }SENÃO{
        CLASSIFICAÇÃO <- TCP;
    }}
}
SENÃO{CLASSIFICAÇÃO <- UDP}
```

De acordo com os critérios acima, todo fluxo que não utilizar o protocolo *TCP* será classificado no conjunto *UDP*. Não existem apenas dois protocolos de camada de transporte no modelo *TCP/IP*, portanto é falsa a afirmação de que um fluxo que não é *TCP* necessariamente é *UDP*, porém o *TCP* é o único protocolo largamente utilizado na *Internet* que utiliza transmissão confiável e estabelecimento de conexão. Todos os outros protocolos identificados no cabeçalho dos pacotes *IP* e que são comumente utilizados na *Internet*, como por exemplo, os protocolos de controle associados ao *IP* (*ICMP*, *IGMP*, entre outros) e protocolos de roteamento (*OSPF*, *GRE*, *RSVP*, entre outros), têm características de transmissão semelhantes ao protocolo *UDP* e portanto são classificados no mesmo conjunto da partição de tráfego.

Os fluxos classificados no conjunto *NULO* são todos os fluxos que utilizem o protocolo *TCP* e possua apenas a *flag RST* ativa. Essa classificação pode ser realizada desta forma, pois quando um determinado elemento de rede, capaz de entender o protocolo *TCP* recebe um pacote endereçado a alguma porta que não possui conexão aberta com o remetente ou qualquer outra tentativa de conexão inválida, o destinatário deve responder com um pacote *TCP* com a *flag RST* sinalizada.

O conjunto da classificação *SEMI-ABERTOS* irá receber todos os fluxos que utilizam o protocolo *TCP* e possuem as *flags RST* e *ACK*, que quando estão ativadas simultaneamente indicam que houve algum erro na conexão *TCP*, porém, era uma

conexão válida, pois o destinatário ativou a *flag ACK*, mostrando que realmente esperava-se a transmissão de alguma informação ainda nessa conexão.

O conjunto da classificação *TCP* irá conter todos os fluxos que utilizam o protocolo *TCP* e que não forem atribuídos aos conjuntos *NULO* e *SEMI-ABERTOS*.

O objetivo pelo qual são agrupados no conjunto *NULO* os fluxos inválidos é de observamos de maneira específica esses pacotes, pois pacotes inválidos são por muitas vezes utilizados para realizar escaneamento de portas e de elementos de rede conforme citado na Seção 2.3.1. Nesses tipos de escaneamento, um determinado elemento de rede malicioso envia diversos pacotes *TCP* para variados endereços *IP* e porta *TCP* e apenas espera o pacote *TCP* com a *flag RST* marcada. Com isso, o elemento de rede malicioso pode descobrir a existência de elementos de rede num determinado endereço e até mesmo inferir se um determinado elemento de rede possui um *firewall* bem configurado e os serviços prestados por aplicações nele instaladas.

O objetivo pelo qual são agrupados no conjunto *SEMI-ABERTOS* os fluxos cuja conexão não foi devidamente encerrada ou iniciada é de capturar tentativas maliciosas de reserva de recursos. Conforme citado na Seção 2.3.1 existem vários tipos de ataque, onde busca-se exaurir os recursos de um determinado elemento de rede. Tais ataques serão observados de maneira mais específica dentro do conjunto *SEMI-ABERTOS*.

3.3 Obtenção de métricas

Neste método são analisados quatro conjuntos de séries temporais, sendo os conjuntos descritos na Seção 3.2. As métricas aqui utilizadas foram inspiradas nas métricas utilizadas no trabalho [36], cada métrica tem seu valor indexado pelo instante em que é medida e é utilizada para a elaboração de uma série temporal. Cada série temporal elaborada com uma determinada métrica representa o comportamento de alguma característica da rede.

Para indexação do tempo é usada uma granularidade de cinco minutos, ou seja, todos os fluxos que chegam ao servidor de coleta no espaço de tempo de cinco minutos são indexados no mesmo instante de tempo. Esta granularidade foi escolhida por ser largamente utilizada por outros métodos propostos na literatura, incluindo os métodos citados na seção 2.5, o que facilita qualquer comparação que seja feita.

A cada cinco minutos o método observa todos os fluxos que foram recebidos nesse espaço de tempo e extrai as seguintes métricas de interesse:

- *TOTAL DE BYTES* - É a soma dos bytes transmitidos nos fluxos que são analisados num intervalo de tempo de cinco minutos.

- *TOTAL DE PACOTES* - É a soma da quantidade de pacotes de todos os fluxos que são analisados num intervalo de tempo de cinco minutos.

- *SOCKET DE DESTINO* - É o numero de fluxos observados num intervalo de tempo de cinco minutos que possuem o mesmo *socket* de destino, ou seja, é o número de pacotes que possuem um endereço *IP* de destino e porta de destino. Esta métrica é feita sob medida para a detecção de variações bruscas de demanda de conexões a um determinado destino da rede, de modo que ataques de negação de serviço possam ser evidenciados.

- *PORTAS DE DESTINO* - É o número de fluxos observados num intervalo de tempo de cinco minutos e que possuem os mesmos endereços *IP* de origem e destino, porém com diferentes portas de destino. Esta métrica é feita sob medida para a detecção de conexões a múltiplas portas de destino de um determinado elemento de rede, de modo que escaneamentos de portas possam ser evidenciados.

Para a contabilização da métrica *TOTAL DE BYTES* é levado em conta apenas o valor de quantidade de bytes transmitidos pelo fluxo. A métrica é obtida somando-se todos os valores dos fluxos coletados num período de tempo de cinco minutos. A contabilização da métrica *TOTAL DE PACOTES* é feita de maneira semelhante, porém, o valor considerado é o total de pacotes transmitidos num fluxo.

Para a contabilização da métrica *SOCKET DE DESTINO* são analisados todos os fluxos recebidos no instante de tempo de cinco minutos. Logo em seguida é construída uma lista, onde cada elemento corresponde a um fluxo, porém são colocados apenas os valores referentes ao endereço *IP* de destino e porta de destino referentes aos fluxos. Esta lista é ordenada e a cada elemento repetido encontrado nesta lista é incrementado em uma unidade o valor da métrica *SOCKET DE DESTINO*.

Para a contabilização da métrica *PORTAS DE DESTINO* são analisados todos os fluxos recebidos no instante de tempo de cinco minutos e é construída uma lista de elementos onde cada elemento corresponde a um fluxo e possui três valores, que são o endereço *IP* de destino, o endereço *IP* de origem e a porta de destino. Logo em seguida ordenamos esta lista e são removidos todos os itens repetidos. Após a remoção dos itens repetidos é construída uma nova lista, com base na lista anterior, porém contendo apenas os endereços *IP* de origem e destino. Para cada item repetido encontrado nesta nova lista o valor da métrica *PORTAS DE DESTINO* é incrementado em uma unidade.

Na Seção 2.3 são abordadas algumas técnicas de visualização de características da rede, pois os eventos que são caracterizados como anômalos dependem diretamente da percepção dos aspectos da rede e da definição do que seria o padrão de normalidade. Após a conclusão da etapa de obtenção das métricas é formada a

	TCP	UDP	SEMI-ABERTO	NULO
BYTES	TCP_BYTES	UDP_BYTES	SEMI-ABERTO_BYTES	NULO_BYTES
PACOTES	TCP_PACOTES	UDP_PACOTES	SEMI-ABERTO_PACOTES	NULO_PACOTES
SOCKETS	TCP_SOCKETS	UDP_SOCKETS	SEMI-ABERTO_SOCKETS	NULO_SOCKETS
PORTAS	TCP_PORTAS	UDP_PORTAS	SEMI-ABERTO_PORTAS	NULO_PORTAS

Figura 3.6: Métricas utilizadas pelo método proposto

visualização das características da rede utilizada nesse método e que consiste em dezesseis séries temporais, sendo quatro séries referentes às métricas aqui descritas para cada uma das quatro partições descritas na Seção 3.2.

3.4 Detecção de Anomalias

Já possuindo as séries temporais, resta realizar uma análise das séries com o objetivo de caracterizar o seu comportamento como comum ou anômalo. A análise aqui proposta não tem por objetivo a caracterização da série como um todo, mas sim dos valores em um determinado instante de tempo, em outras palavras, não se busca caracterizar a série X e sim o valor X_t em qualquer instante de tempo t.

A implementação do módulo de detecção de anomalias, referente à esta parte do método foi feita na linguagem de programação C e se encontrada instalada no servidor IPTRAF. A ferramenta completa não se encontra finalizada, pois ainda necessita de uma interface gráfica e de um módulo capaz de enviar alarmes para os gerentes de rede via email ou mensagens de texto.

A Figura 3.7 ilustra de maneira simplificada a maneira pela qual são caracterizadas as anomalias. É utilizado o método de previsão de valores em séries temporais Holt-Winters, para com base nos valores da série até o instante de tempo t, ser realizada uma estimativa do valor da série para o instante $(t+1)$. É definido também um limite superior e inferior em torno do valor estimado, e sempre que um valor medido se encontrar fora do intervalo esperado, consideramos tal valor como anômalo.

3.4.1 Modelo de Previsão de Holt-Winters

Para a elaboração do modelo de previsão, foi feita a escolha do modelo de Holt-Winters em sua forma multiplicativa, que foi descrito na Seção 2.4.3. A escolha do método de Holt-Winters se deu pela fácil implementação em ambientes computacionais e pelo fato de apresentar ótimos resultados [22], em séries temporais com uma grande variabilidade e com características de tendência e sazonalidade, características essas observadas na natureza dos dados coletados.

Para a escolha da sazonalidade utilizada no método Holt-Winters tomou-se como base o estudo de Papagiannaki [46], que conclui com base em estudos de séries

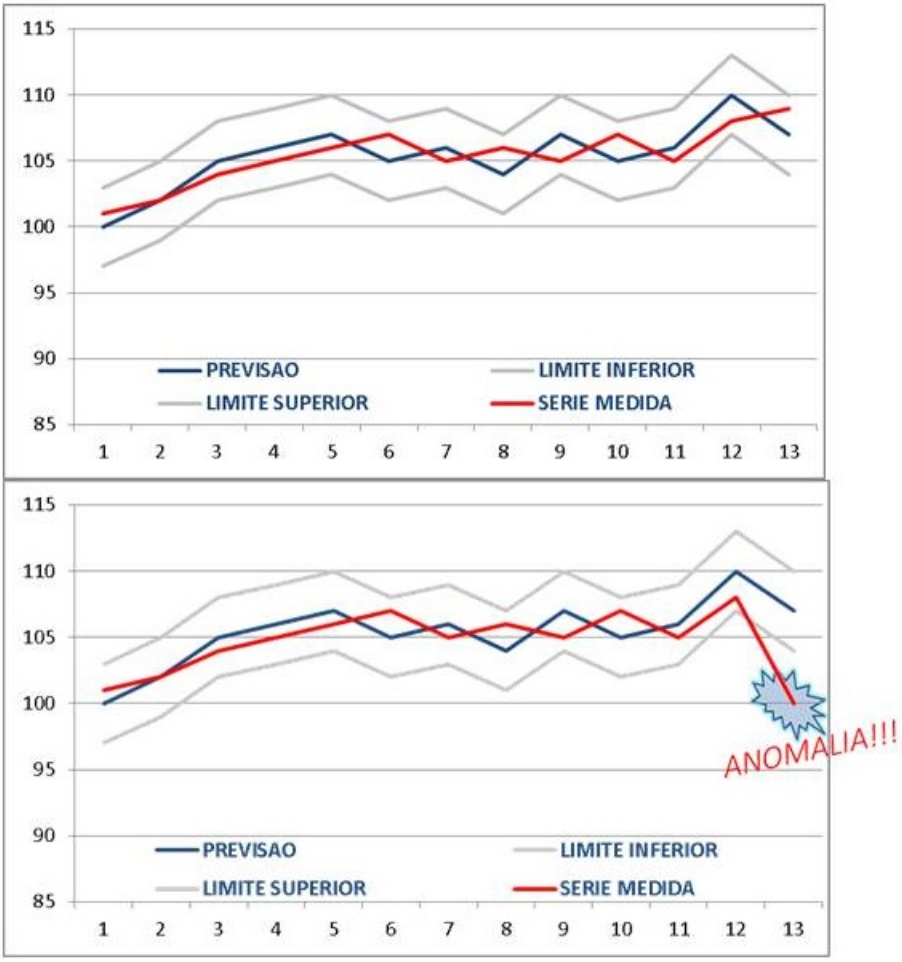

Figura 3.7: Exemplo ilustrativo da caracterização de uma anomalia

temporais coletadas em ambientes de redes de *Backbones IP*, que as características da rede apresentam fortes periodicidades considerando ciclos de doze horas, um dia e uma semana. A periodicidade considerada para a elaboração deste método é de um dia, o que corresponde a 288 unidades de tempo, dado que as medidas são realizadas em intervalos de cinco minutos.

Considerando-se as particularidades do contexto apresentado, o método de Holt-Winters mostrado na seção 2.4.3 deve ter suas constantes determinadas. Atribui-se 1 ao valor da constante h que define a quantidade de períodos discretos no futuro que a previsão será realizada. Atribui-se 288 ao valor da constante p que corresponde a uma sazonalidade de um dia.

$$\widehat{X}_t(1) = (L_t + T_t)I_{t-287} \tag{3.1}$$

$$L_t = \alpha(\frac{X_t}{I_{t-288}}) + (1-\alpha)(L_{t-1} + T_{t-1}) \tag{3.2}$$

$$T_t = \gamma(L_t - L_{t-1}) + (1-\gamma)T_{t-1} \tag{3.3}$$

$$I_t = \sigma(\frac{X_t}{L_t}) + (1-\sigma)I_{t-288} \tag{3.4}$$

Como observado na Equação 3.1 as previsões são feitas, em relação ao instante de tempo atual, apenas para um período de tempo posterior, e é o valor dessa previsão que servirá de base para a definição dos limites inferior e superior mostrados na Figura 3.7.

Os valores iniciais utilizados na implementação do modelo de Holt-Winters são:

$$\text{Para } t = 0, \text{ temos: } L_0 = X_0, T_0 = 0, I_0 = 1 \tag{3.5}$$

Para $t \in [1, 287]$, temos:

$$L_t = \alpha(X_t) + (1-\alpha)(L_{t-1} + T_{t-1}) \tag{3.6}$$

$$I_t = \sigma(\frac{X_t}{L_t}) + (1-\sigma)1 \tag{3.7}$$

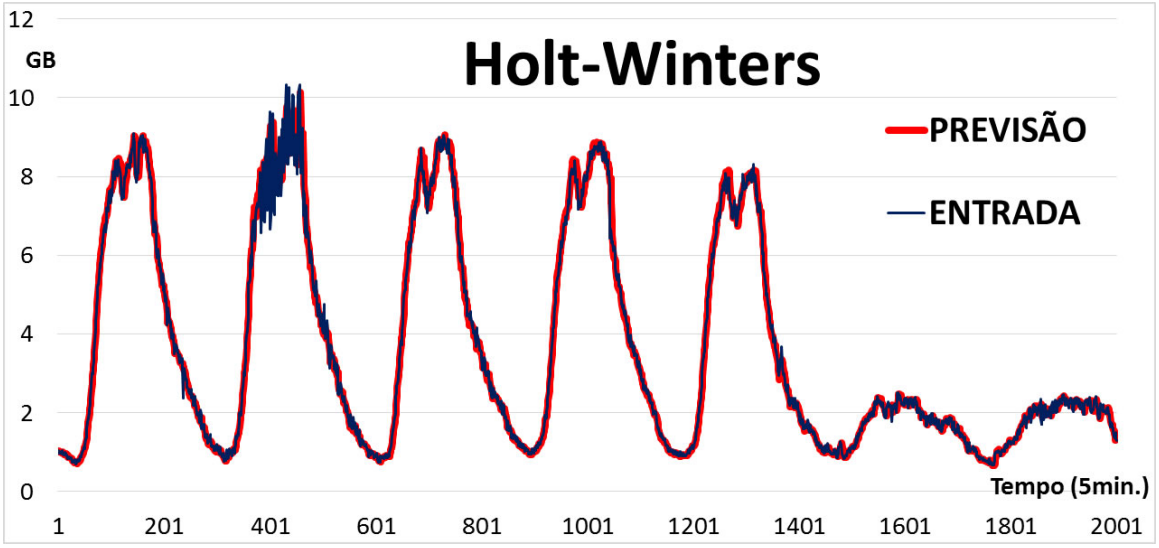

Figura 3.8: Exemplo de previsão construída com o modelo de Holt-Winters

3.4.2 Definição do Intervalo de Confiança

Os limites superior e inferior aqui utilizados são baseados nos trabalhos [36] e [34] comentados respectivamente em 2.5.3 e 2.5.2 que são duas principais referências

deste trabalho e que se baseiam em [47]. Os limites aqui utilizados são calculados tomando-se como base o desvio do valor previsto em relação ao valor medido. A fórmula utilizada para calcular o desvio consiste em:

$$d_t = \sigma \mid X_t - \widehat{X}_{t-1}(1) \mid + (1 - \sigma) d_{t-1} \tag{3.8}$$

Fazendo com que os limites possam ser obtidos da seguinte forma:

$$(\widehat{X}_t(1) - \delta d_{t-1}, \widehat{X}_t(1) + \delta d_{t-1}) \tag{3.9}$$

Onde, δ é um valor de ajuste de sensibilidade do método. Como o δ é multiplicado em ambos os lados da Equação 3.9 com maiores valores de δ obtêm-se um intervalo de confiança maior e portanto um método menos sensível [34].

O δ pode ser obtido formalmente através de um teste de hipóteses, onde a hipótese nula deste consiste em verificar se há evidências de que o valor absoluto da diferença ente o valor da série X no instante t e o valor da previsão realizada no instante $t-1$ para a série X no instante t é maior do que o desvio d_t.

$$H_0 :\mid \widehat{X}_t(1) - X_{t+1} \mid > d_t \tag{3.10}$$

Em outras palavras, H_0 busca saber através da realização desta diferença, definir se existe uma anomalia no instante t na série X.

Onde o δ é o valor limite escolhido da estatística do teste, e seus valores parciais podem ser calculados como:

$$\frac{\mid \widehat{X}_t(1) - X_{t+1} \mid}{d_t} = \delta_t \tag{3.11}$$

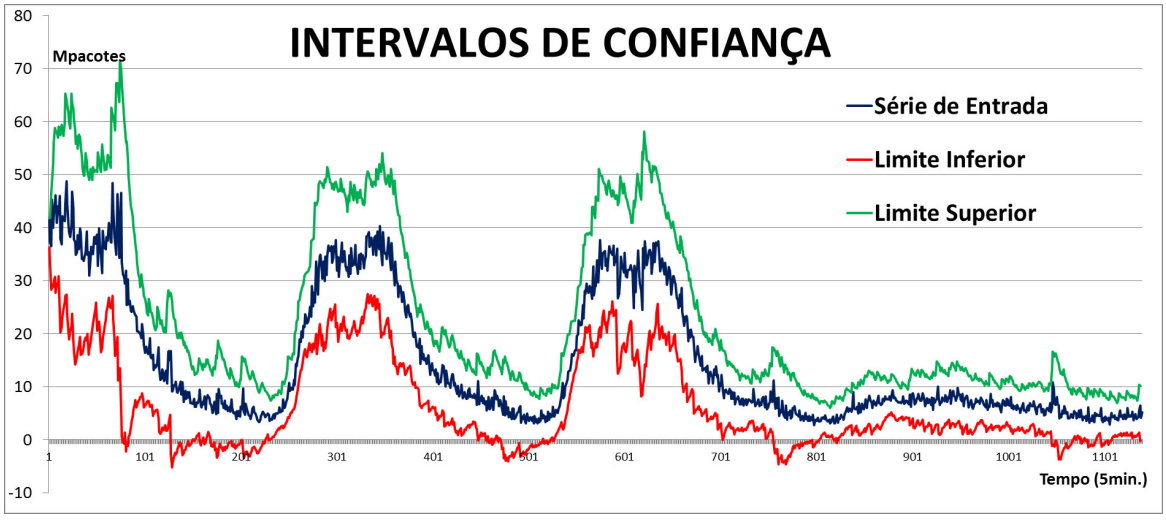

Figura 3.9: Exemplo de intervalo de confiança gerado pelo método

Os trabalhos relacionados a esse teste, assumem δ_t como normalmente distribuído e tomam valores para δ entre 2 e 3 [47] com o objetivo de estabelecer intervalos de confiança com significância entre 98% e 99, 9% de acordo com a tabela normal. Porém, realizando a análise da distribuição do δ exposta na Seção 4.2, será tomada como referência a distribuição qui-quadrado com 1 grau de liberdade, portanto para a obtenção de intervalos de 99% de confiança temos $\delta = 6, 63$. Para que se obtenha um método mais sensível basta escolher $\delta = 3, 84$ que corresponde a um intervalo de confiança de 95%, ou $\delta = 2, 71$ que corresponde a um intervalo de confiança de 90%.

3.4.3 Ajuste do Modelo de Holt-Winters

O ajuste do modelo de Holt-Winters aqui implementado consiste em avaliar o erro relativo médio (ERM) entre o valor medido e a previsão realizada, a fim de escolher as constantes de suavização onde o erro do método seja mínimo. Foi feito também um ajuste para que os valores anômalos não venham a interferir na qualidade da estimativa dos parâmetros nível médio, tendência e sazonalidade do modelo. O erro foi calculado da seguinte forma:

$$ERM = \sum_{t=0}^{n} \frac{\mid X_{t-1}(1) - X_t \mid}{L_t} \tag{3.12}$$

O valor das constantes de suavização α, γ e σ devem variar entre 0 e 1, portanto, ao realizarmos a escolha das constantes foi utilizada a seguinte metodologia:

- PASSO 1 - É construída uma lista contendo todas as combinações possíveis de valores de α, γ e σ. Com α, γ e σ variando no intervalo $[0, 1;\ 0, 2;\ 0, 3;\ 0, 4;\ 0, 5;\ 0, 6;\ 0, 7;\ 0, 8;\ 0, 9]$.
- PASSO 2 - Executa-se o método de Holt-Winters utilizando como constantes cada um dos elementos listados anteriormente e calcula-se o ERM para cada um das combinações possíveis.
- PASSO 3 - Toma-se os valores de α, γ e σ como o item da lista que obteve o menor ERM.

O método de Holt-Winters através de suas suavizações exponenciais, considera os valores da série de entrada X_t por um grande período de tempo para o cálculo do nível médio, tendência e sazonalidade. Essa característica se mostra um problema quando os valores X_t da série de entrada do método são discrepantes para algum t, o que leva o método a apresentar instabilidade conforme detalhado na Seção 4.3 e mostrado na Figura 4.7.

A solução aqui proposta para este problema consiste em sempre que detectada uma anomalia manter os valores de nível médio, tendência e sazonalidade anteriores utilizados pelo método.

$$\text{SE } X_t \text{ É ANOMALIA } \implies L_t = L_{t-1}, T_t = T_{t-1}, I_t = I_{t-288} \tag{3.13}$$

3.4.4 Classificação de anomalias

Uma anomalia identificada em uma série temporal não corresponde necessariamente à ocorrência de uma ação maliciosa na rede, portanto, ainda é necessário associar a quais eventos de rede correspondem as anomalias percebidas pelo método.

Como já comentado na Seção 2.3, uma anomalia é uma alteração no padrão de normalidade da rede, e isso pode ocorrer devido a eventos não maliciosos, como por exemplo, a transmissão em larga escala de vídeo de alta definição de um evento esportivo, abertura de inscrição de uma grande instituição acadêmica, entre outros.

A fim de diferenciar os eventos anômalos maliciosos dos não maliciosos é proposta nesse trabalho uma série de interpretações dos impactos gerados pelos eventos anômalos maliciosos em cada uma das métricas analisadas.

Logo abaixo, serão listados os ataques de rede descritos na Seção 2.3.1, juntamente com uma descrição sobre o impacto esperado nas métricas analisadas pelo do método de detecção.

- *Smurf Attack* - É esperado que sempre que ocorra um ataque deste tipo, sejam detectadas anomalias nas séries temporais correspondentes às métricas PACOTES e SOCKETS DE DESTINO da partição UDP.

- *UDP Floods* - É esperado que sempre que um ataque deste tipo ocorrer, sejam percebidas anomalias nas métricas SOCKET DE DESTINO, PACOTES e BYTES da partição UDP.

- *TCP SYN Flood* - Espera-se que este ataque seja detectado através de anomalias nas métricas SOCKETS DE DESTINO e PACOTES da partição de fluxos SEMI-ABERTOS. A detecção de anomalias na métrica PACOTES é esperada pelo fato de que o tamanho da partição de fluxos SEMI-ABERTOS é pequena, e para ataques desse tipo que sejam de alta intensidade também se espera a detecção de anomalias na métrica BYTES.

- *TCP Connect Flood* - Esse tipo de ataque possui duas variações, onde a primeira variação realiza uma conexão *TCP* completa e não a mantém aberta. Na segunda variação o atacante não só completa o processo de conexão *TCP*, como também mantém tal conexão aberta. Na primeira variação os fluxos

correspondentes serão atribuídos à partição SEMI-ABERTOS e na segunda variação serão atribuídos à partição TCP. Na ocorrência de algum ataque de negação de serviço deste tipo é esperado que o método detecte anomalias na métrica SOCKETS DE DESTINO.

- *TCP SYN Scan* - Neste escaneamento é esperado que se observe anomalias na séries temporal PORTAS DE DESTINO da partição de fluxos SEMI-ABERTOS. Devido ao tamanho da partição reduzido da partição SEMI-ABERTOS, espera-se que também sejam observadas anomalias na métrica PACOTES.

- *TCP Connect Scan* - Neste escaneamento é esperado que sejam observadas anomalias na série temporal PORTAS DE DESTINO na partição de fluxos SEMI-ABERTOS. Ao se realizar um escaneamento deste tipo, o atacante não envia pacotes que possam ser entendidos pela camada de aplicação, então após o estabelecimento da conexão ser completado o elemento de rede alvo envia um pacote com as *flags RST* e *ACK* ativas, o que faz com que os fluxos correspondentes a esse escaneamento sejam classificados na partição SEMI-ABERTOS. Ao contrário do escaneamento *TCP SYN Scan*, onde se espera detecção de anomalias na métrica PACOTES, nesse ataque não se espera, pois o atacante é obrigado a completar o processo de conexão do TCP o que reduz sua capacidade de realizar um escaneamento de alta intensidade.

- *FIN Scan* - Neste tipo de escaneamento é esperado que se observe anomalias nas métricas PORTAS DE DESTINO e PACOTES da partição de fluxos NULOS. Este ataque é detectado na partição de fluxos NULOS e como esta se trata de uma pequena partição é provável que se observem anomalias na métrica BYTES.

Os impactos esperados são inseridos na Tabela 4.4, que é utilizada pelo método para classificação dos eventos anômalos maliciosos, o que permite ao método de detecção gerar alarmes apenas na ocorrência destes eventos e apenas avisos na ocorrência de eventos anômalos não maliciosos.

Tabela 3.1: Tabela de associação de eventos anômalos maliciosos

	Smurf Attack	*UDP Floods*	*TCP SYN Flood*	*TCP Connect Flood*	*TCP SYN Scan*	*TCP Connect Scan*	*FIN Scan*
TCP - BYTES							
TCP - PACOTES							
TCP - PORTAS						X	
TCP - SOCKETS				X			
UDP - BYTES		X					
UDP - PACOTES	X	X					
UDP - PORTAS							
UDP - SOCKETS	X	X					
SEMI ABERTOS - BYTES							
SEMI ABERTOS - PACOTES			X		X		
SEMI ABERTOS - PORTAS					X		
SEMI ABERTOS - SOCKETS			X	X			
NULOS - BYTES							
NULOS - PACOTES							X
NULOS - PORTAS							X
NULOS - SOCKETS							

Capítulo 4

Resultados, Análises e Discussões

Para a elaboração deste capítulo foram realizados três períodos de coleta distintos. O primeiro período consiste na coleta de 6027 amostras do tráfego da borda da RedeRio, durante um período de aproximadamente três semanas entre o domingo dia 14 de junho de 2015 e o sábado 04 de julho de 2015.

O segundo período consiste na coleta do tráfego da borda da RedeRio entre a segunda-feira dia 20 de julho de 2015 e o sábado dia 1 de agosto de 2015, de onde foram retiradas 3605 amostras. Nesse segundo período de coleta foram induzidos ataques de rede contra a rede do laboratório Ravel, onde foi montada uma estrutura específica para receber estes ataques, a fim de realizar a validação do método de detecção.

Os ataques foram induzidos pelos alunos do laboratório Ravel, Felipe Afonso Espósito e Vander Luiz Proença da Silva. O aluno Felipe Afonso Espósito realizou os escaneamentos através de dois computadores de alto poder computacional e conectados à *Internet* com uma velocidade de 300Mbps. O aluno e autor deste trabalho Vander Luiz Proença da Silva realizou os ataques de negação de serviço através da utilização de um conjunto de dez servidores virtuais gerados através do serviço de computação elástica na nuvem provido pela *Amazon Web Services* ou AWS [48].

O terceiro período de coleta foi realizado de igual forma ao segundo, onde foram induzidos ataques contra a estrutura do laboratório Ravel, porém os ataques não foram realizados de maneira simultânea e constante. O objetivo do terceiro período de coleta é obter uma melhor visualização das anomalias, dado que no segundo período elas se misturavam e confundiam, tornando impossível a classificação das anomalias de maneira individual. O terceiro período de coleta aconteceu entre os dias 03 de agosto de 2015 e 15 de agosto de 2015.

A descrição da maneira pela qual os ataque de negação de serviço e de escaneamento de portas são realizados é apresentada com detalhes no Anexo B.

4.1 Análise da Classificação

Através da utilização do primeiro conjunto de dados coletados, foi realizada a análise do tamanho das partições obtidas através da classificação proposta na Seção 3.2.

Utilizando este conjunto de entrada, foi obtida a média das amostras de cada série temporal em todas as partições e comparadas entre si, a fim de observar a variação de métricas iguais em partições diferentes. Na Figura 4.1 pode ser notada a discrepância entre os valores médios das séries temporais entre si. As partições de fluxos *SEMI ABERTOS* e *NULOS* apresentam medidas de valores muito inferiores às medidas das partições *TCP* e *UDP*, pois consistem em conjuntos de fluxos com características muito específicas, porém essenciais para a detecção de eventos anômalos maliciosos.

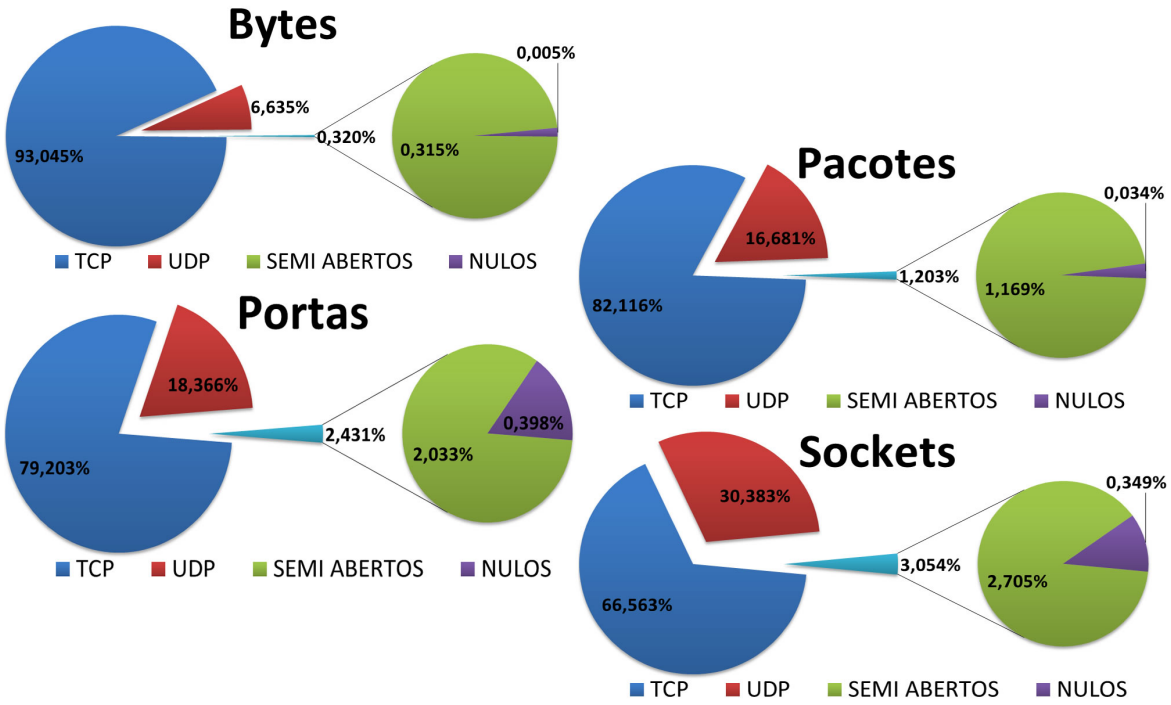

Figura 4.1: Tamanho das partições formadas pela classificação dos fluxos

A explicação para tal discrepância reside em duas características: a primeira é que os fluxos das partições *UDP* e *TCP* tendem a ter mais pacotes e consequentemente transmitir mais *bytes*, e a segunda característica é o fato de que os fluxos não se distribuem igualmente entre as partições, mas existe uma grande concentração de fluxos nos conjuntos *TCP* e *UDP*, conforme mostrado na Figura 4.2, que foi obtida através da contabilização da quantidade de fluxos em cada partição ao longo do período de coleta.

Se a técnica de detecção fosse aplicada sobre as séries temporais obtidas através da contagem das métricas em todo o conjunto de fluxos, ao invés de ser aplicada de maneira separada sobre as partições de fluxos, as anomalias que são facilmente percebidas nos conjuntos de fluxos *SEMI ABERTOS* e *NULOS* teriam muito mais

dificuldades em ser detectadas.

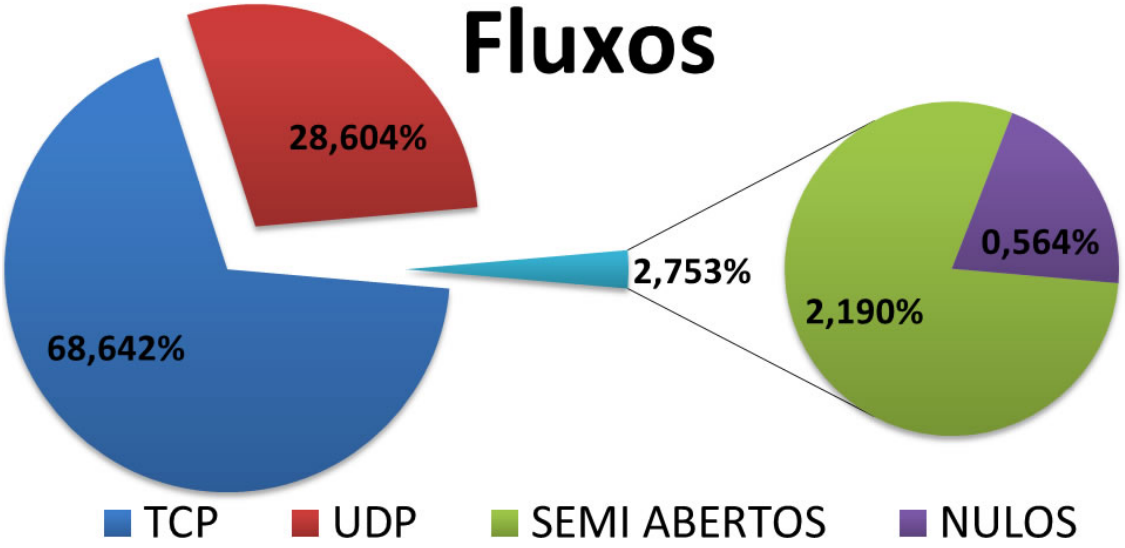

Figura 4.2: Quantidade de fluxos classificados em cada partição

Tal acontecimento pode ser percebido através das Figuras 4.3 e 4.4, onde a Figura 4.3 possui duas séries temporais, em azul a série temporal montada através da coleta da métrica *BYTES* da partição de fluxos *NULOS* e, em vermelho é mostrada a série temporal correspondente aos valores previstos. É fácil notar que no instante de tempo 225 existe um evento anômalo, pois a alteração do comportamento padrão ocorre de forma muito significativa.

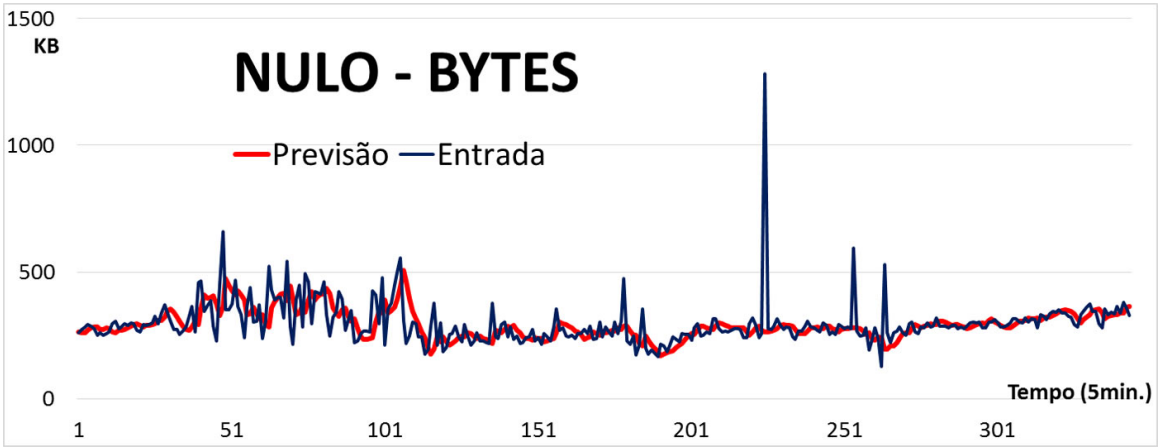

Figura 4.3: Comparação de evento anômalo em relação ao uso da classificação

Em contraste à Figura 4.3 é mostrada a Figura 4.4, onde temos em azul a série temporal referente a métrica *BYTES*, porém analisada no conjunto total dos fluxos da rede sem uso de classificação e em vermelho é mostrada a série de previsão correspondente a esta métrica.

Pode ser observado que o evento anômalo mostrado na Figura 4.3 não provoca

alteração no comportamento da série mostrada na Figura 4.4, e isso se dá pelo fato de que a anomalia em questão corresponde a uma medida de pouco menos de 1, 3MB, que em comparação ao valor da métrica no conjunto total dos fluxos é muito pequena.

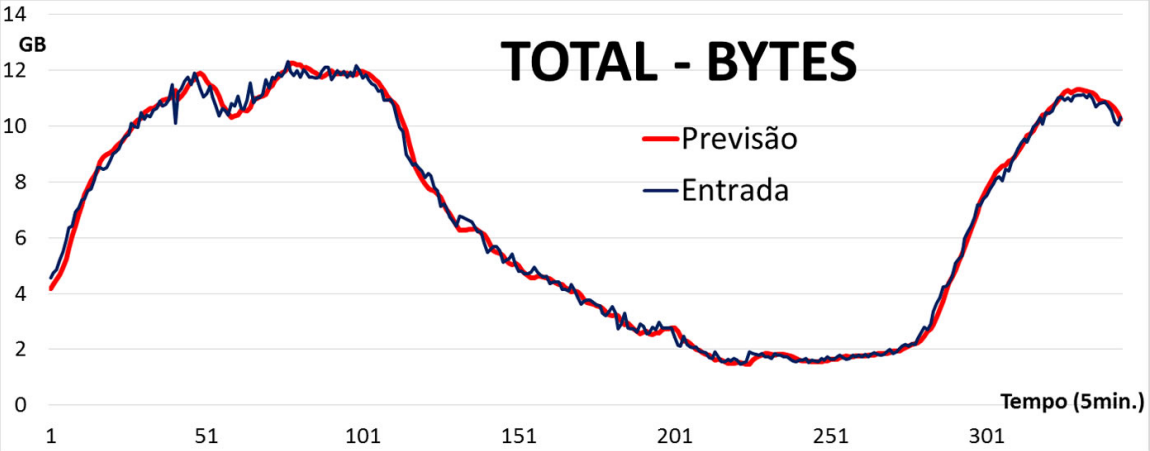

Figura 4.4: Comparação de evento anômalo em relação ao uso da classificação

4.2 Análise do Desvio

Conforme comentado na Seção 3.4.2, a bibliografia [34] referente à construção do intervalo de confiança das séries temporais, assume os valores de δ como normalmente distribuídos. A fim de confirmar tal suposição, foram calculados todos os valores parciais da estatística do teste de detecção de anomalias, conforme exposto na Equação 3.11. O cálculo dos valores δ_t foi realizado utilizando como base os valores referentes ao primeiro período de coleta dos fluxos.

Ao ser plotada a distribuição dos valores de δ_t, foi notado que tal distribuição não se aproximava da distribuição normal, mas sim de uma distribuição qui-quadrado. Na Figura 4.5 pode-se observar à esquerda a distribuição empírica de 6027 valores calculados de δ_t e à direita a distribuição de 6027 valores aleatórios gerados pelo programa R [49], utilizando a distribuição qui-quadrado com $1,07$ graus de liberdade. O R é programa de computador que possui um conjunto de ferramentas para cálculos estatísticos, geração de números aleatórios e plotagem de valores.

O parâmetro de graus de liberdade da distribuição qui-quadrado tende para a média amostral quando se observa uma grande quantidade de amostras [50]. Como a média amostral dos 6027 valores de δ_t é $1,07$, pode-se utilizar este valor como número de graus de liberdade da distribuição qui-quadrado.

O teste de Kolmogorov-Smirnov [51] apresentado com maiores detalhes no Apêndice A foi utilizado para avaliar se o primeiro período de amostras vem da

distribuição qui-quadrado. Neste teste, existe uma estatística D_n que serve para medir a diferença máxima entre os valores das amostras da distribuição empírica e a distribuição avaliada. A estatística D_n para nossos dados de entrada teve seu valor calculado e é igual a $0,009$. Considerando um nível de confiança de 99% pode-se afirmar que não existem evidências para rejeitar a hipótese de que os valores δ_t seguem uma distribuição qui-quadrado com $1,07$ graus de liberdade.

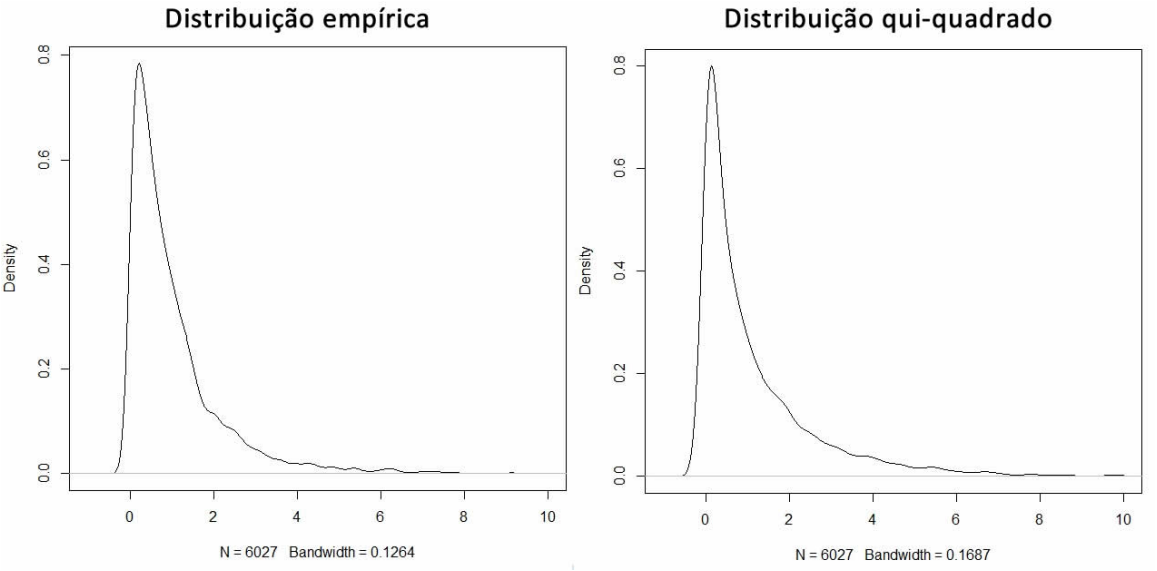

Figura 4.5: Curva de densidade empírica dos valores de δ_t e curva de densidade da distribuição qui-quadrado

Dada a natureza dos dados coletados, pode-se afirmar que amostras pequenas com grande quantidade de anomalias tendem a rejeitar a hipótese de que se ajustam à uma distribuição qui-quadrado segundo o teste de Kolmogorov-Smirnov. Porém, a distribuição qui-quadrado foi escolhida por ser a que melhor se ajusta aos dados observados.

4.3 Ajuste do modelo Holt-Winters

Ao observar a qualidade do ajuste da previsão construída através do modelo de Holt-Winters aos dados de entrada, foi notado que o modelo não apresentava bons resultados quando exposto a valores anômalos muito discrepantes nos dados de entrada. Como mostrado na Figura 4.6, o modelo entende que tal valor discrepante se encaixa nas características sazonais do comportamento da série.

A proposta implementada neste trabalho para correção do modelo de previsão, consiste em não utilizar os valores anômalos para a elaboração da previsão. Sempre que é identificada uma anomalia, as previsões passam a desconsiderar os valores de nível médio, tendência e sazonalidade calculados com base no valor anômalo e

em seu lugar considerar os valores correspondentes a estes observados no instante de tempo anterior. O resultado obtido por tal abordagem é mostrado na Figura 4.7, onde pode-se notar que a anomalia detectada não é absorvida pelo método de previsão como valor esperado para os próximos períodos sazonais.

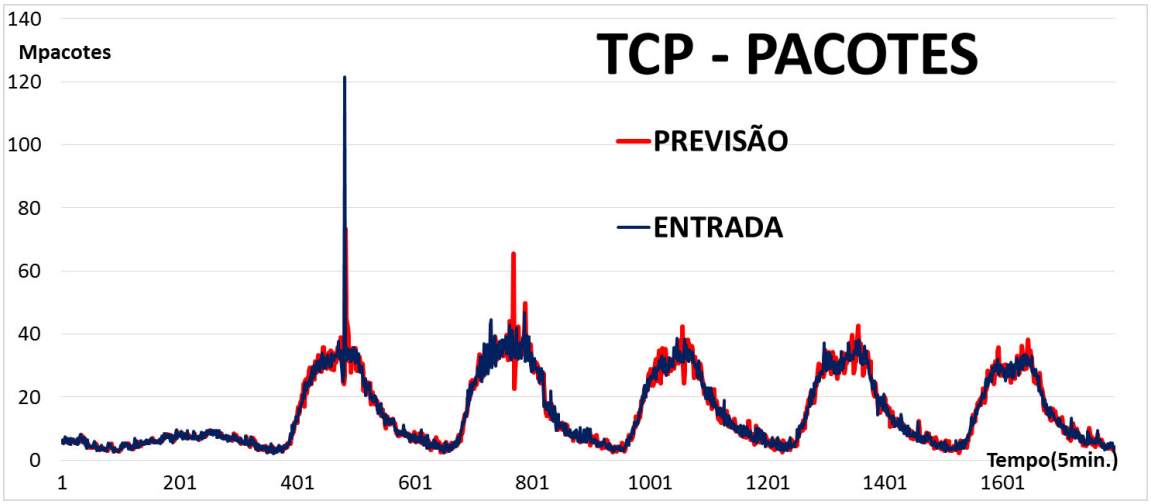

Figura 4.6: Previsão utilizando o modelo Holt-Winters considerando valores anômalos

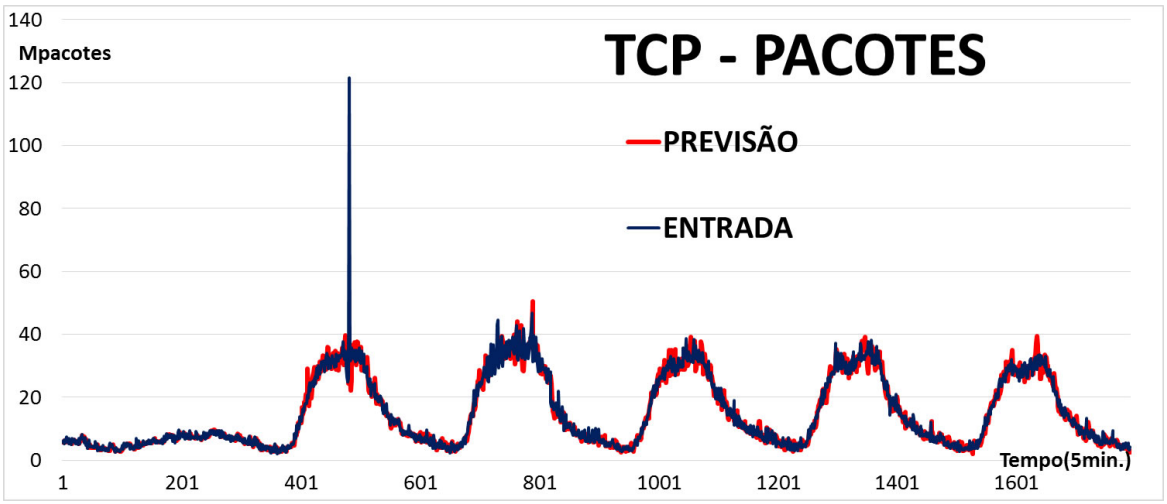

Figura 4.7: Previsão utilizando o modelo Holt-Winters sem considerar valores anômalos

Como mostrado na Seção 3.4.3, a implementação da previsão pelo método de Holt-Winters aqui proposta busca escolher as constantes de amortecimento de modo que o *ERM* seja minimizado. Para todos as amostras utilizadas nas análises, os valores de γ e σ escolhidos pelo método foi de $0,1$. Esse fato nos mostra que o valor ideal para essas duas constantes se encontra entre 0 e $0,1$, pois as métricas coletadas mostram uma grande variabilidade e o método de Holt-Winters constrói previsões com erros menores quando a tendência e sazonalidade são mais estáveis.

A Figura 4.8 mostra duas curvas, em azul a série $X(t)$ contendo os valores da métrica PORTAS DE DESTINO da partição *TCP* e em vermelho sua previsão $\widehat{X}_{t-1}(1)$. O valor de α utilizado para elaboração desta previsão foi $0,9$ que foi o valor escolhido pela metodologia proposta por alcançar o menor *ERM* entre os valores testados, que foi de $10,7\%$.

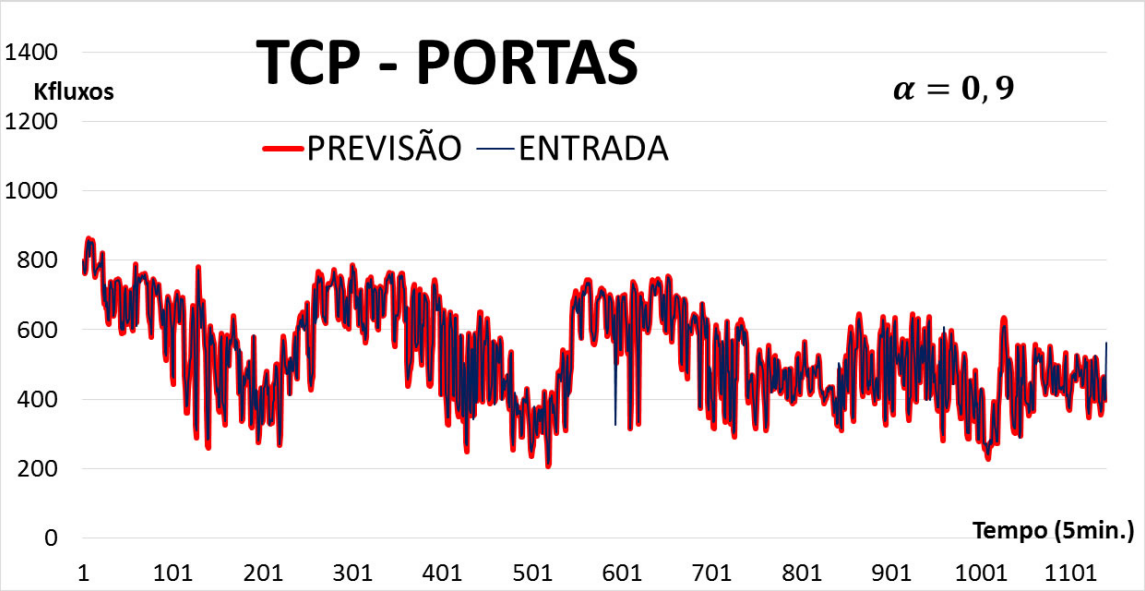

Figura 4.8: Previsão utilizando o modelo Holt-Winters com constantes de amortecimento com valores altos

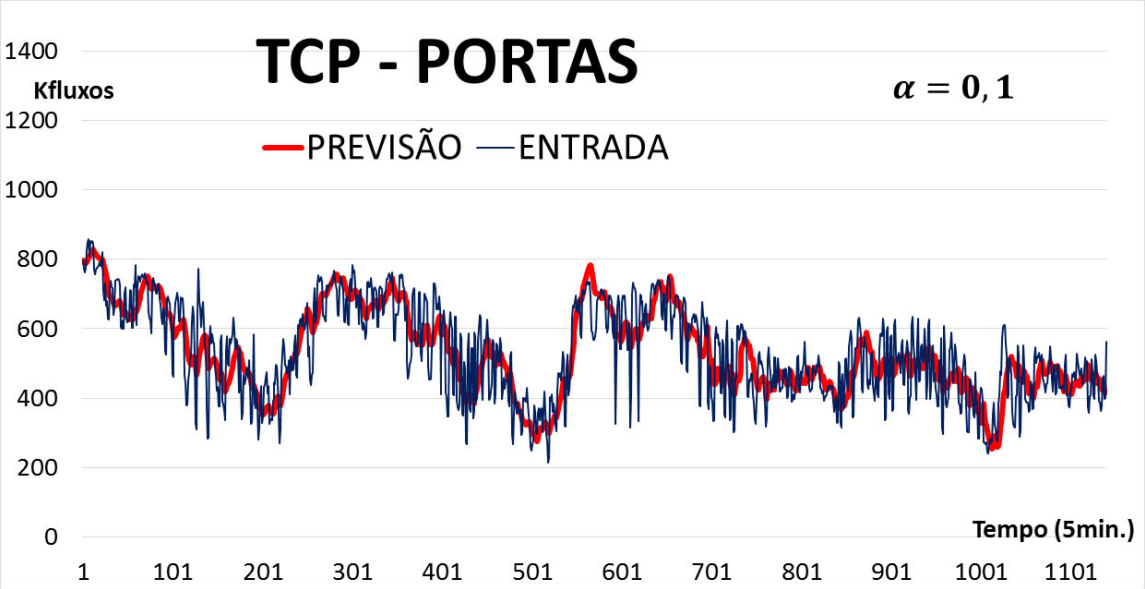

Figura 4.9: Previsão utilizando o modelo Holt-Winters com constantes de amortecimento com valores baixos

Na Figura 4.9 são mostradas duas curvas semelhantes as da Figura 4.8, porém a previsão foi construída com o valor de α igual a $0,1$ mantendo assim o nível médio

do modelo de Holt-Winters com um comportamento mais estável. Apesar do *ERM* obtido com a utilização de $\alpha = 0,1$ ser de $14,01\%$ e portanto maior do que o *ERM* obtido com $\alpha = 0,9$ o comportamento padrão da métrica é mais bem aproximado. A detecção de anomalias realizada considerando este outro padrão de comportamento difere do primeiro, de modo que quando utilizada a previsão mostrada na Figura 4.8 é feita a detecção de três anomalias e considerando-se a previsão mostrada na Figura 4.9 não é realizada a detecção de nenhuma anomalia.

Cabe a um trabalho futuro propôr uma melhor maneira de estimar as constantes de amortecimento de modo que o padrão de comportamento seja mais bem aproximado bem como avaliar o impacto na detecção de anomalias.

4.4 Correlações entre as métricas do método

A fim de analisar o comportamento das métricas entre si, foram medidas as correlações ente cada par de métricas dentro de cada partição.

A correlação entre os pares de métricas foram calculadas utilizando-se do coeficiente de correlação de Pearson. O coeficiente de correlação entre duas variáveis aleatórias X e Y é dado por:

$$\rho_{X,Y} = \frac{E(XY) - E(X)E(Y)}{\sigma_X \sigma_Y} \tag{4.1}$$

Onde, $E(X)$ é o valor esperado da variável aleatória X e σ_X é o seu desvio padrão.

Para o cálculo das correlações, utilizamos os dados referentes ao primeiro período de coleta, que não possui anomalias induzidas.

Através da aplicação da Equação 4.1 em cada par de métricas de cada uma das partições separadamente nos deparamos com as seguintes matrizes de correlação.

Tabela 4.1: Matriz de correlação das métricas da partição *TCP*

	BYTES	PORTAS DE DESTINO	SOCKETS DE DESTINO	PACOTES
BYTES	1,00	0,60	0,68	0,98
PORTAS DE DESTINO	0,60	1,00	0,98	0,53
SOCKETS DE DESTINO	0,68	0,98	1,00	0,61
PACOTES	0,98	0,53	0,61	1,00

Tabela 4.2: Matriz de correlação das métricas da partição *UDP*

	BYTES	PORTAS DE DESTINO	SOCKETS DE DESTINO	PACOTES
BYTES	1,00	-0,30	-0,28	0,72
PORTAS DE DESTINO	-0,30	1,00	0,96	-0,28
SOCKETS DE DESTINO	-0,28	0,96	1,00	-0,22
PACOTES	0,72	-0,28	-0,22	1,00

Tabela 4.3: Matriz de correlação das métricas da partição *SEMI-ABERTOS*

	BYTES	PORTAS DE DESTINO	SOCKETS DE DESTINO	PACOTES
BYTES	1,00	-0,08	0,46	0,65
PORTAS DE DESTINO	-0,08	1,00	0,79	-0,01
SOCKETS DE DESTINO	0,46	0,79	1,00	0,33
PACOTES	0,65	-0,01	0,33	1,00

Tabela 4.4: Matriz de correlação das métricas da partição *NULOS*

	BYTES	PORTAS DE DESTINO	SOCKETS DE DESTINO	PACOTES
BYTES	1,00	0,55	0,60	0,51
PORTAS DE DESTINO	0,55	1,00	0,34	0,27
SOCKETS DE DESTINO	0,60	0,34	1,00	0,29
PACOTES	0,51	0,27	0,29	1,00

Na partição de fluxos *TCP*, podemos observar que as métricas possuem uma correlação positiva entre si e com valores sempre acima de 0,5, o que nos mostra que todas as métricas tendem a crescer ou decrescer juntas. O acontecimento de tal fenômeno se dá pelo fato de alcançarem seus maiores valores sempre no horário de pico de utilização da rede. Na partição *TCP* destaca-se também a alta correlação ente as métricas *PACOTES* e *BYTES*, bem como *SOCKETS DE DESTINO* e *PORTAS DE DESTINO*.

Nas partições *UDP* e *SEMI-ABERTOS*, as únicas correlações com valores relevantes são as correlações entre as métricas *PACOTES* e *BYTES* e também entre as métricas *SOCKETS de DESTINO* e *PORTAS DE DESTINO*.

Na partição *NULOS*, pode-se notar uma correlação moderada entre a métrica *BYTES* e todas as outras métricas.

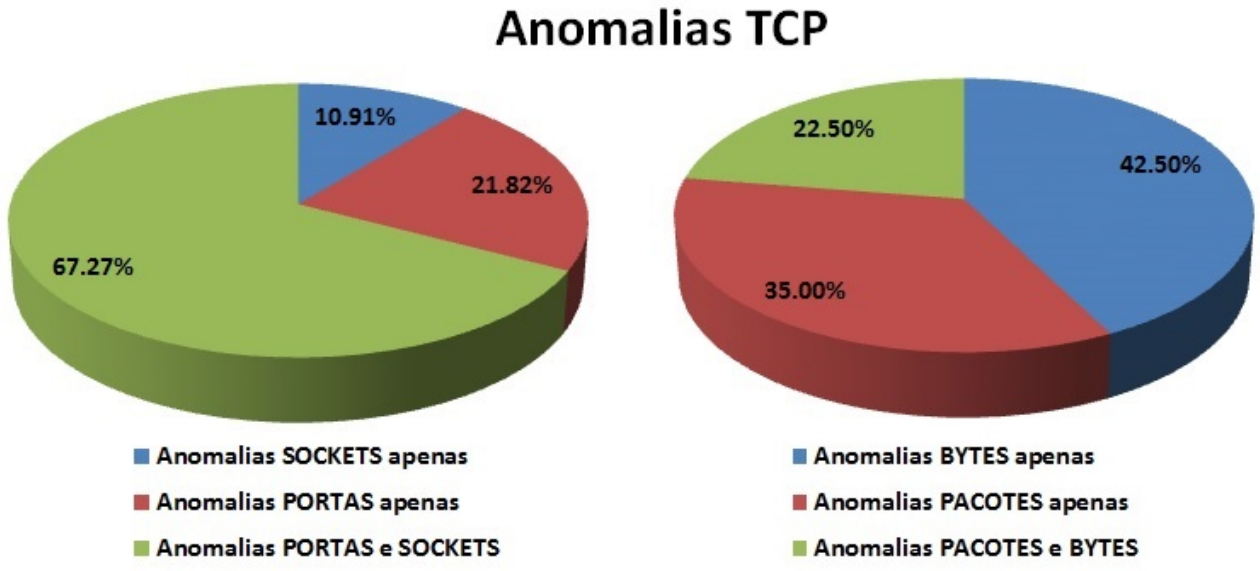

Figura 4.10: Comparativo das anomalias na partição *TCP*

Apesar do alto índice de correlação observado entre alguns pares de métricas das partições *TCP* e *UDP*, não se pode dizer que é possível que a análise de apenas uma das métricas do par seja suficiente para uma adequada detecção de anomalias.

Nas Figuras 4.10 e 4.11, podem ser observados quatro gráficos onde, na cor verde é mostrada a porcentagem de eventos anômalos simultâneos num determinado par de métricas com alta correlação.

O único caso onde a correlação dos dados se refletia de alguma forma em correlação de anomalias se dá entre as métricas *SOCKETS DE DESTINO* e *PORTAS DE DESTINO* da partição *TCP*, onde cerca de 67% das anomalias aconteceram simultaneamente em ambas as métricas. Em todos os outros casos, a ocorrência simultânea de anomalias se deu em menos de 23% dos casos.

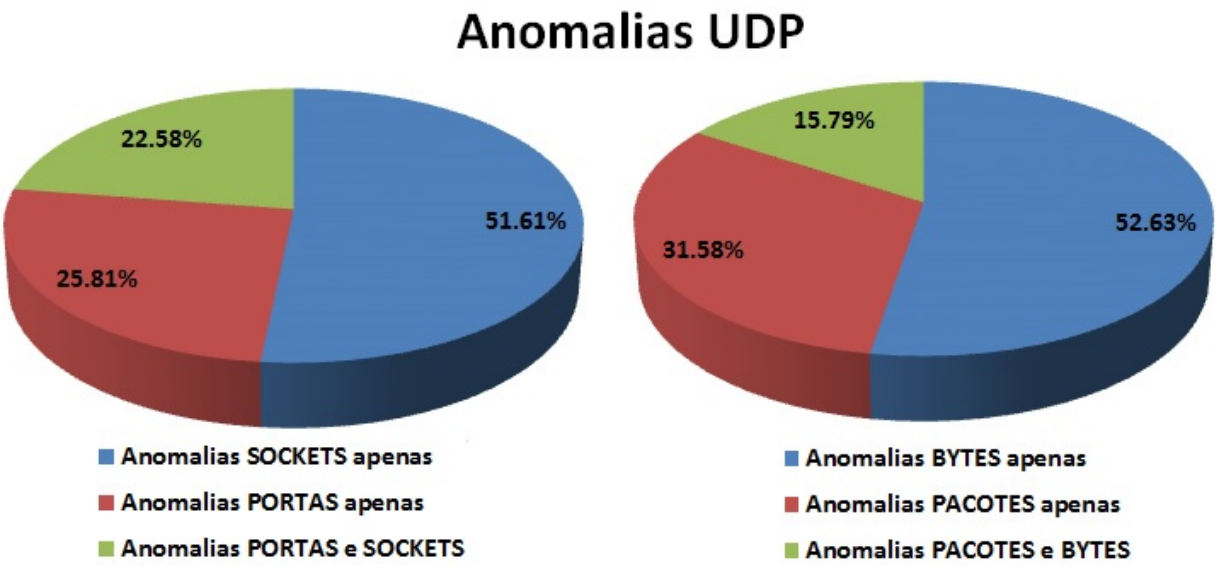

Figura 4.11: Comparativo das anomalias na partição *UDP*

4.5 Análise da Detecção de Anomalia

Nesta seção serão avaliados os ataques e escaneamentos gerados contra a rede do laboratório Ravel a fim de validar o método. Nas primeiras subseções serão abordados alguns dos escaneamentos e ataques mais comuns de maneira individual. Na última subseção será apresentada uma comparação do método de detecção proposto neste trabalho e do método proposto pela dissertação de mestrado de José Barbosa [52], aluno do programa de engenharia de sistemas e computação da COPPE e ligado ao laboratório Ravel.

4.5.1 Escaneamento *FIN Scan*

O primeiro teste realizado consiste em um escaneamento de portas do tipo *FIN Scan* em toda a rede do laboratório RAVEL. O *FIN Scan* é descrito com mais detalhes na Seção 2.3.1. Os dados analisados nesta seção foram obtidos no segundo período de coleta.

Após a elaboração do ataque o método de detecção foi executado e com os dados vindos do método foi construída a Figura 4.12, onde é mostrado o comportamento das séries temporais da partição de fluxos NULOS, onde a anomalia foi detectada. As séries temporais das partições *SEMI-ABERTAS* e *UDP* não tiveram alteração de comportamento relevante e por isso não são exibidas. A Figura 4.12 possui no eixo vertical o valor da métrica indicada no gráfico e no eixo horizontal o número referente ao instante de tempo onde foi feita a medida.

A execução do método nos aponta a detecção de quatro intervalos de tempo contendo anomalias. O primeiro intervalo com presença de anomalias consiste em

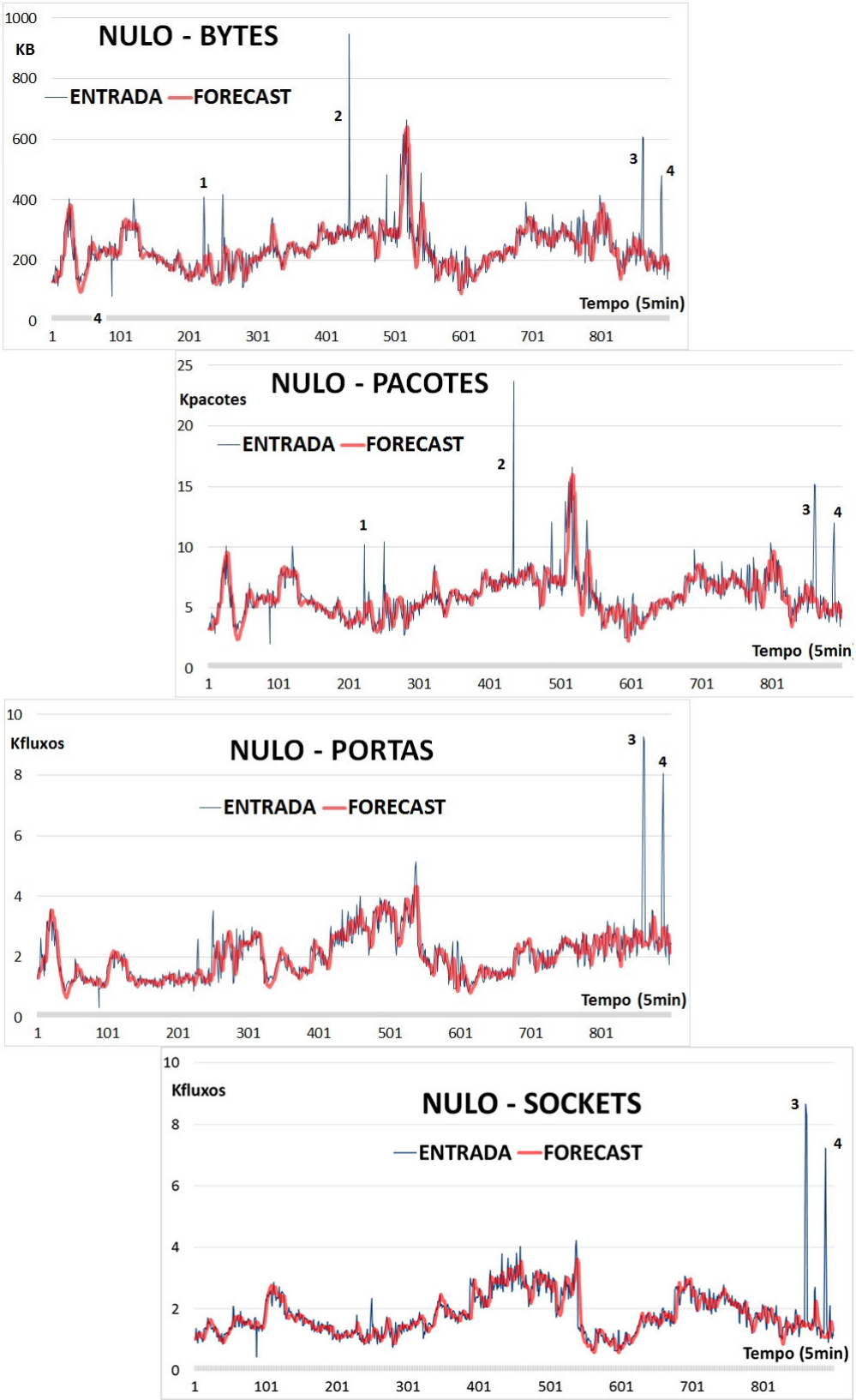

Figura 4.12: Análise de evento de escaneamento de portas com a técnica *FIN Scan*

apenas um período de tempo que é o instante 223, onde nota-se anomalias nas séries temporais referentes às medidas de *BYTES* e de *PACOTES*. O segundo intervalo

com presença de anomalias consiste nos períodos de tempo 434 e 435, onde também se pode notar anomalias apenas nas séries temporais referentes às medidas de *BYTES* e de *PACOTES*. O comportamento observado nesses dois primeiros intervalos de tempo, não corresponde a nenhum dos padrões de eventos anômalos maliciosos descritos na Seção 3.4.4.

O terceiro intervalo de tempo com presença de anomalias consiste em dois períodos de tempo que são os instantes 861 e 862. O quarto intervalo de tempo consiste nos intervalos de tempo 889 e 890. Em ambos os intervalos de tempo, o método detectou anomalias em todas as séries temporais da partição de fluxos *NULOS*.

De acordo com a Tabela 4.4 de eventos maliciosos mostrada na Seção 3.4.4, a detecção de anomalias nas partição de fluxos *NULOS* nas métricas *PACOTES* e *PORTAS DE DESTINO*, configura um escaneamento de portas do tipo *FIN Scan*. Portanto, o método realizou com sucesso a detecção do ataque. Nesse tipo de ataque não se esperava que as métricas *BYTES* e *SOCKETS DE DESTINO* fossem alarmadas, porém, devido ao fato de o tamanho da partição ser muito pequena a variação gerada pelo ataque foi suficiente para a ocorrência de uma anomalia.

4.5.2 Escaneamento *TCP SYN Scan*

A Figura 4.13, foi construída com dados retirados do terceiro período de coleta, onde foi induzido um escaneamento de portas do tipo *TCP SYN Scan* que é abordado com detalhes na Seção 2.3.1.

Quando ocorre um escaneamento deste tipo na rede é esperado que o método aqui proposto seja capaz de detectar anomalias nas métricas, *PORTAS* e *PACOTES* da partição de fluxos *SEMI-ABERTOS*.

Porém, ao analisar-se os resultados obtidos, percebe-se que não houve detecção de anomalias no comportamento da métrica *PACOTES* da partição de fluxos *SEMI-ABERTOS*. Isso se deve ao fato de que a quantidade de pacotes emitidos pelo ataque não foi suficiente para ativar a métrica PACOTES, que teve seu tamanho subestimado ao ser construída a Tabela 4.4 de associação de eventos anômalos maliciosos, que deve ser corrigida numa futura implementação.

A métrica *PORTAS DE DESTINO* foi alarmada com sucesso durante a ocorrência da anomalia, portanto, o escaneamento foi detectado. Como crítica ao método, tem-se o fato de que quando ocorre uma anomalia por um grande período de tempo, o método começa a entender a anomalia como padrão de funcionamento comum da rede. O escaneamento durou cerca de três horas e cinco minutos, porém, só foi alarmado durante seus vinte primeiros minutos e seus cinco últimos minutos. Isso se deve ao fato de que durante o escaneamento, os valores da métrica analisada

possuem uma grande variação e portanto aumentam muito o valor do desvio d_t tornando a detecção mais permissiva. Esta característica pode ser observada na figura 4.21 e comentada com maiores detalhes na seção 4.5.6.

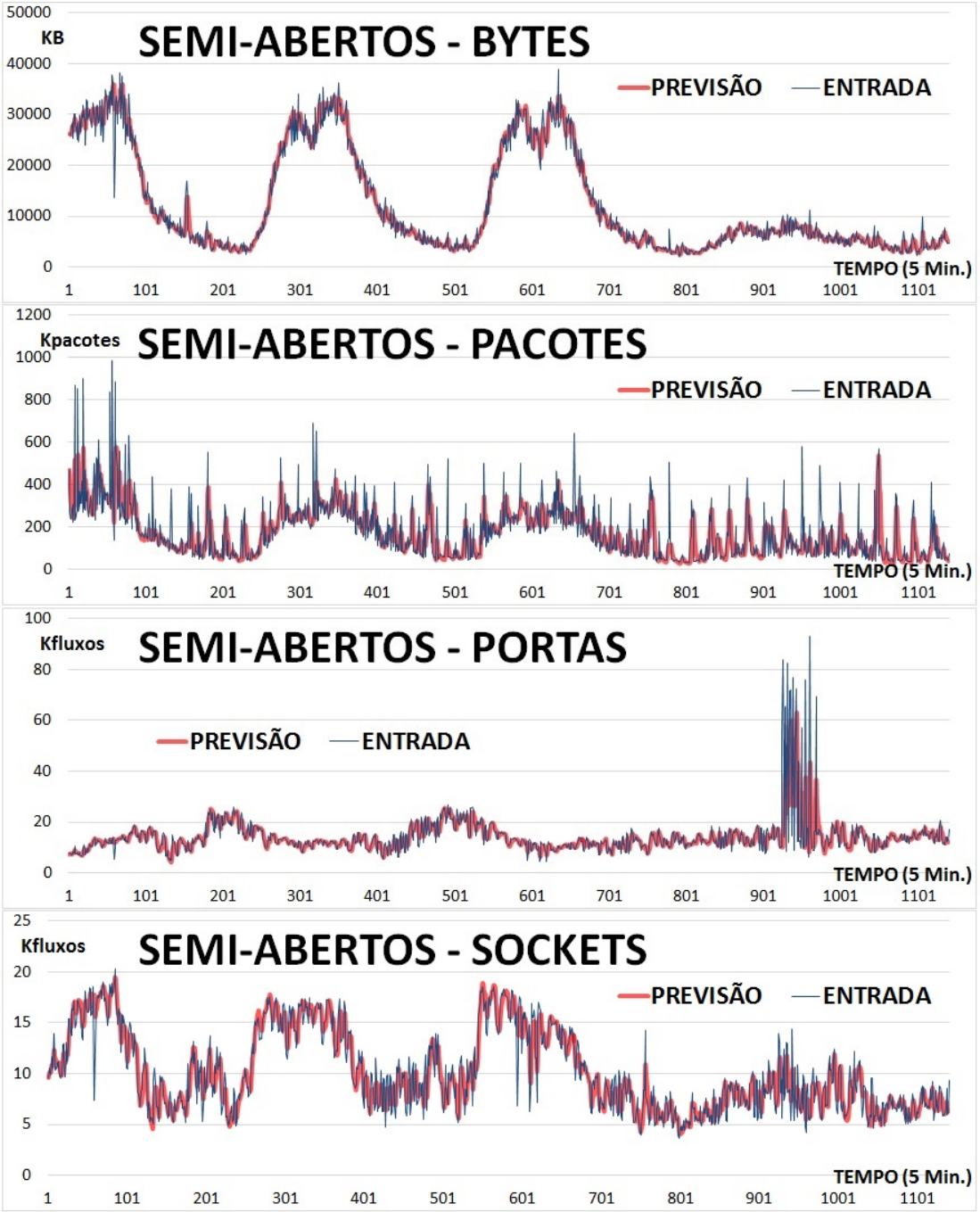

Figura 4.13: *TCP SYN Scan*

4.5.3 Ataque de negação de serviço *TCP SYN Flood*

Neste tipo de ataque, era esperado que fossem observadas anomalias nas métricas *SOCKET DE DESTINO* e *PACOTES* da partição de fluxos *SEMI-ABERTOS*, e

em caso de ataques de alta intensidade que também fossem detectadas anomalias na métrica *BYTES*.

Na Figura 4.14 pode ser percebida a ocorrência de anomalias nas métricas *SOCKETS DE DESTINO*, *PACOTES* e *BYTES* da partição de fluxos *SEMI-ABERTOS*, anomalias essas detectadas pelo método a partir do intervalo de tempo 522.

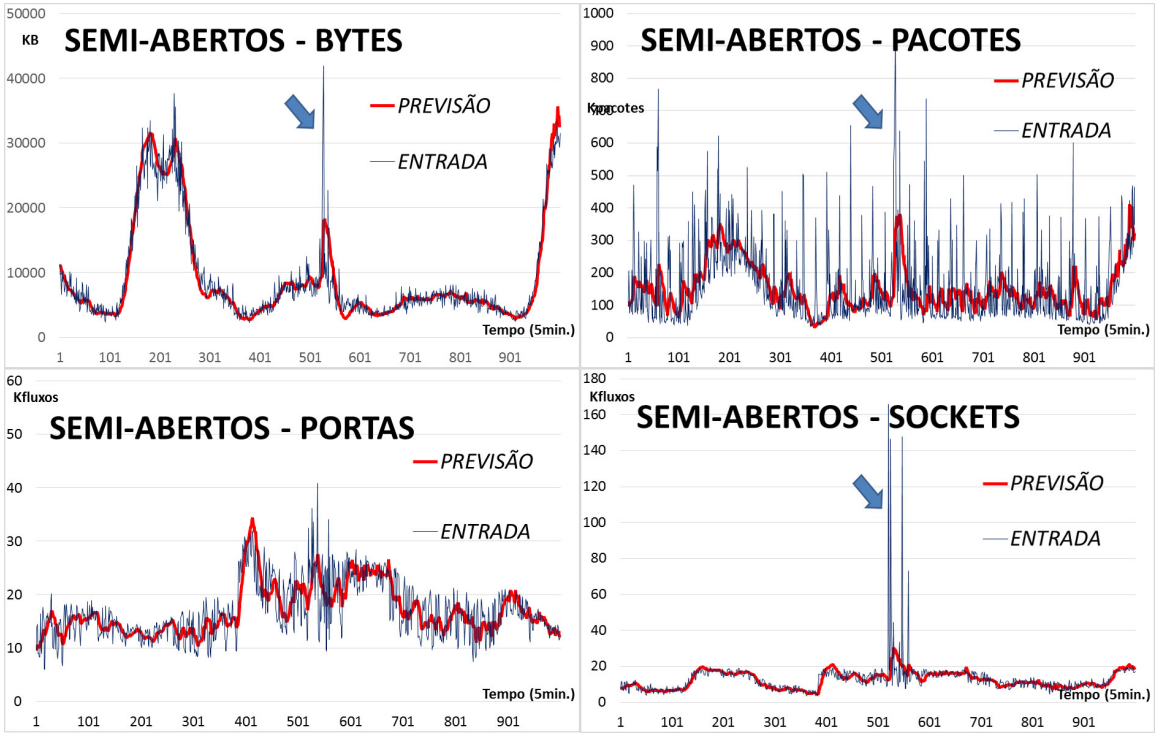

Figura 4.14: *TCP SYN Flood* - Partição de fluxos SEMI-ABERTOS

Na Figura 4.15, podem ser percebidas anomalias na métricas *SOCKETS DE DESTINO* e *PACOTES* da partição *TCP*, onde não eram esperadas anomalias. A explicação para tal ocorrência se deve ao fato de que, para que um fluxo seja classificado na partição *SEMI-ABERTOS* ele necessita de um pacote de resposta marcado com as *flags SYN* e *ACK*, que são enviadas pelo elemento de rede que está sendo atacado. Porém, no ataque de teste realizado, o elemento de rede se tornou completamente indisponível após cerca de uma hora de ataque e os pacotes relacionados ao ataque foram classificados na partição *TCP*. Na partição *TCP*, as anomalias na métrica *PACOTES* foram detectadas, mas anomalias na métrica *SOCKETS DE DESTINO* não foram detectadas devido à alta variabilidade dos valores da métrica.

4.5.4 Ataque de negação de serviço *UDP Flood*

Na Figura 4.16 pode ser observado o comportamento das métricas do método durante a ocorrência de um ataque de negação de serviço *UDP Flood*. Esta Figura foi construída com dados do terceiro período de coleta, e o ataque foi gerado utilizando-

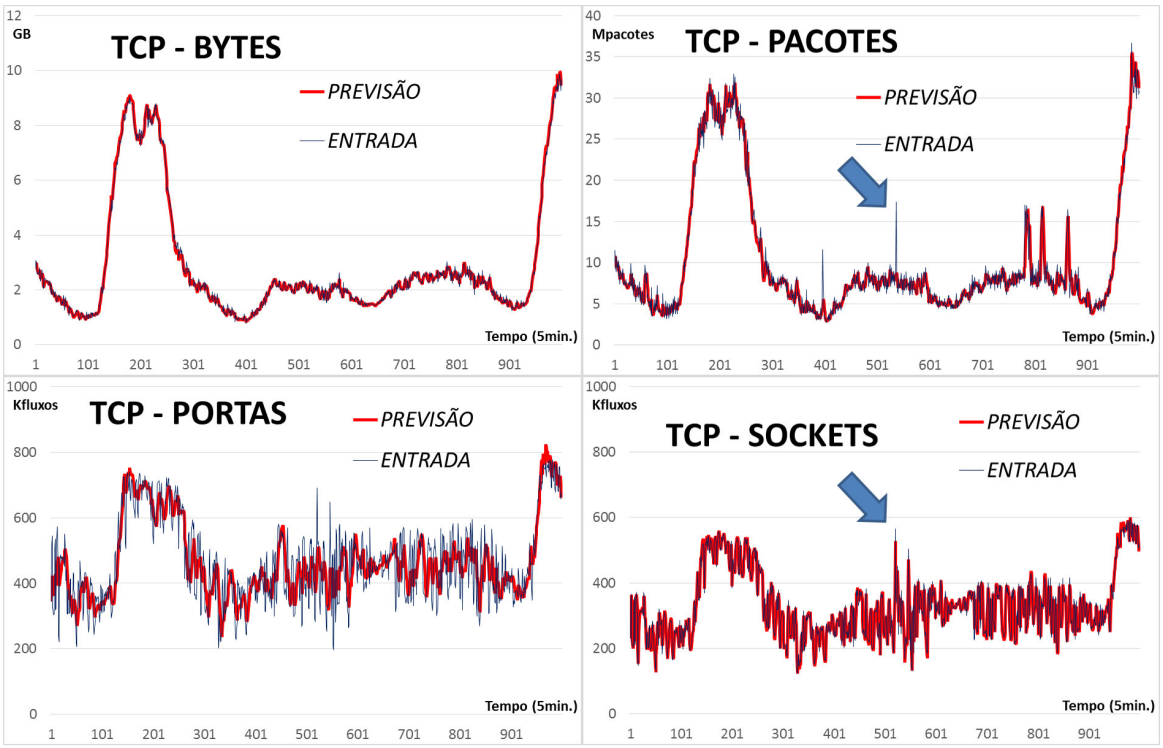

Figura 4.15: *TCP SYN Flood* - Partição de fluxos TCP

se a estrutura dos servidores virtuais gerados pelo *AWS*.

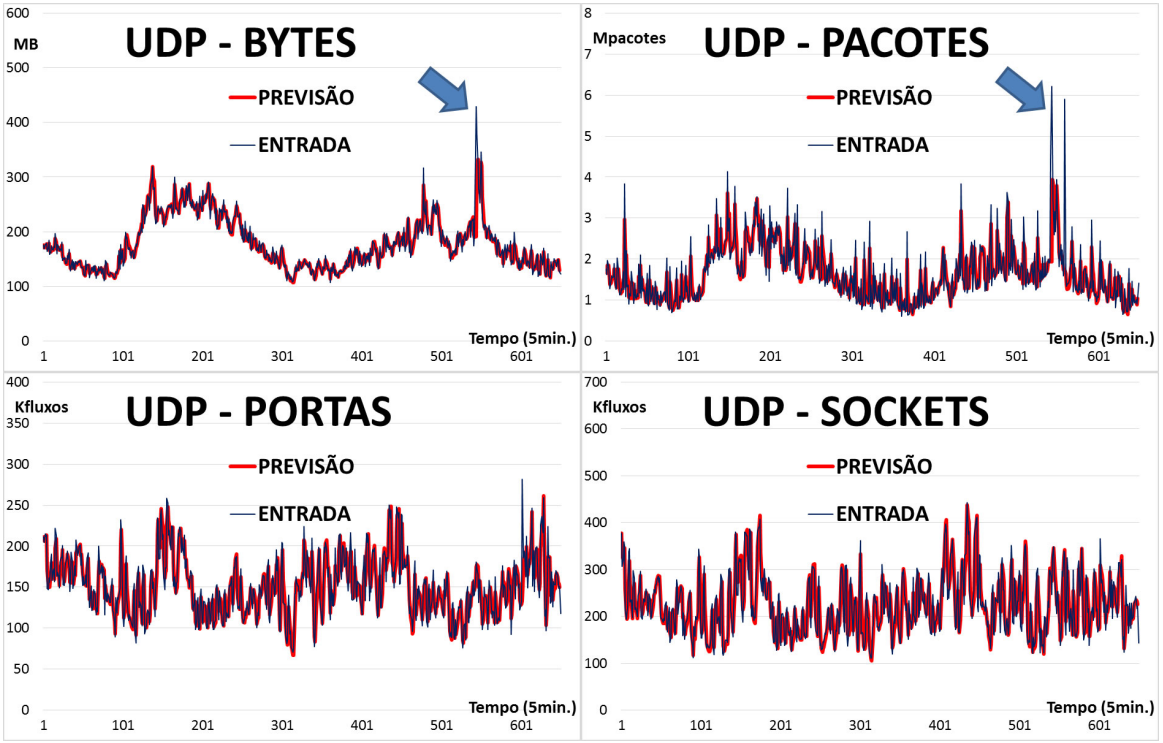

Figura 4.16: *UDP Flood* - Partição de fluxos UDP

Este ataque foi detectado pelo método nas métricas *BYTES* e *PACOTES*, e ao contrário do esperado e do que é observado nos outros ataques de negação de serviço,

o *UDP Flood* não provocou anomalias na métrica *SOCKETS DE DESTINO*. Isso aconteceu devido à maneira como o *NetFlow* agrupa fluxos *UDP*.

O protocolo *UDP* não possui sinalizadores de inicialização ou finalização de conexão, o que torna impossível ao *NetFlow* diferenciar fluxos diferentes entre as mesmas portas e endereços *IP* de origem e destino. O *NetFlow* então realiza o agrupamento de diferentes fluxos utilizando o mesmo destino e origem através do temporizador de fluxos ativos, que foi configurado com um tempo de um minuto conforme citado na Seção 3.1.

Apesar da anomalia ter sido detectada com sucesso, é necessário o ajuste da Tabela 4.4 de associação de eventos anômalos maliciosos, para que se obtenha uma correta classificação do ataque.

4.5.5 Ataque de negação de serviço *TCP Connect Flood*

A Figura 4.17 foi construída utilizando dados do terceiro período de coleta, onde foi forçado um ataque do tipo *TCP Connect Flood* a partir da estrutura dos servidores virtuais gerados pelo *AWS*. Na Figura 4.17 podem ser observados nos instantes de tempo 632 e 633 anomalias nas métricas *BYTES* e *SOCKETS DE DESTINO* da partição de fluxos *SEMI-ABERTOS*.

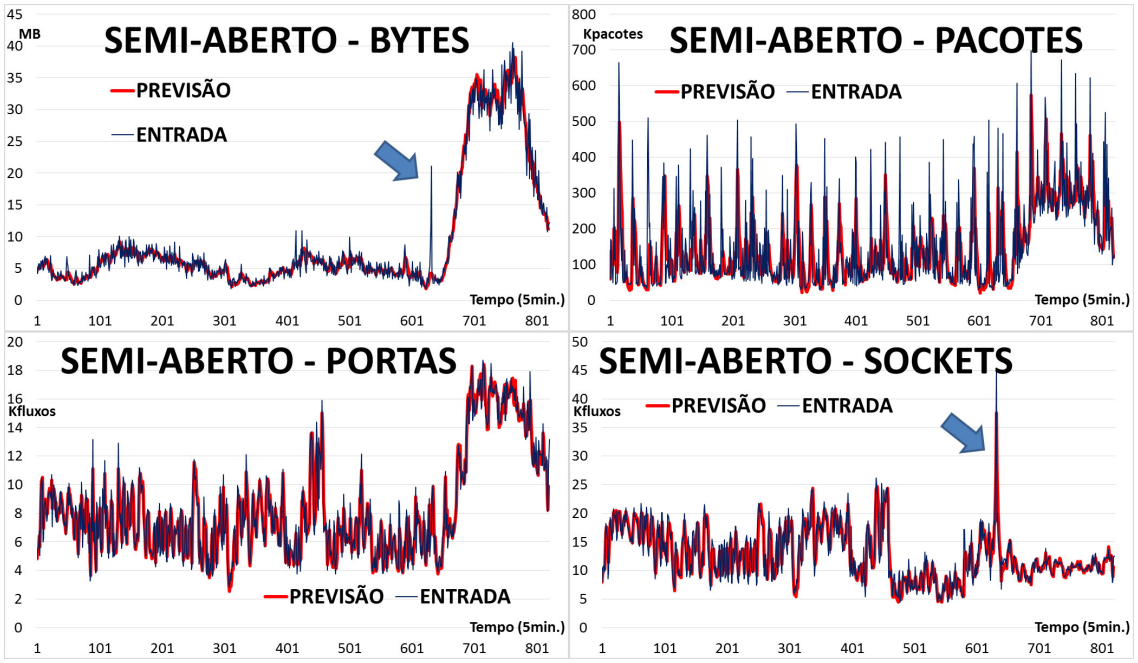

Figura 4.17: *TCP-CONNECT-FLOOD* - Partição de fluxos *SEMI-ABERTOS*

O ataque *TCP Connect Flood* possui duas variações, conforme mostrado na Seção 3.4.4, porém, só a primeira variação do ataque é mostrada aqui, devido à dificuldade de instrumentalizar a estrutura necessária para a realização de um ataque deste tipo.

De acordo com a Tabela 4.4 de associação de eventos anômalos maliciosos, o esperado neste tipo de ataque seria a detecção apenas de anomalia na métrica *SOCKETS DE DESTINO*, porém, foi detectada anomalia também na métrica *BYTES*.

Ao realizar este tipo de ataque, os servidores virtuais realizavam requisições *HTTP* para o elemento de rede alvo que respondia com a página *web* nele hospedada. Acredita-se que a detecção da anomalia na métrica *BYTES* se deva a este fato e seja completamente dependente da intensidade do ataque, portanto, não será realizada correção na Tabela 4.4 para a inclusão desta métrica como esperada na observação deste tipo de ataque.

4.5.6 Comparação entre métodos de detecção de anomalia

O método proposto por José Barbosa [52] consiste na identificação de anomalias em redes de computadores através da coleta de fluxos de redes de computadores, classificação utilizando-se as portas de destino e identificação através de redes neurais.

Foi realizada de maneira independente a análise dos fluxos de rede pertencentes ao segundo período de coleta pelo método proposto neste trabalho e pelo método proposto por José Barbosa.

Dos 3605 intervalos de tempo correspondentes ao segundo período de coleta, em cerca de 9% das medidas houve algum alarme ativado por algum dos dois métodos e em cerca de $3,5\%$ das medidas ambos os métodos geraram alarmes de maneira conjunta, conforme mostra a Figura 4.18.

Figura 4.18: Anomalias encontradas pelos métodos de detecção

O método de redes neurais proposto por José Barbosa [52] gerou 266 alarmes, e o método aqui proposto gerou 191 alarmes, o que é aproximadamente 71% da quantidade de alarmes gerados pelo método de redes neurais. Levando-se em consideração os intervalos de tempo onde houve alarme gerado por algum dos dois métodos, o método aqui proposto gerou alarmes em cerca de 58% do total. Dentre os 191 alarmes gerados pelo método proposto neste trabalho de dissertação, 127 foram também identificados pelo método de redes neurais, que em comparação corresponde a cerca de 66%. A Figura 4.19 mostra a quantidade de alarmes gerados por cada um dos métodos e por ambos os métodos dentre o total de alarmes.

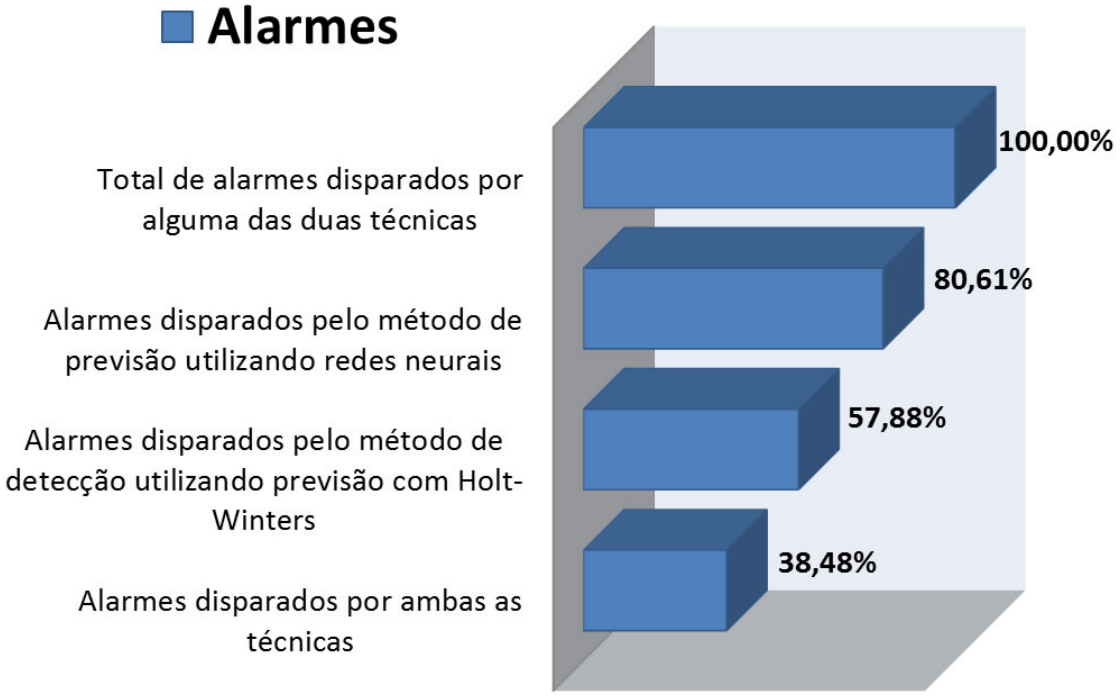

Figura 4.19: Comparação entre as quantidades de alarmes gerados

Durante o segundo período de coleta, foram gerados, entre ataques e escaneamentos de portas, 11 eventos de rede anômalos. Todos esses eventos anômalos foram gerados por dispositivos localizados fora da infraestrutura da RedeRio e tinham como alvo elementos de rede do laboratório Ravel.

Na construção desta análise, uma anomalia foi considerada detectada se durante toda a sua duração houve a geração de pelo menos um alarme. Dos 11 eventos anômalos gerados, 7 foram detectados por ambos os métodos, sendo 9 detectados pelo método de redes neurais e 8 detectados pelo método aqui proposto. De maneira individual o método aqui proposto detectou cerca de 73% dos ataques gerados e ambos os métodos foram capazes de detectar 91% dos ataques.

Na Figura 4.20 pode-se observar dentre o total de eventos anômalos gerados, os que foram detectados por ambos os métodos, apenas por um dos métodos ou por nenhum dos métodos.

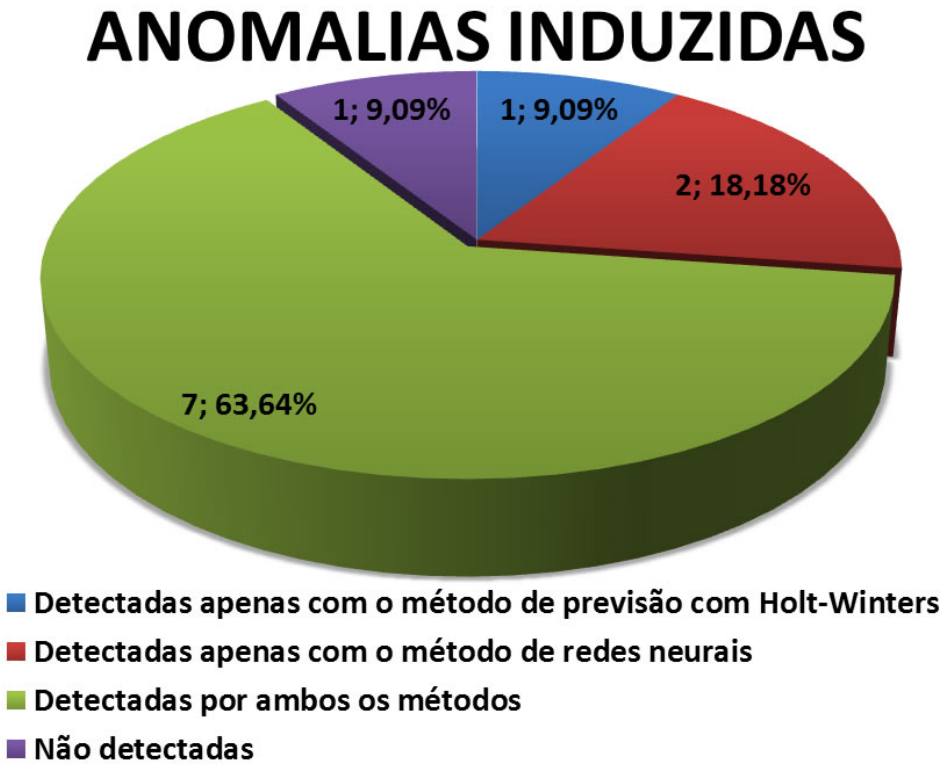

Figura 4.20: Comparação da detecção de anomalias induzidas

A falha de detecção de cerca de 27% dos casos se deve ao fato de que as anomalias foram geradas em intervalos de tempo muito próximos e o desvio d_t tende a aumentar durante a ocorrência de anomalias, o que torna o método permissivo e faz com que as anomalias não sejam alarmadas durante toda a sua duração. Tal característica pode ser corrigida tomando $\delta = 3,84$ que corresponde a um intervalo de confiança de 95% ao invés de tomar $\delta = 6,63$ que corresponde a um intervalo de confiança de 99%.

Na Figura 4.21 podemos observar três gráficos, onde o primeiro possui a métrica *SOCKETS DE DESTINO* da partição *SEMI-ABERTOS* e sua previsão, nas segunda podemos observar os intervalos de confiança com $\delta = 6,63$ e na terceira podemos observar os intervalos de confiança com $\delta = 3,84$. Tal ajuste no método de detecção seria suficiente para corrigir as falhas mas aumentaria consideravelmente o número de alarmes gerados pelo método.

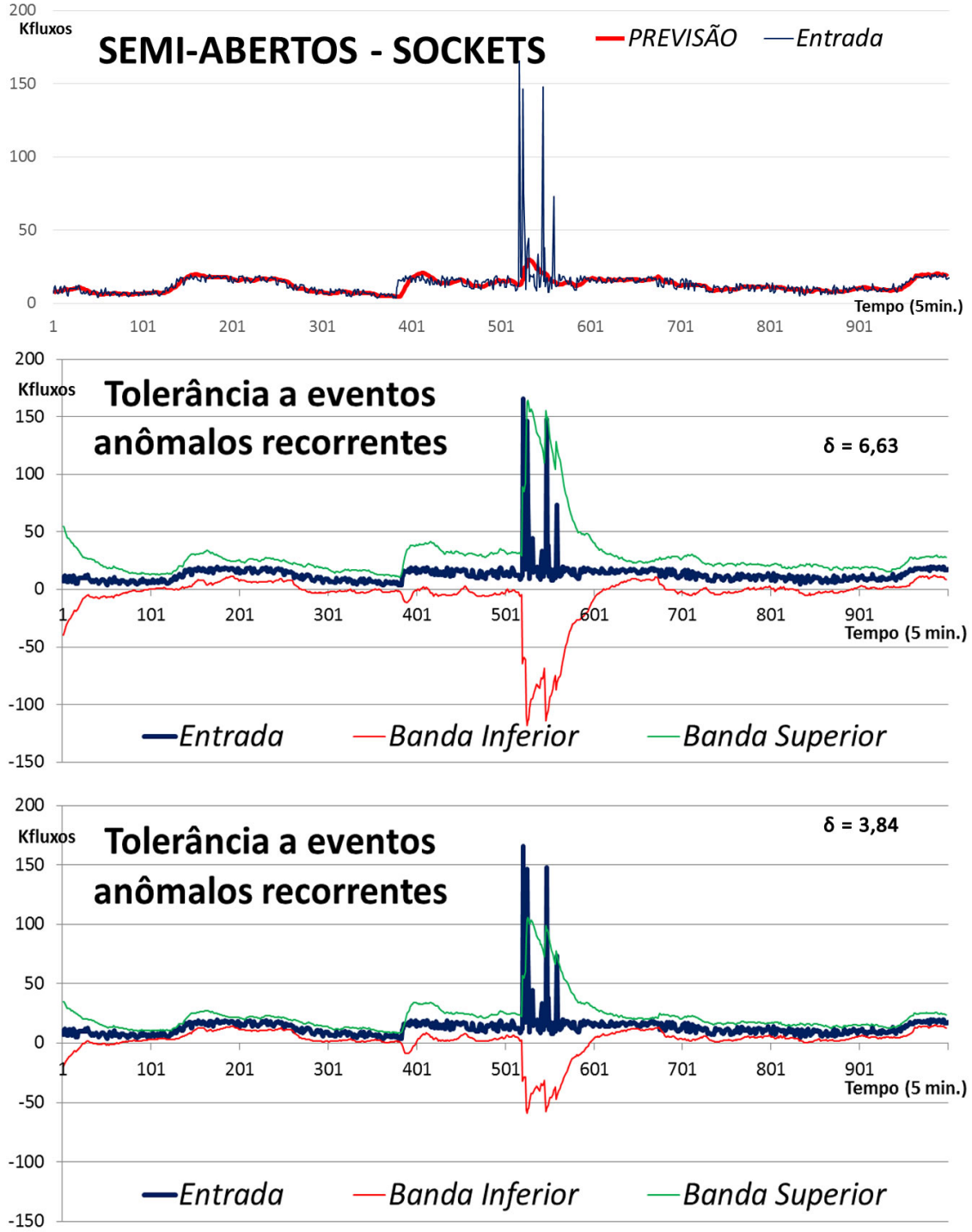

Figura 4.21: Detecção de anomalias consecutivas

Capítulo 5

Conclusão e Trabalhos Futuros

O principal objetivo deste trabalho é de construir uma metodologia que possibilite a detecção de anomalias a partir do monitoramento apenas da borda de uma rede. A rede na qual a metodologia foi testada é a RedeRio de computadores, que possui milhares de usuário conectados e uma infraestrutura de rede com conexões de 10Gbps. Detectar uma anomalia que tem como alvo apenas um dentre os milhares de usuários, identificando o tráfego anômalo que passa na borda da rede dentre vários *Gigabits* por segundo de tráfego legítimo é um grande desafio.

A classificação proposta para a utilização neste método e mostrada na Seção 3.2, teve como principal objetivo evidenciar o tráfego suspeito, classificando-o dentro de partições menores, o que permite a detecção de anomalias com maior precisão e torna possível a detecção de alguns eventos anômalos de menor amplitude.

No Seção 4.5, foi possível notar que as métricas utilizadas por este método foram capazes de quantificar determinadas características da rede que evidenciam o acontecimento de eventos anômalos maliciosos ao sofrerem grandes variações. A métrica *PORTAS DE DESTINO* que foi construída especialmente para a detecção de escaneamentos de porta teve alterações significativas em seu comportamento padrão ao serem gerados eventos deste tipo na rede, de igual forma a métrica *SOCKETS DE DESTINO* teve alterações significativas em seu comportamento ao serem gerados ataques de negação de serviço. As métricas *BYTES* e *PACOTES* tiveram um papel secundário, servindo em alguns casos para medir a intensidade de um evento anômalo, ou até mesmo contribuindo para identificar o tipo de evento que ocorre na rede.

Para estimar o comportamento da métrica, foi usada a construção de previsões através da técnica de Holt-Winters para o próximo período de tempo. Esta estimativa se mostrou adequada ao uso, pelo fato de que os valores previsto se mantiveram próximos dos valores medidos e de maneira estável.

O método aqui proposto, implementado e avaliado mostrou uma boa capacidade de detecção sendo capaz de identificar todos os ataque mais comuns realizados em

redes de computadores. Porém alguns desses ataques, foram detectados de maneira diferente da esperada, devido à características da coleta, da classificação e até mesmo dos alvos, sendo assim necessárias algumas correções na Tabela 4.4 de associação de eventos anômalos maliciosos, citadas na Seção 4.5 para que os tipos de ataques sejam identificados de maneira mais precisa. Essas alterações farão parte da implementação deste método que ficará instalada no laboratório Ravel.

Foi possível através deste método a detecção de anomalias na rede, sem que haja um excesso de alarmes sendo disparados contra o gerente de rede. O excesso de alarmes por muitas vezes pode fazer com que o método de detecção seja deixado de lado, dado que é impossível atender cada um de centenas de alarmes sendo disparados a todo o tempo.

5.1 Trabalhos Futuros

Na Seção 4.3, foi comentado que a escolha das constantes de amortecimento possui impacto na sensibilidade do método, e que constantes de amortecimento com valores mais baixos tendem a aproximar melhor o comportamento padrão de uma série. Cabe ainda a um trabalho futuro propôr uma melhor forma de realizar a escolha das constantes de amortecimento e avaliar seu impacto na distribuição dos valores de δ_t, bem como na sua capacidade de detecção de anomalias.

A natureza da classificação proposta por este método na Seção 3.2 é de dependência da resposta dos elementos alvo de um ataque. Isso se deve ao fato de que um dos elementos utilizados para a realização da classificação é a análise do sinalizador *RST* do pacote *TCP* e esse sinalizador é enviado pelo dispositivo atacado e não pelo atacante. Essa característica torna a classificação correta dos fluxos de um determinado ataque dependente da disponibilidade de dispositivo alvo, conforme pôde ser observado na Seção 4.5.3. Recomenda-se como trabalho futuro, que seja feita uma proposta de classificação que seja independente da resposta do elemento alvo e dependente apenas da análise do tráfego gerado pelo atacante.

Como sugestão para uma futura proposta, pode-se substituir as métricas utilizadas neste trabalho pelas 4 métricas propostas por Wang *et.al* no trabalho [32], a fim de comparar os resultados.

Recomenda-se também que seja realizado em tempo real o cálculo do δ, que é obtido através do teste de detecção de anomalias exposto em 3.4.2. Esse cálculo será realizado utilizando a média amostral dos valores δ_t e o cálculo do valor do quantil da distribuição qui-quadrado com o nível de confiança escolhido.

Surgere-se também como trabalho futuro a avaliação de detecção de anomalias utilizando outros modelos de previsão em séries temporais, tais como os modelos ARIMA e SARIMA, bem como sua comparação com os resultados aqui obtidos.

Tais modelos foram contemplados por Evandro Macedo em [53], mostrando uma boa capacidade para a realização de estimativas de valores futuros em séries temporais.

Referências Bibliográficas

[1] SCHNEIER, B. "China's Great Cannon - Schneier on Security". Julho de 2015. Disponível em: <https://www.schneier.com/blog/archives/2015/04/chinas_great_ca.html>.

[2] IT, F. "Deep dive into QUANTUM INSERT — Fox-IT International blog". Julho de 2015. Disponível em: <http://blog.fox-it.com/2015/04/20/deep-dive-into-quantum-insert/>.

[3] KUROSE, J. F., ROSS, K. W. *Computer Networking: A Top-Down Approach*. 5th ed. USA, Addison-Wesley Publishing Company, 2009. ISBN: 0136079679, 9780136079675.

[4] IETF. "The Internet Engineering Task Force (IETF®)". Novembro de 2014. Disponível em: <https://www.ietf.org/>.

[5] RFC. "Request for Comments(RFC)Pages". Dezembro de 2014. Disponível em: <http://www.ietf.org/rfc.html>.

[6] TANENBAUM, A. S., WETHERALL, D. J. *Computer Networks*. 5th ed. USA, Massachusetts, Boston, Prentice Hall, 2011. ISBN: 978-0-13-212695-3.

[7] JUNIPER. "Juniper Flow Monitoring". Fevereiro de 2015. Disponível em: <http://kb.juniper.net/library/CUSTOMERSERVICE/GLOBAL_JTAC/BK16677/3500204-en.pdf>.

[8] CISCO. "Cisco IOS NetFlow - Cisco". Fevereiro de 2015. Disponível em: <http://www.cisco.com/c/en/us/products/ios-nx-os-software/ios-netflow/index.html>.

[9] SFLOW.ORG. "sFlow.org - Making the Network Visible". Fevereiro de 2015. Disponível em: <http://www.sflow.org/index.php>.

[10] MARCHETTE, D. J. *Computer Intrusion Detection and Network Monitoring: A Statistical Viewpoint*, v. 1, *Statistics for Engineering and Information Science*. 1 ed. New York, USA, Springer New York,

2001. ISBN: 978-1-4419-2937-2,978-1-4757-3458-4. doi: 10.1007/978-1-4757-3458-4. Disponível em: <http://gen.lib.rus.ec/book/index.php?md5=1caf2f245e4fbe47dff6d3cb1bbfc269>.

[11] REDERIO. "Bem-Vindo ao RedeRio — RedeRio". Janeiro de 2015. Disponível em: <http://www.rederio.br/site/>.

[12] RAVEL. "IPTraf". Fevereiro de 2015. Disponível em: <http://200.20.185.2/cgi-bin/graficos.pl>.

[13] RAVEL. "IPTraf". Fevereiro de 2015. Disponível em: <http://iptraf.ravel.ufrj.br/>.

[14] SILVA, C. D. A. *Proposta de Implementação de Uma Ferramenta Para Gerência em Redes Baseada Numa Nova Metodologia Usando Análise de Tráfego em Backbones IP*. Dissertação de Mestrado, COPPE/UFRJ, Rio de Janeiro, RJ, Brasil, 2006.

[15] RAVEL. "IPTraf". Fevereiro de 2015. Disponível em: <http://iptraf.ravel.ufrj.br/anomalia/>.

[16] CARL, G., KESIDIS, G., BROOKS, R., et al. "Denial-of-service attack-detection techniques", *Internet Computing, IEEE*, v. 10, n. 1, pp. 82–89, Jan 2006. ISSN: 1089-7801. doi: 10.1109/MIC.2006.5.

[17] RAO, S., REED, M. *InfoSec Reading Room*.

[18] GADGE, J., PATIL, A. "Port scan detection". In: *Networks, 2008. ICON 2008. 16th IEEE International Conference on*, pp. 1–6, Dec 2008. doi: 10.1109/ICON.2008.4772622.

[19] LYON, G. "Nmap Reference Guide (Man Pages)". Janeiro de 2015. Disponível em: <http://nmap.org/man/>.

[20] CHATFIELD, C. *Time-Series Forecasting*. CHAPMAN & HALL/CRC, 2000. ISBN: 1-58488-063-5.

[21] MORETTIN, P., DE CASTRO TOLOI, C. *Modelos para Previsão de Séries Temporais*. Modelos para Previsão de Séries Temporais. Instituto de Matemática Pura e Aplicada, 1981.

[22] CHATFIELD, C., YAR, M. "Holt-Winters Forecasting: Some Practical Issues", *Journal of the Royal Statistical Society. Series D (The Statistician)*, v. 37, n. 2, pp. pp. 129–140, 1988. ISSN: 00390526. Disponível em: <http://www.jstor.org/stable/2348687>.

[23] NORTHCUTT, S. *Network Intrusion Detection: An Analyst's Handbook*. Thousand Oaks, CA, USA, New Riders Publishing, September 2003. ISBN: 0735708681.

[24] SABAHI, F., MOVAGHAR, A. "Intrusion Detection: A Survey". In: *Systems and Networks Communications, 2008. ICSNC '08. 3rd International Conference on*, pp. 23–26, Oct 2008. doi: 10.1109/ICSNC.2008.44.

[25] SCARFONE, K., MELL, P. "Guide to Intrusion Detection and Prevention Systems (IDPS)", *NIST Special Publication 800-94*, February 2007.

[26] MUKHERJEE, B., HEBERLEIN, L., LEVITT, K. "Network intrusion detection", *Network, IEEE*, v. 8, n. 3, pp. 26–41, May 1994. ISSN: 0890-8044. doi: 10.1109/65.283931.

[27] MAHONEY, M. V. *A Machine Learning Approach to Detecting Attacks by Identifying Anomalies in Network Traffic*. Tese de Doutorado, Melbourne, FL, USA, 2003. AAI3081393.

[28] AKGUL, T., BAYKUT, S., EROL-KANTARCI, M., et al. "Periodicity-Based Anomalies in Self-Similar Network Traffic Flow Measurements", *Instrumentation and Measurement, IEEE Transactions on*, v. 60, n. 4, pp. 1358–1366, April 2011. ISSN: 0018-9456. doi: 10.1109/TIM.2010.2084711.

[29] YAN, W., HOU, E., ANSARI, N. "Anomaly detection and traffic shaping under self-similar aggregated traffic in optical switched networks". In: *Communication Technology Proceedings, 2003. ICCT 2003. International Conference on*, v. 1, pp. 378–381 vol.1, April 2003. doi: 10.1109/ICCT.2003.1209102.

[30] SONG, S., LING, L., MANIKOPOULO, C. "Flow-based Statistical Aggregation Schemes for Network Anomaly Detection". In: *Networking, Sensing and Control, 2006. ICNSC '06. Proceedings of the 2006 IEEE International Conference on*, pp. 786–791, 2006. doi: 10.1109/ICNSC.2006.1673246.

[31] CHAPPLE, M., WRIGHT, T., WINDING, R. "Flow Anomaly Detection in Firewalled Networks". In: *Securecomm and Workshops, 2006*, pp. 1–6, Aug 2006. doi: 10.1109/SECCOMW.2006.359576.

[32] HONG, W., ZHENGHU, G., QING, G., et al. "Detection Network Anomalies Based on Packet and Flow Analysis". In: *Networking, 2008. ICN 2008. Seventh International Conference on*, pp. 497–502, April 2008. doi: 10.1109/ICN.2008.83.

[33] DE MOURA, A. S. *DETECÇÃO DE ANOMALIAS EM REDES WAN USANDO ESTIMATIVA DE HOLT-WINTERS APLICADA A MEDIDAS DE ENTROPIA*. Dissertação de Mestrado, UNIVERSIDADE FEDERAL DO ESTADO DO RIO DE JANEIRO, RIO DE JANEIRO, RJ – BRASIL, Setembro 2009.

[34] BRUTLAG, J. D. "Aberrant Behavior Detection in Time Series for Network Monitoring". In: *Proceedings of the 14th USENIX Conference on System Administration*, LISA '00, pp. 139–146, Berkeley, CA, USA, 2000. USENIX Association. Disponível em: <http://dl.acm.org/citation.cfm?id=1045502.1045530>.

[35] OETIKER, T. "RRDtool - About RRDtool". Fevereiro de 2015. Disponível em: <http://oss.oetiker.ch/rrdtool/index.en.html>.

[36] NGUYEN, H. A., TAM VAN NGUYEN, T., KIM, D. I., et al. "Network traffic anomalies detection and identification with flow monitoring". In: *Wireless and Optical Communications Networks, 2008. WOCN '08. 5th IFIP International Conference on*, pp. 1–5, May 2008. doi: 10.1109/WOCN.2008.4542524.

[37] CISCO. "Cisco ASR 9000 Series Aggregation Services Routers - Products & Services - Cisco". Fevereiro de 2015. Disponível em: <http://www.cisco.com/c/en/us/products/routers/asr-9000-series-aggregation-services-routers/index.html>.

[38] CISCO. "Cisco Systems, Inc". Fevereiro de 2015. Disponível em: <http://www.cisco.com/>.

[39] CISCO. *Introduction to Cisco IOS® NetFlow*. Relatório técnico, Fevereiro de 2015. Disponível em: <http://www.cisco.com/c/en/us/products/collateral/ios-nx-os-software/ios-netflow/prod_white_paper0900aecd80406232.pdf>.

[40] CISCO. "Configuring NetFlow on Cisco ASR 9000 Series Aggregation Services Router". Fevereiro de 2015. Disponível em: <http://www.cisco.com/c/en/us/td/docs/routers/asr9000/software/asr9k_r4-0/netflow/configuration/guide/nfc40/nfc40flow.pdf>.

[41] UBUNTU. "About Ubuntu — Ubuntu". Julho de 2015. Disponível em: <http://www.ubuntu.com/about/about-ubuntu>.

[42] "NfSen". Maio de 2015. Disponível em: <http://nfsen.sourceforge.net/>.

[43] "NFDUMP". Maio de 2015. Disponível em: <http://nfdump.sourceforge.net/>.

[44] PERL.ORG. "About Perl - www.perl.org". Julho de 2015. Disponível em: <https://www.perl.org/about.html>.

[45] LINUX.ORG. "BASH—Linux.org". Julho de 2015. Disponível em: <http://www.linux.org/resources/bash.25/>.

[46] PAPAGIANNAKI, K., TAFT, N., ZHANG, Z.-L., et al. "Long-term forecasting of Internet backbone traffic: observations and initial models". In: *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, v. 2, pp. 1178–1188 vol.2, March 2003. doi: 10.1109/INFCOM.2003.1208954.

[47] WARD, A., GLYNN, P., RICHARDSON, K. "Internet Service Performance Failure Detection", *SIGMETRICS Perform. Eval. Rev.*, v. 26, n. 3, pp. 38–43, dez. 1998. ISSN: 0163-5999. doi: 10.1145/306225.306237. Disponível em: <http://doi.acm.org/10.1145/306225.306237>.

[48] AMAZON. "Amazon Web Services(AWS) - Serviços de computação em nuvem". Junho de 2015. Disponível em: <https://aws.amazon.com/pt/>.

[49] FOUNDATION., . T. R. "R: What is R?" Junho de 2015. Disponível em: <https://www.r-project.org/about.html>.

[50] LEON-GARCIA, A. *Probability, Statistics, and Random Processes For Electrical Engineering*. 3rd ed. Upper Saddle River, NJ, USA, Pearson Prentice Hall, Pearson Education, 2008. ISBN: 978-0-13-147122-1.

[51] MORRIS H. DEGROOT, M. J. S. *Probability and Statistics*. 4th ed. Boston, MA, USA, Addison-Wesley, Pearson Education, 2012. ISBN: 978-0-321-50046-5.

[52] DA SILVA FILHO, J. B. *Identificação de Anomalias em Fluxos de Rede Utilizando Métodos de Inteligência Artificial*. Dissertação de Mestrado em andamento, COPPE/UFRJ, Rio de Janeiro, RJ, Brasil, Setembro 2015.

[53] MACEDO, E. L. C. *Previsão de Tráfego em Enlaces de Redes Utilizando Séries Temporais*. Dissertação de Mestrado em andamento, COPPE/UFRJ, Rio de Janeiro, RJ, Brasil, Setembro 2015.

[54] "Welcome to The Apache Software Foundation!" Maio de 2015. Disponível em: <http://www.apache.org/>.

[55] PRAETOX. “praetox.com”. Maio de 2015. Disponível em: <http://praetox.com/>.

[56] LYON, G. “Gordon ’Fyodor’ Lyon”. Janeiro de 2015. Disponível em: <http://insecure.org/fyodor/>.

Apêndice A

Teste de Kolmogorov-Smirnov

Em alguns problemas, os dados observados apresentam características similares à algumas distribuições específicas. Nestes casos, podem ser realizados testes [51] que avaliam a qualidade do ajuste destas distribuições.

Neste trabalho foi utilizado o teste de Kolmogorov-Smirnov para a validação da distribuição qui-quadrado como aproximação da distribuição empírica dos valores de δ_t.

Suponha que as variáveis aleatórias $\Delta_1, ..., \Delta_n$ formem uma amostra aleatória de alguma distribuição contínua e seja $\delta_1, ..., \delta_n$ os valores observados de $\Delta_1, ..., \Delta_n$. Vamos considerar a função $F_n(\delta)$ construída a partir dos valores de $\delta_1, ..., \delta_n$ da seguinte forma.

Para cada δ o valor da $F_n(\delta)$ é definido como a proporção dos valores observados na amostra que são menores ou iguais a δ. Em outras palavras, se exatamente k dos valores observados na amostra forem menores ou iguais a δ, então $F_n(\delta) = k/n$. A função $F_n(\delta)$ definida desta forma é denominada função de distribuição amostral, ou simplesmente distribuição empírica.

A distribuição empírica $F_n(\delta)$ pode ser obtida como uma distribuição discreta que possui probabilidade $1/n$ para cada um dos n valores de $\delta_1, ..., \delta_n$. Portanto, $F_n(\delta)$ será uma função degrau com um salto de magnitude $1/n$ para cada ponto $\delta_i (i = 1, ..., n)$, conforme mostrado na Figura A.1.

Segue da Lei dos Grandes Números que quando $n \to \infty$, a proporção $F_n(\delta)$ de observações na amostra que são menores ou iguais a δ irá convergir em probabilidade para $F_n(\delta)$. Esta relação expressa o fato de que para cada ponto δ, a distribuição $F_n(\delta)$ irá convergir para a atual distribuição $F(\delta)$ de cada uma das amostras tomadas. Mais precisamente,

$$D_n = sup \mid F_n(\delta) - F(\delta) \mid \text{ , onde } D_n \xrightarrow{p} 0 \tag{A.1}$$

Com o objetivo de testar agora se a função de densidade $F(\delta)$ segue uma deter-

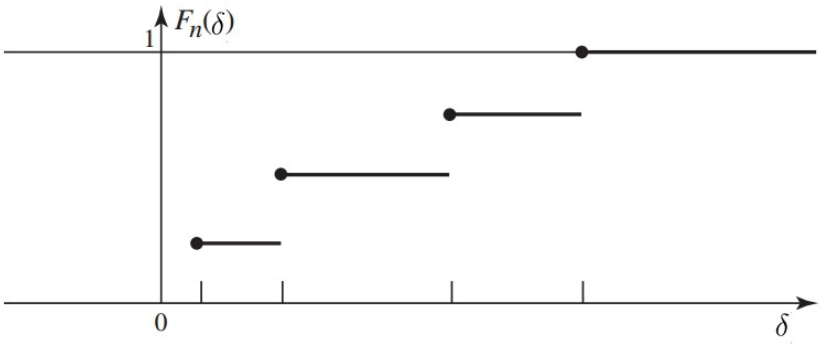

Figura A.1: Exemplo de distribuição empírica - Retirado de [51]

minada distribuição contínua $F^*(\delta)$, foi formulada a seguinte hipótese:

$$H_0 : F(\delta) = F^*(\delta), \text{para} - \infty < \delta < \infty \tag{A.2}$$

H_1: A hipótese nula não é verdadeira.

Como definido anteriormente D_n será a diferença máxima entre a distribuição empírica e a função de distribuição hipotética $F^*(\delta)$. Quando a hipótese nula for verdadeira a distribuição de probabilidade de D_n será uma certa distribuição que é a mesma para todo valor possível de $F^*(\delta)$. Esta distribuição é mostrada na Tabela A.1.

Segue da lei dos grande números que o valor de D_n tende a ser pequeno se a hipótese nula for verdadeira. Portanto um procedimento de teste razoável para as hipóteses descritas é rejeitar H_0 se $n^{\frac{1}{2}}D_n > c$, onde c é uma constante apropriada comumente obtida com a consulta na Tabela A.1.

O procedimento de teste que rejeita H_0 quando $n^{\frac{1}{2}}D_n > c$ é chamado de Kolmogorov-Smirnov [51].

Tabela A.1: Tabela da distribuição H - Retirada de [51]

t	$H(t)$	t	$H(t)$	t	$H(t)$	t	$H(t)$
0,30	0,0000	0,75	0,3728	1,20	0,8878	1,80	0,9969
0,35	0,0003	0,80	0,4559	1,25	0,9121	1,90	0,9985
0,40	0,0028	0,85	0,5347	1,30	0,9319	2,00	0,9993
0,45	0,0126	0,90	0,6073	1,35	0,9478	2,10	0,9997
0,50	0,0361	0,95	0,6725	1,40	0,9603	2,20	0,9999
0,55	0,0772	1,00	0,7300	1,45	0,9702	2,30	0,9999
0,60	0,1357	1,05	0,7798	1,50	0,9778	2,40	1,0000
0,65	0,2080	1,10	0,8223	1,60	0,9880	2,50	1,0000
0,70	0,2888	1,15	0,8580	1,70	0,9938		

Apêndice B

Descrição dos Experimentos

Neste Apêndice serão descritas as formas pelas quais os ataques de negação de serviço e escaneamentos de portas foram gerados.

B.1 Ataques de Negação de Serviço

A infraestrutura utilizada para a realização dos ataques de negação de serviço consiste em dez servidores virtuais gerados através do serviço de computação elástica na nuvem provido pela *Amazon Web Services* ou AWS [48]. Esses servidores consistem em máquinas virtuais com *hardware* com recursos computacionais emulados pelo *AWS*. Em cada um dos dez servidores foi emulada uma arquitetura com processador de 10 núcleos com $2,5$GHz e 1GB de memória, e com capacidade de rede de até 100Mbps.

A infraestrutura utilizada para receber o ataque consiste em um servidor localizado no laboratório Ravel com o sistema operacional *Linux Ubuntu Server* 14 [41] instalado bem como o servidor *web* Apache [54]. O *hardware* do servidor consiste em um processador de $2,8$GHz, 2GB de memória e conexão de rede *Fast Ethernet* com capacidade de até 100 Mbps.

Em cada um desses servidores, foi carregado o aplicativo LOIC que é uma ferramenta de teste de estresse de rede e de ataque de negação de serviço produzido pela PRAETOX [55] e tem sua imagem mostrada na Figura B.1.

Para a geração de ataques de negação de serviço do tipo *UDP flood* foi utilizado o LOIC com o parâmetro "Method"com a opção *UDP*. Nessa opção o aplicativo envia pacotes *UDP* com o conteúdo indicado no campo "*TCP/UDP message*"na velocidade indicada. As mensagens eram da ordem de pouco menos de 64KB que é o tamanho máximo do pacote *UDP* e a velocidade foi configurada para a maior possível com 10 *threads*.

Para a geração de ataques de negação de serviço do tipo *TCP SYN flood* foi utilizado o LOIC com o parâmetro "Method"com a opção *TCP* e com a caixa

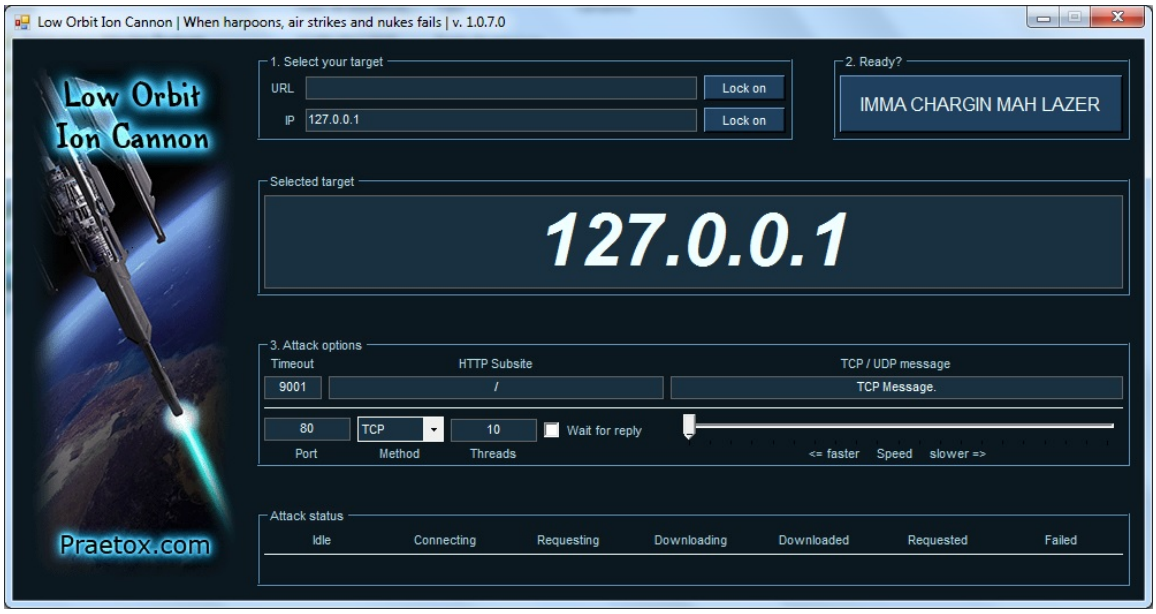

Figura B.1: Aplicativo *LOIC*

"*Wait for reply*"desmarcada. Nessa opção o aplicativo envia apenas pacotes de requisição de conexão do *TCP*. O aplicativo foi configurado para enviar os pacotes na velocidade mais alta o possível e utilizou 10 *threads*.

Para a geração de ataques de negação de serviço do tipo *TCP SYN Connect* foi utilizado o LOIC com o parâmetro "Method"com a opção *HTTP* e com a caixa "*Wait for reply*"marcada. Com essas opções o aplicativo realiza conexões *TCP* de maneira completa e envia requisições *HTTP* da página *Web* do servidor alvo que é indicada no campo "*HTTP Subsite*". O aplicativo foi configurado para realizar as requisições com a maior velocidade possível usando 10 *threads*.

B.2 Escaneamento de portas

Com o objetivo de realizar os escaneamentos de portas, foi utilizada a ferramenta Nmap [19] que é gratuita, de código aberta e escrita por Gordon Lyon[56].

Os escaneamentos foram realizados em toda infraestrutura de rede do laboratório Ravel, que consiste em duas redes *IP*, com 320 endereços. Os escaneamentos foram executados através das seguintes linhas de comando:

- *FIN Scan* - nmap -sF IP1/RANGE1; nmap -sF IP2/RANGE2
- *TCP Connect Scan* - nmap -sS IP1/RANGE1; nmap -sS IP2/RANGE2
- *TCP SYN Scan* - nmap -sT IP1/RANGE1; nmap -sT IP2/RANGE2

Onde, IP1/RANGE1 e IP2/RANGE2 endereçam as redes do laboratório Ravel.