

Universidade Estadual de Campinas  
Faculdade de Engenharia Elétrica e de Computação

## **Detecção de Anomalias em Redes de Computadores**

**Autor: Bruno Bogaz Zarpelão**

**Orientador: Prof. Dr. Leonardo de Souza Mendes**

**Co-orientador: Prof. Dr. Mario Lemes Proença Junior**

**Tese de Doutorado** Apresentada a Faculdade de Engenharia Elétrica e de Computação como parte dos requisitos para obtenção do título de Doutor em Engenharia Elétrica. Área de Concentração: **Telecomunicações e Telemática.**

### **Banca Examinadora**

Leonardo de Souza Mendes, Dr.	DECOM/FEEC/UNICAMP
Rodolfo Miranda de Barros, Dr.	DC/UEL
Taufik Abrão, Dr.	DEEL/UEL
Maurício Ferreira Magalhães, Dr.	DCA/FEEC/UNICAMP
Renato Baldini Filho, Dr.	DECOM/FEEC/UNICAMP

Campinas, SP

Setembro/2010

FICHA CATALOGRÁFICA ELABORADA PELA  
BIBLIOTECA DA ÁREA DE ENGENHARIA E ARQUITETURA - BAE - UNICAMP

Z19d	<p>Zarpelão, Bruno Bogaz Detecção de anomalias em redes de computadores / Bruno Bogaz Zarpelão. --Campinas, SP: [s.n.], 2010.</p> <p>Orientadores: Leonardo de Souza Mendes, Mario Lemes Proença Junior. Tese de Doutorado - Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação.</p> <p>1. Redes de computação - Gerência. 2. Anomalias. 3. Alarmes. 4. Sistemas de segurança. 5. Telecomunicações - Tráfego. I. Mendes, Leonardo de Souza. II. Proença Junior, Mario Lemes. III. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. IV. Título.</p>
------	---

Título em Inglês: Anomaly detection in computer networks

Palavras-chave em Inglês: Computer networks - Management, Anomaly, Alarm,  
Security systems, Telecommunication - Traffic

Área de concentração: Telecomunicações e Telemática

Titulação: Doutor em Engenharia Elétrica

Banca examinadora: Rodolfo Miranda de Barros, Taufik Abrão, Maurício Ferreira  
Magalhães, Renato Baldini Filho

Data da defesa: 23/09/2010

Programa de Pós Graduação: Engenharia Elétrica

## **COMISSÃO JULGADORA - TESE DE DOUTORADO**

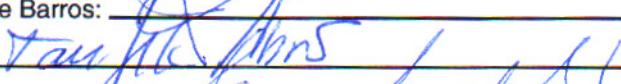
**Candidato:** Bruno Bogaz Zarpelão

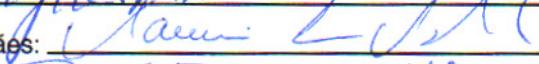
**Data da Defesa:** 23 de setembro de 2010

**Título da Tese:** "Detecção de anomalias em redes de computadores"

Prof. Dr. Leonardo de Souza Mendes (Presidente): 

Prof. Dr. Rodolfo Miranda de Barros: 

Prof. Dr. Taufik Abrão: 

Prof. Dr. Maurício Ferreira Magalhães: 

Prof. Dr. Renato Baldini Filho: 

## Resumo

Anomalias em redes de computadores são desvios súbitos e acentuados que ocorrem no tráfego em consequência de diversas situações como defeitos em softwares, uso abusivo de recursos da rede, falhas em equipamentos, erros em configurações e ataques. Nesta tese, é proposto um sistema de detecção de anomalias em redes de computadores baseado em três níveis de análise. O primeiro nível de análise é responsável por comparar os dados coletados em um objeto SNMP (*Simple Network Management Protocol*) com o perfil de operações normais da rede. O segundo nível de análise correlaciona os alarmes gerados no primeiro nível de análise utilizando um grafo de dependências que representa as relações entre os objetos SNMP monitorados. O terceiro nível de análise reúne os alarmes do segundo nível utilizando informações sobre a topologia de rede e gera um alarme de terceiro nível que reporta a propagação da anomalia pela rede. Os testes foram realizados na rede da Universidade Estadual de Londrina, utilizando situações reais. Os resultados mostraram que a proposta apresentou baixas taxas de falsos positivos combinadas a altas taxas de detecção. Além disso, o sistema foi capaz de correlacionar alarmes gerados para diferentes objetos SNMP em toda a rede, produzindo conjuntos menores de alarmes que ofereceram ao administrador de redes uma visão panorâmica do problema.

**Palavras-chave:** Redes de computação – Gerência, Anomalias, Alarmes, Sistemas de segurança, Telecomunicações – Tráfego.

# **Abstract**

Anomalies in computer networks are unexpected and significant deviations that occur in network traffic due to different situations such as software bugs, unfair resource usage, failures, misconfiguration and attacks. In this work, it is proposed an anomaly detection system based on three levels of analysis. The first level of analysis is responsible for comparing the data collected from SNMP (Simple Network Management Protocol) objects with the profile of network normal behavior. The second level of analysis correlates the alarms generated by the first level of analysis by using a dependency graph, which represents the relationships between the SNMP objects. The third level of analysis correlates the second level alarms by using network topology information. The third level generates a third level alarm that presents the anomaly propagation path through the network. Tests were performed in the State University of Londrina network, exploring real situations. Results showed that the proposal presents low false positive rates and high detection rates. Moreover, the proposed system is able to correlate alarms that were generated for SNMP objects at different places of the network, producing smaller sets of alarms that offer a wide-view of the problem to the network administrator.

**Keywords:** Computer networks - Management, Anomaly, Alarm, Security systems, Telecommunication – Traffic.

*“Valeu a pena? Tudo vale a pena  
Se a alma não é pequena.  
Quem quer passar além do Bojador  
Tem que passar além da dor.”*

(Mar Português, Fernando Pessoa)

*À minha esposa Juliana, aos meus pais José Roberto e Helena e à minha irmã Letícia.*

## Agradecimentos

A Deus e a Nossa Senhora Aparecida.

A minha esposa Juliana, pelo amor, carinho e compreensão. Obrigado por me apoiar desde o primeiro dia, principalmente nos momentos mais difíceis. Esta conquista, você sabe, também é sua.

Aos meus pais, José Roberto e Helena, que sempre deram total prioridade aos meus estudos. A vocês dois e a minha irmã, Letícia, agradeço pelo apoio, amor e carinho.

Ao meu orientador Prof. Leonardo pelo suporte, pelas diversas oportunidades e por toda a confiança depositada.

Ao meu co-orientador Prof. Mario, que me acompanha desde a iniciação científica, também agradeço pelo suporte, pelas oportunidades e pela confiança no meu trabalho. Saiba que a minha opção pela vida acadêmica é culpa sua!

Ao Prof. Joel Rodrigues, meu orientador durante o período sanduíche no Instituto de Telecomunicações em Portugal, agradeço pela oportunidade e por sua preocupação em me fazer sentir em casa, mesmo estando tão longe de casa.

Ao meu grande amigo Rodrigo Miani, pelas revisões, discussões e ponderados conselhos, o que influenciou diretamente neste trabalho. Agradeço por ser sempre um bom companheiro, presente em todas as horas, mas com um péssimo defeito: ser são paulino.

Ao meu grande amigo e padrinho de casamento José Henrique, sempre prestativo, seja para resolver enormes problemas, seja para tomarmos uma cerveja e fazermos um churrasco.

Ao meu grande amigo Marcelo Cubas, parceiro nos bons momentos e em várias missões impossíveis. Agradeço pelo incentivo constante e pelos inúmeros conselhos.

Aos amigos do LaRCom, pelos ótimos momentos. Em especial: Ekler, Márlon, Dhérik, Zanoni, André Rosot, Felipe, André Panhan, Maurício e Gean. Valeu, pessoal!

Aos amigos do Instituto de Telecomunicações-Portugal, agradeço pela hospitalidade. Em especial, agradeço ao João Alfredo e ao Pedro Rosa pela amizade que construímos. Ainda volto à Covilhã para tomar uma mini no Leões com vocês!

À Fundação de Amparo à Pesquisa do Estado de São Paulo, FAPESP, pelo suporte financeiro que permitiu a realização deste trabalho (Processo número 05/52973-6).

# Sumário

Lista de Figuras .....	xvii
Lista de Tabelas .....	xxi
Glossário.....	xxiii
Lista de Símbolos .....	xxv
1    Introdução.....	1
2    Principais definições.....	7
2.1    Anomalias e suas principais causas .....	7
2.2    Coleta de informações da rede.....	10
2.2.1    Estatísticas obtidas com o SNMP .....	10
2.2.2    Fluxos de pacotes .....	13
2.2.3    Estatísticas coletadas por sondas ( <i>network probes</i> ).....	14
2.2.4    Cabeçalhos de pacotes.....	15
2.2.5    Topologia de redes .....	16
2.3    Detecção de anomalias.....	16
2.3.1    Detecção baseada nas assinaturas das anomalias .....	17
2.3.2    Detecção baseada na caracterização do comportamento normal.....	18
2.4    Localização de anomalias .....	21
2.4.1    Sistemas especialistas .....	22
2.4.2    Sistemas baseados na modelagem da rede .....	23
2.4.3    Sistemas baseados em modelos de propagação das anomalias .....	23
2.5    Tomografia de redes .....	24

2.6	Trabalhos relacionados .....	25
3	Caracterização de tráfego: modelo BLGBA e DSNS.....	31
4	Detecção de anomalias em três níveis .....	39
4.1	Definição parametrizada de anomalia.....	39
4.2	Arquitetura da solução proposta .....	41
4.3	Módulo de Detecção de Anomalias .....	45
4.3.1	Módulo de Análise de Objeto SNMP .....	46
4.3.2	Módulo de Correlação .....	51
4.4	Módulo de Configuração Automática.....	70
4.5	Módulo de Localização de Anomalias.....	75
5	Implementação e resultados.....	79
5.1	Ambiente de rede monitorado.....	79
5.2	Desenvolvimento com a ferramenta GBA.....	81
5.3	Ajustes no modelo para implantação na rede da UEL.....	84
5.4	Resultados do Módulo de Detecção de Anomalias .....	89
5.5	Resultados do Módulo de Configuração Automática .....	97
5.6	Casos reais de anomalias .....	101
6	Conclusão .....	117
7	Bibliografia .....	123
Anexo A .....		129
A.1	Anomalias do grupo A .....	129
A.2	Anomalias do grupo B .....	134

# Lista de Figuras

Figura 2.1 - Interações previstas no protocolo SNMP entre o gerente e o agente.....	11
Figura 2.2 - Informações da MIB dispostas de forma hierárquica (MAURO; SCHMIDT, 2001).....	12
Figura 3.1 - Monitoramento de fim de semana no servidor Proxy da UEL, objeto <i>ipInReceives</i> .....	34
Figura 3.2 - Monitoramento de dias úteis no servidor Proxy da UEL, objeto <i>ipInReceives</i> .35	35
Figura 3.3 - Monitoramento de fim de semana com os respectivos DSNS no servidor Proxy da UEL, objeto <i>ipInReceives</i> .....	36
Figura 3.4 - Monitoramento de dias úteis com os respectivos DSNS no servidor Proxy da UEL, objeto <i>ipInReceives</i> .....	37
Figura 4.1 - Fluxo da informação e níveis de análise no sistema proposto.....	43
Figura 4.2 - Visão geral do modelo proposto para tratamento de anomalias. ....	44
Figura 4.3 - Ilustração do mecanismo de histerese do RMON.....	47
Figura 4.4 - Primeira proposta de mecanismo de histerese para o Módulo de Análise de Objeto SNMP. ....	48
Figura 4.5 – Diagrama de atividades para o algoritmo do Módulo de Análise de Objeto SNMP. ....	50
Figura 4.6 - Diagrama de Case para o grupo <i>interface</i> .....	56
Figura 4.7 - Diagrama de Case para o grupo <i>ip</i> .....	57
Figura 4.8 - Diagrama de Case para o grupo <i>tcp</i> .....	57
Figura 4.9 - Diagrama de Case para o grupo <i>udp</i> .....	57
Figura 4.10 - Grafo de dependências.....	63
Figura 4.11 - As camadas do protocolo TCP/IP e os fluxos de dados. ....	64

Figura 4.12 - Caminhos de propagação de anomalias no fluxo de entrada do grafo de dependências.....	66
Figura 4.13 - Caminhos de propagação de anomalias no fluxo de saída do grafo de dependências.....	67
Figura 4.14 - Caminhos de propagação de anomalias no fluxo de encaminhamento do grafo de dependências.....	68
Figura 4.15 - Possíveis classificações para alarmes. ....	72
Figura 4.16 - Passos do algoritmo de configuração automática de parâmetros. ....	74
Figura 4.17 - Ambiente de rede monitorado na UEL. ....	76
Figura 4.18 - Grafo que modela a topologia da rede monitorada.....	77
Figura 5.1 - Elementos de rede monitorados para o desenvolvimento e testes do sistema de detecção de anomalias. .....	81
Figura 5.2 - Diagrama de componentes da ferramenta GBA versão 6.0.....	82
Figura 5.3 - Diagrama de componentes da ferramenta GBA incluindo a solução de detecção de anomalias. ....	84
Figura 5.4 - Grafo de dependências utilizado nos testes. ....	86
Figura 5.5 - Objetos monitorados dentro do contexto do ambiente de rede da UEL. ....	87
Figura 5.6 - Objetos monitorados dentro do contexto do ambiente de rede da UEL – remoção de objetos com problemas.....	88
Figura 5.7 - Grafo da topologia de rede aplicado nos testes.....	88
Figura 5.8 - Curva ROC para o servidor Proxy, anomalias do grupo A. ....	90
Figura 5.9 - Curva ROC para o servidor Web, anomalias do grupo A. ....	91
Figura 5.10 - Curva ROC para o Firewall, anomalias do grupo A.....	92
Figura 5.11 - Curva ROC para o servidor Proxy, anomalias do grupo B.....	93
Figura 5.12 - Curva ROC para o servidor Web, anomalias do grupo B.....	93

Figura 5.13 - Curva ROC para o Firewall, anomalias do grupo B.....	94
Figura 5.14 - Comparação entre desempenho do Módulo de Detecção de Anomalias para os grupos A e B de anomalias no Servidor Proxy.....	95
Figura 5.15 - Comparação entre desempenho do Módulo de Detecção de Anomalias para os grupos A e B de anomalias no Servidor Web.....	95
Figura 5.16 - Comparação entre desempenho do Módulo de Detecção de Anomalias para os grupos A e B de anomalias no Firewall.....	96
Figura 5.17 - Taxas de detecção para anomalias do grupo A.....	98
Figura 5.18 - Taxas de falsos positivos para anomalias do grupo A.....	99
Figura 5.19 - Taxas de detecção para anomalias do grupo B.....	99
Figura 5.20 - Taxas de falsos positivos para anomalias do grupo B. ....	100
Figura 5.21 - Visão da rede que mostra a propagação da primeira anomalia apresentada.	103
Figura 5.22 - Alarmes de primeiro nível nos objetos <i>ipOutRequests</i> e <i>tcpOutSegs</i> do servidor Web. ....	103
Figura 5.23 - Alarmes de primeiro nível no objeto <i>ifInOctets</i> , porta 3016, no switch BD.	104
Figura 5.24 - Alarmes de primeiro nível no objeto <i>ifOutOctets</i> , porta 3001, no switch BD. ....	104
Figura 5.25 - Alarmes de primeiro nível nos objetos <i>ipInReceives</i> e <i>ipForwDatagrams</i> no Firewall.....	105
Figura 5.26 - Alarmes de primeiro nível no objeto <i>ifInOctets</i> , porta 1, switch Transit.....	106
Figura 5.27 - Cenário de propagação do primeiro caso de anomalia. ....	107
Figura 5.28 - Visão da rede que mostra a propagação da segunda anomalia apresentada.	108
Figura 5.29 - Alarmes de primeiro nível no objeto <i>ifOutOctets</i> , porta 1, switch Transit...110	110
Figura 5.30 - Alarmes de primeiro nível nos objetos <i>ipInReceives</i> e <i>ipForwDatagrams</i> do Firewall.....	110

Figura 5.31 - Alarmes de primeiro nível no objeto <i>ifInOctets</i> , porta 3001, switch BD. ....	111
Figura 5.32 - Alarmes de primeiro nível no objeto <i>ifOutOctets</i> , porta 3011, switch BD...	111
Figura 5.33 - Alarmes de primeiro nível nos objetos <i>ipInReceives</i> , <i>ipInDelivers</i> e <i>tcpInSegs</i> , servidor Proxy .....	112
Figura 5.34 - Alarmes de primeiro nível nos objetos <i>ipOutRequests</i> e <i>tcpOutSegs</i> , servidor Proxy.....	113
Figura 5.35 - Alarmes de primeiro nível no objeto <i>ifInOctets</i> , porta 3011, switch BD. ....	113
Figura 5.36 - Alarmes de primeiro nível no objeto <i>ifOutOctets</i> , porta 4011, switch BD...	114
Figura 5.37 - Alarmes de primeiro nível no objeto <i>ifInOctets</i> , porta 50, switch do Departamento de Computação.....	114
Figura 5.38 - Cenário de propagação do segundo caso de anomalia.....	115

# **Lista de Tabelas**

Tabela 4.1 - Algoritmo utilizado para definir se um desvio de comportamento deve ser considerado como anomalia. ....	41
Tabela 4.2 - Alarmes gerados para o objeto <i>ipInReceives</i> , servidor Proxy da UEL de 29/03/2009 a 04/04/2009. ....	51
Tabela 4.3 - Algoritmo para tradução dos diagramas de Case em grafos de dependências.	59
Tabela 4.4 - Algoritmo de busca em profundidade para o Módulo de Correlação. ....	70
Tabela 5.1 - Períodos de treinamento e respectivas semanas analisadas.....	97
Tabela 5.2 - Parâmetros escolhidos para o grupo A de anomalias. ....	100
Tabela 5.3 - Parâmetros escolhidos para o grupo B de anomalias. ....	101

## Glossário

ACM	<i>Association for Computing Machinery</i>
AIR	<i>Active Integrated Fault Reasoning</i>
ALAD	<i>Application Layer Anomaly Detector</i>
ARIMA	<i>Autoregressive Integrated Moving Average</i>
ATI	Assessoria de Tecnologia de Informação da UEL
BGP	<i>Border Gateway Protocol</i>
BLGBA	<i>Baseline para Gerenciamento de Backbone Automático</i>
CSI-KNN	<i>Combined Strangeness and Isolation measure K-Nearest Neighbors</i>
CUSUM	<i>Cumulative Sum</i>
DC	Departamento de Computação da UEL
DDoS	<i>Distributed Denial of Service</i>
DNS	<i>Domain Name System</i>
DoS	<i>Denial of Service</i>
DSNS	<i>Digital Signature of Network Segment</i>
EJB	<i>Enterprise Java Beans</i>
EWMA	<i>Exponentially Weighted Moving Average</i>
FTP	<i>File Transfer Protocol</i>
GBA	Gerenciamento de Backbone Automático
GLR	<i>Generalized Likelihood Ratio</i>
HTTP	<i>Hypertext Transfer Protocol</i>
ICMP	<i>Internet Control Message Protocol</i>
IEEE	<i>Institute of Electrical and Electronics Engineers</i>
IETF	<i>Internet Engineering Task Force</i>
IMAPIT	<i>Integrated Measurement Analysis Platform for Internet Traffic</i>
IP	<i>Internet Protocol</i>
IPFIX	<i>IP Flow Information eXport</i>

MIB	<i>Management Information Base</i>
P2P	<i>Peer to Peer</i>
PCA	<i>Principal Component Analysis</i>
PDU	<i>Protocol Data Unit</i>
PHAD	<i>Packet Header Anomaly Detector</i>
PSO-KM	<i>K Means based on Particle Swarm Optimization</i>
PTF	<i>Passive TCP/IP Fingerprinting</i>
RFC	<i>Request for Comments</i>
RMON	<i>Remote Network Monitoring</i>
RTP	<i>Real-time Transport Protocol</i>
SMTP	<i>Simple Mail Transfer Protocol</i>
SNMP	<i>Simple Network Management Protocol</i>
SOFM	<i>Self-Organized Feature Map</i>
SVM	<i>Support Vector Machine</i>
TCP	<i>Transmission Control Protocol</i>
TCP/IP	<i>Transmission Control Protocol/Internet Protocol</i>
TTL	<i>Time to Live</i>
UDP	<i>User Datagram Protocol</i>
UEL	Universidade Estadual de Londrina
UNICAMP	Universidade Estadual de Campinas
VoIP	<i>Voice over IP</i>

## Lista de Símbolos

<b>M</b>	Matriz que contém dados do histórico analisado pelo BLGBA para geração do DSNS
<i>I</i>	Número de linhas da matriz <b>M</b>
<i>N</i>	Número de colunas da matriz <b>M</b>
$m_{i,j}$	Elemento da matriz <b>M</b>
$gr_i$	Maior elemento da linha <i>i</i> da matriz <b>M</b>
$sm_i$	Menor elemento da linha <i>i</i> da matriz <b>M</b>
$amp_i$	Diferença entre o maior ( $gr_i$ ) e o menor ( $sm_i$ ) elemento da linha <i>i</i> da matriz <b>M</b>
$Lim_k$	Limite da <i>k</i> -ésima classe
$Bl_i$	Valor contido no DSNS para o instante de amostragem <i>i</i>
$\alpha$	Fator de tolerância no algoritmo de Definição Parametrizada de Anomalia
$\gamma$	Número de violações consecutivas toleradas no algoritmo de Definição Parametrizada de Anomalia
$\delta$	Número de violações toleradas dentro do intervalo de histerese no Módulo de Análise de Objeto SNMP
$G$	Grafo, $G=(V,A)$
$V$	Conjunto de vértices de um grafo
$A$	Conjunto de arestas de um grafo
$O_i$	Conjunto de objetos definidos como iniciais
$O_f$	Conjunto de objetos definidos como finais
$O_a$	Conjunto de objetos que apresentaram alarmes de primeiro nível
<i>Pilha</i>	Pilha utilizada na busca em profundidade
$C(o)$	Função que retorna todos os objetos correlacionados ao objeto <i>o</i>

$w_n$	Semana a ser monitorada com configuração escolhida pelo Módulo de Configuração Automática
$w_I$ a $w_{n-1}$	Semanas de treinamento do Módulo de Configuração de Automática
$T$	Taxa de detecção
$F$	Taxa de falsos positivos
$E$	Eficiência
$H$	Conjunto de valores de intervalo de histerese utilizados no treinamento do Módulo de Configuração Automática
$h_i$	Elemento do conjunto $H$
$D$	Conjunto de valores de $\delta$ utilizados no treinamento do Módulo de Configuração Automática
$d_i$	Elemento do conjunto $D$
$P$	Conjunto de valores para parâmetros formados pela combinação dos conjuntos $H$ e $D$
$p_i$	Elemento do conjunto $P$
$S$	Conjunto de alarmes de segundo nível
$s_i$	Elemento do conjunto $S$
$L$	Conjunto de anomalias inseridas pelo administrador de redes
$l_i$	Elemento do conjunto $L$
$J_{i,j}$	Conjunto de pares que combinam alarmes contidos no conjunto $S$
$el_i$	Elemento de rede monitorado

## Trabalhos Publicados Pelo Autor

1. ZARPELÃO, Bruno Bogaz; MENDES, Leonardo de Souza; PROENÇA JR., Mario Lemes; RODRIGUES, Joel José Puga Coelho; **Three Levels Network Analysis for Anomaly Detection**, Proceedings of 2009 International Conference on Software, Telecommunications and Computer Networks (SoftCOM 2009), 2009.
2. ZARPELÃO, Bruno Bogaz; MENDES, Leonardo de Souza; PROENÇA JR., Mario Lemes; RODRIGUES, Joel José Puga Coelho; **Parameterized Anomaly Detection System with Automatic Configuration**, Proceedings of IEEE Global Telecommunications Conference (IEEE GLOBECOM 2009), Communications Software and Services Symposium (GC'09 CSS), 2009.
3. ZARPELÃO, Bruno Bogaz; MENDES, Leonardo de Souza; PROENÇA JR., Mario Lemes. **Anomaly Detection Using DSNS and a Dependency Graph for SNMP Objects**, Proceedings of Fourth Advanced International Conference on Telecommunications (AICT 2008), p. 56-63, 2008.
4. ZARPELÃO, Bruno Bogaz; MENDES, Leonardo de Souza; PROENÇA JR., Mario Lemes. **Dependency Graph to Improve Notifications Semantic on Anomaly Detection**, Proceedings of 11th IEEE IFIP Network Operations and Management Symposium, p. 726-729, 2008.
5. PROENÇA JR., Mario Lemes; ZARPELÃO, Bruno Bogaz; MENDES, Leonardo de Souza. **Anomaly Detection Using Digital Signature of Network Segment Aiming to Help Network Management**, Journal of Communication and Information Systems, v. 23, n.1, p. 1-11, 2008.
6. ZARPELÃO, Bruno Bogaz; MENDES, Leonardo de Souza; PROENÇA JR., Mario Lemes. **Anomaly Detection Aiming Pro-Active Management of Computer Network Based on Digital Signature of Network Segment**, Journal of Network and Systems Management, v. 15 n. 2, p. 267-283, jun 2007.
7. ZARPELÃO, Bruno Bogaz; MENDES, Leonardo de Souza; PROENÇA JR., Mario Lemes. **Graph-based Correlation of SNMP Objects for Anomaly Detection**,

International Journal of Computer Science and Network Security, v. 6, n. 5b, p. 194-202, 2006.

8. ZARPELÃO, Bruno Bogaz; MENDES, Leonardo de Souza; BOTTOLI, Maurício; BREDA, Gean Davis; PROENÇA JR., Mario Lemes. **Correlação de Objetos SNMP na Detecção de Anomalias em Servidores de Rede**, Anais do XXII Simpósio Brasileiro de Telecomunicações (SBrT'05), p. 1007-1012, 2005.
9. ZARPELÃO, Bruno Bogaz; MENDES, Leonardo de Souza; PROENÇA JR., Mario Lemes. **Anomaly Detection Aiming Pro-Active Management of Computer Network Based on Digital Signature of Network Segment**, Proceedings of 4<sup>th</sup> Latin American Network Operations and Management Symposium, p. 53-64, 2005.
10. PROENÇA JR., Mario Lemes; ZARPELÃO, Bruno Bogaz; MENDES, Leonardo de Souza. **Anomaly Detection for Network Servers using Digital Signature of Network Segment**. Proceedings of Advanced Industrial Conference on Telecommunications 2005, p. 290-295, 2005.

# 1 Introdução

As redes de computadores se tornaram parte integrante do nosso cotidiano, permitindo a transmissão e processamento de diferentes tipos de informações que incluem voz, vídeos e dados. Através delas, são oferecidos serviços como voz sobre IP (VoIP), vídeo sob demanda, mensagens instantâneas, sistemas de compartilhamento de arquivos, acesso remoto e serviços diversos. Para atender a crescente demanda por novos serviços com qualidade, as redes se tornaram sistemas complexos, compostos por elementos heterogêneos que operam em alta velocidade e interagem constantemente entre si. O crescimento das redes tornou suas gerências por operadores humanos uma tarefa difícil, criando a necessidade da automatização das funções administrativas (HAJJI, 2005) (PATCHA; PARK, 2007) (PROENÇA JUNIOR, 2005).

A complexidade e a utilização em larga escala das redes de computadores dificultam o monitoramento e o controle, tornando comum a ocorrência de anomalias. As anomalias de rede são definidas como situações onde os níveis de tráfego apresentam um desvio súbito e acentuado de seu comportamento normal. Elas podem surgir de diferentes situações como falhas em elementos da rede, defeitos em softwares, configurações erradas e ataques de negação de serviço (*Denial of Service, DoS*), tornando muito difícil a tarefa de desenvolver técnicas para detectá-las.

Mesmo quando não afetam profundamente o funcionamento da rede, as anomalias acabam tendo impacto na qualidade dos serviços oferecidos aos usuários finais (LAKHINA et al., 2004). Assim sendo, a aplicação de uma abordagem pró-ativa de gerência que inclua a detecção antecipada de anomalias se tornou a chave para garantir confiabilidade, conectividade e segurança às redes. Ao possibilitar a rápida solução dos problemas, evita-se que os serviços oferecidos sejam prejudicados e que haja desperdício de recursos e perdas econômicas.

As técnicas para detecção de anomalias são classificadas, de maneira geral, em duas áreas: detecção baseada na caracterização do comportamento normal da rede e detecção

baseada nas assinaturas das anomalias. Na detecção baseada nas assinaturas, os sistemas possuem uma base de dados com a descrição das características das anomalias que podem ser detectadas. Os indicadores de funcionamento da rede são constantemente monitorados e quando uma situação semelhante a um registro desta base é encontrada, um alarme é disparado. Estes sistemas apresentam poucos alarmes falsos, mas não têm a capacidade de detectar anomalias novas e desconhecidas, comuns atualmente devido ao constante crescimento das redes e surgimento de novas tecnologias. (THOTTAN; JI, 2003) (PATCHA; PARK, 2007) (KIND et al., 2009)

As técnicas de detecção de anomalias baseadas na caracterização do comportamento normal da rede, por outro lado, não necessitam de um prévio conhecimento sobre as características das anomalias a serem detectadas. As principais vantagens destes sistemas são a capacidade em detectar anomalias desconhecidas e a facilidade em se adaptar a novos ambientes. Estes sistemas estabelecem um perfil de comportamento normal da rede estudando seu histórico de dados. A detecção é alcançada a partir da busca por mudanças significativas nos indicadores de funcionamento da rede que não são coerentes com as estimativas definidas no perfil de comportamento normal (SHON; MOON, 2007) (ESTEVEZ-TAPIADOR et al., 2004) (PROENÇA JUNIOR 2005).

Mesmo com estas vantagens, a detecção de anomalias baseada na caracterização de tráfego ainda não é largamente aplicada. A primeira dificuldade encontrada é o fato de haver vários modelos de caracterização do tráfego de rede (HAJJI, 2005). O comportamento do tráfego de rede é não estacionário e profundamente influenciado pelo padrão de utilização ditado pelos usuários, que pode mudar conforme o horário do dia ou dia da semana. Redes corporativas, por exemplo, costumam apresentar níveis de tráfego mais elevados no horário comercial e nos dias úteis.

Outra deficiência da detecção baseada na caracterização do tráfego está na dificuldade em definir quais variações de tráfego devem ser consideradas anomalias. O tráfego de rede apresenta variações em seu comportamento o tempo todo. Uma variação pode ser classificada de maneiras diferentes de acordo com a política de gerência empregada. Para um dado administrador de redes, mesmo os menores desvios de comportamento devem ser identificados, de forma a detectar todas as possibilidades de utilização indesejada dos recursos da rede. Um segundo administrador de rede pode estar

interessado apenas em desvios mais intensos, que remetem a problemas mais graves. Na literatura, abordagens distintas são utilizadas para definir quais desvios de comportamento da rede são anomalias. No trabalho de Thottan e Ji (2003), são considerados como anomalias somente os desvios que causam interrupções nas operações da rede. Lakhina et al. (2004), Roughan et al. (2004) e Estevez-Tapiador et al. (2004) mostraram alguns eventos que não foram reportados nos registros de eventos da rede e não causaram interrupções, mas refletiram na qualidade do serviço oferecido aos usuários finais e, por isso, deveriam ter sido detectados.

Por fim, há dificuldade de implantar as soluções propostas em ambientes de produção reais. Muitas das propostas apresentadas não seguem os padrões de gerência de redes, já que são poucas as propostas que fazem uso apenas do protocolo SNMP (*Simple Network Management Protocol*) (RFC 1157, 1990). Muitos trabalhos coletam dados utilizando protocolos como o NetFlow (RFC 3954, 2004), o qual oferece recursos sofisticados, mas não é encontrado em equipamentos de rede da grande maioria dos fabricantes, como é o caso do SNMP. Além disso, mesmo soluções que trazem bons resultados são difíceis de implantar, pois não há ferramentas que facilitem a configuração e adaptação destes sistemas às diferentes políticas de gerência. Um exemplo desta situação é a utilização da técnica PCA (*Principal Component Analysis*) na detecção de anomalias, inicialmente proposta por Lakhina et al. (2004). Os bons resultados obtidos neste trabalho fizeram com que a aplicação da PCA se tornasse uma das abordagens mais conhecidas para detecção de anomalias. Porém, o trabalho de Ringberg et al. (2007), ao tentar estender a utilização da PCA, concluiu que pequenas modificações na configuração da PCA causam grandes alterações na taxas de falsos positivos obtidas, dificultando muito a utilização desta técnica.

Esta tese propõe um sistema de detecção de anomalias em redes de computadores baseado na caracterização do comportamento normal de tráfego. O objetivo é oferecer uma abordagem prática para a detecção de anomalias, disponibilizando facilidades para a configuração do sistema e utilizando padrões de gerência de redes que permitirão a adoção da solução em diferentes redes. O sistema é capaz de identificar a ocorrência da anomalia e mostrar uma visão panorâmica do seu comportamento em toda a rede para o administrador.

O primeiro passo a ser dado na detecção de anomalias é a coleta das informações da rede para identificação dos problemas. Em nosso trabalho, utilizamos apenas informações coletadas dos objetos de gerência presentes na MIB-II (*Management Information Base*) (RFC 1213, 1991) através do protocolo SNMP. Tanto o SNMP quanto a MIB-II são padrões de gerência para redes IP definidos pela IETF (*Internet Engineering Task Force*), sendo suportados pela grande maioria dos dispositivos de rede disponíveis no mercado.

Para definir o comportamento normal das informações coletadas nos objetos SNMP, utilizamos o modelo BLGBA (*Baseline para Gerenciamento de Backbone Automático*), proposto por Proença Junior (2005). A aplicação deste modelo é realizada para cada objeto SNMP monitorado e resulta em perfis de comportamento normal, denominados *baselines* ou DSNS (*Digital Signature of Network Segment*). Cada *baseline* ou DSNS é específico para um dia da semana e contém estimativas para cada instante do dia, traçando um perfil detalhado do comportamento normal de um objeto SNMP em um elemento de rede.

A análise das informações coletadas na rede é realizada em três diferentes níveis:

- **Primeiro nível de análise:** as informações coletadas para cada objeto são analisadas individualmente pelo Módulo de Análise de Objeto SNMP. Em cada um dos objetos SNMP, as informações obtidas em tempo real são comparadas com o respectivo baseline/DSNS. Esta comparação é realizada com a aplicação de um algoritmo determinístico, criado com base no mecanismo de alarmes previsto no protocolo RMON (*Remote Monitoring*) (RFC 1757, 1995). A configuração da sensibilidade deste algoritmo é automatizada, permitindo que o sistema se adapte a diferentes políticas de gerência. Os alarmes de primeiro nível são gerados quando desvios significativos do tráfego real em relação ao DSNS são detectados.
- **Segundo nível de análise:** os alarmes de primeiro nível gerados para os diferentes objetos monitorados são enviados para o Módulo de Correlação, que executa o segundo nível de análise. O Módulo de Correlação traz as relações entre os objetos SNMP monitorados no elemento de rede gerenciado. Os alarmes de primeiro nível são cruzados com base em um grafo de dependências que reúne as relações entre os objetos SNMP,

causando a geração de um alarme de segundo nível quando a anomalia é confirmada para o elemento de rede.

- **Terceiro nível de análise:** é executado no Módulo de Localização de Anomalias, que tem como função oferecer uma visão panorâmica do problema ao administrador de rede. Neste módulo, os alertas de segundo nível gerados para os diferentes elementos de rede monitorados são cruzados com base na topologia da rede analisada. As informações da topologia da rede são inseridas manualmente pelo administrador de redes no formato de um grafo de dependências entre os equipamentos monitorados. Desta forma, é possível mostrar ao administrador de rede como a anomalia está se propagando pela rede, facilitando a sua solução.

Esta proposta de detecção de anomalias que reúne a utilização do protocolo SNMP, o modelo de caracterização de tráfego BLGBA e a análise das informações coletadas na rede em três diferentes níveis traz as seguintes contribuições:

- **Possibilidade de configuração da sensibilidade do sistema de detecção:** um sistema que detecte todos os desvios de comportamento não atende a um administrador de rede que prefere uma gerência de redes mais permissiva, que vise detectar apenas os desvios de comportamento mais graves. Da mesma forma, um sistema de detecção que só alerte o administrador nos casos mais graves, pode não atender a uma política que pretenda detectar a grande maioria dos desvios. Por esta razão, é importante o desenvolvimento de um algoritmo que possa ter sua sensibilidade alterada e se adapte às diferentes políticas de gerência.
- **Criação de um grafo de dependências entre os objetos SNMP da MIB-II:** o monitoramento de diferentes objetos possibilita a gerência utilizando SNMP. Cada objeto proporciona a visão de uma determinada etapa do processamento do pacote dentro do elemento de rede. O cruzamento dos dados coletados nos objetos permite que seja mapeado o comportamento da anomalia dentro do elemento de rede analisado. O grafo de dependências proposto nesta tese visa proporcionar uma análise objetiva e eficaz dos indicadores obtidos para cada um dos objetos monitorados.

- **Visão geral da rede:** o método proposto cruza os mapas de comportamento das anomalias obtido em cada elemento de rede e as informações sobre a topologia da rede para oferecer ao administrador uma visão panorâmica do problema. Com este diagnóstico, o administrador poderá determinar como a anomalia está se propagando em sua rede, facilitando a solução do problema.
- **Desenvolvimento e testes em um ambiente real de rede:** a proposta é desenvolvida e aplicada sobre informações reais coletadas da rede da Universidade Estadual de Londrina (UEL). Este ambiente reúne as características necessárias para que a viabilidade da aplicação da solução em ambientes de produção seja avaliada.

Esta tese está organizada da seguinte forma: o capítulo 2 traz os diferentes conceitos que envolvem a área de detecção e localização de anomalias, além de diferentes propostas para detecção e localização de anomalias, apresentadas em publicações recentes coletadas em bases de organizações importantes como IEEE (*Institute of Electrical and Electronics Engineers*) e ACM (*Association for Computing Machinery*). O capítulo 3 apresenta os conceitos do modelo de caracterização de tráfego utilizado neste trabalho, conhecido como BLGBA. O capítulo 4 detalha a proposta de sistema de detecção de anomalias desta tese. O capítulo 5 mostra detalhes da implementação, o processo de avaliação e os resultados obtidos com a aplicação da proposta no ambiente real da UEL (Universidade Estadual de Londrina). Por fim, o capítulo 6 apresenta as considerações finais e os trabalhos futuros.

## 2 Principais definições

Este capítulo descreve os conceitos básicos relacionados às anomalias em redes de computadores. São apresentados causas e tipos de anomalias, fontes de coleta de informações sobre a rede e os diferentes métodos utilizados para detecção e localização de anomalias. A tomografia de redes, apesar de não ser o foco deste trabalho, também será apresentada por estar relacionada com o diagnóstico de anomalias. Por fim, é apresentado um levantamento de soluções para detecção e localização de anomalias propostas em trabalhos recentes.

### 2.1 **Anomalias e suas principais causas**

Anomalias são alterações súbitas, significativas e não esperadas nos níveis do tráfego de rede. Elas podem apresentar causas e consequências variadas, o que dificulta a identificação e solução dos problemas provocados por elas. Na gerência de redes, a garantia de confiabilidade e segurança passa necessariamente pelo diagnóstico eficaz destes problemas. Os estudos nesta área visam, principalmente, desenvolver técnicas que proporcionem a detecção rápida e eficiente das anomalias, permitindo ao administrador de rede isolar o problema e tentar solucioná-lo antes que as consequências sejam ainda mais profundas (FARRAPOSO et al., 2007).

As anomalias podem ser causadas por diferentes problemas, que levam à classificação delas em duas categorias. O primeiro grupo reúne as anomalias onde não há a presença de agentes maliciosos. O segundo traz as anomalias causadas por ataques, normalmente arquitetados por agentes que visam romper as barreiras de segurança da rede (THOTTAN; JI, 2003).

Como exemplos de causas de anomalias pertencentes ao primeiro grupo, temos:

- *Flash crowd*: uma grande quantidade de clientes passa, subitamente, de maneira não orquestrada e não maliciosa, a enviar requisições a um servidor, podendo fazer com que ele chegue a interromper as suas operações. Durante

os ataques terroristas de 11 de setembro de 2001 nos Estados Unidos, o portal de notícias da rede CNN sofreu esta anomalia. A razão foi o altíssimo número de usuários que começaram a buscar notícias simultaneamente no portal (JUNG et al., 2002).

- *Babbling node*: um nó da rede entra em estado de falha por tempo indefinido, enviando pacotes aleatoriamente para vários pontos da rede. Um exemplo de anomalia do tipo *babbling node* ocorre quando uma placa de rede com problemas envia uma grande quantidade de pacotes de controle ou sinalização para toda a rede (AL-KASASSBEH; ADDA, 2009).
- Tempestade de *broadcasts*: uma enorme quantidade de pacotes de *broadcast* começa a trafegar pela rede, causando congestionamentos que podem levar à interrupção das operações da rede. Falhas em dispositivos que iniciam o envio de pacotes de *broadcast* para outros ativos da rede, fazendo com que estes outros ativos também enviem *broadcasts* e formem uma reação em cadeia, causam tempestades de *broadcasts*. Como exemplo, há casos de softwares de controle de impressoras HP Laserjet 3, que após a remoção da impressora da estação, causavam tempestades de *broadcasts* na busca pela impressora em toda a rede (AL-KASASSBEH; ADDA, 2009).
- Congestionamentos: ocorrem com o aumento brusco de tráfego em um determinado ponto da rede, que causa dificuldade de processamento destes pacotes, os quais acabam sendo entregues com atraso ou são descartados. A queda de um enlace, por exemplo, dificulta o escoamento do tráfego que chega a determinados pontos da rede, causando congestionamentos. Ocorrências do tipo *flash crowd*, *babbling node* e tempestade de *broadcasts* também causam congestionamentos.
- *Bugs* nos softwares de roteamento: os softwares presentes em equipamentos de roteamento podem não estar preparados para lidarem com pacotes que chegam com determinados problemas, como más formações. Consequentemente, há um comportamento não previsto no roteamento dos pacotes, causando alteração nos níveis de tráfego medidos na rede (ROUGHAN et al., 2004);

- Erros em configurações: dispositivos como *firewalls*, se configurados incorretamente, podem barrar a entrada de uma grande quantidade de pacotes na rede, causando mudanças repentinhas nos níveis de tráfego monitorados. Erros na configuração de servidores podem levá-los a não responder adequadamente às requisições enviadas pelos clientes, causando congestionamentos (THOTTAN; JI, 2003).

No segundo grupo de anomalias, temos os seguintes exemplos:

- Ataque de negação de serviço (*denial of service, DoS*): é diferente de uma invasão, pois tenta tornar o serviço inoperante através da imposição de uma sobrecarga. Nestes ataques, o agente malicioso envia uma grande quantidade de requisições ao servidor de forma que todos os seus recursos de memória e processamento, por exemplo, sejam consumidos. Um exemplo de ataque *DoS* é o *SYN flood*, no qual o cliente inicia o estabelecimento de uma conexão TCP com o servidor e não conclui a sinalização necessária para o estabelecimento da conexão, fazendo com que ela fique aberta esperando pela conclusão do processo no servidor. Isto ocorre de maneira contínua até que os recursos do servidor sejam todos consumidos e este não possa mais atender a novas requisições. Diferencia-se da ocorrência de *flash crowds* pelo simples fato de ser fruto de ações mal intencionadas. No ataque distribuído de negação de serviço (*distributed denial of service, DDoS*), uma máquina mestre domina um conjunto de outras máquinas, que são denominadas zumbis, e as utiliza para praticar um ataque de negação de serviço (COLE et al., 2005).
- Vermes (*worms*): são programas que têm a capacidade de se espalhar e executar cópias em máquinas remotas ao longo da rede, infectando grandes conjuntos de alvos. Primeiramente, o verme é instalado em um *host*. Depois, ele identifica os potenciais alvos para a sua propagação e espalha suas cópias pela rede. Por fim, as várias instâncias realizam os ataques, como por exemplo, o envio de requisições simultâneas a um determinado servidor para causar a sua parada (ELLIS, 2003).

- Escaneamento de portas (*port scan*): apesar de ser uma técnica utilizada por administradores de rede para executar tarefas de gerência, pode também ser utilizada por agentes maliciosos para descobrir vulnerabilidades nas redes escolhidas como alvo. Quando o escaneamento de portas é realizado de forma maliciosa, pode causar congestionamentos e possibilitar futuros ataques ainda mais sérios através das vulnerabilidades descobertas (COLE et al., 2005).

## 2.2 ***Coleta de informações da rede***

O primeiro passo no processo que envolve a identificação e o tratamento de anomalias é a coleta de informações da rede, que trazem detalhes do seu funcionamento, sem os quais é impossível identificar se ela opera em um estado normal ou não. Características dos dados utilizados acabam por nortear decisões ao longo do restante da construção do sistema.

Há diferentes formas de coletar informações de uma rede, e a escolha por uma delas passa por questões como a quantidade de recursos disponíveis para realizar o monitoramento e o tipo de problemas que se deseja identificar. A eficiência na detecção de anomalias está relacionada à escolha de um conjunto apropriado de dados para análise, que retrate de maneira fiel as características do funcionamento da rede monitorada. Os tipos de anomalias que serão detectadas dependem das características dos dados utilizados na detecção (ESTEVEZ-TAPIADOR et al., 2004) (THOTTAN; JI, 2003). Serão apresentadas as seguintes fontes de informações, as quais são normalmente utilizadas na detecção e localização de anomalias: estatísticas obtidas através do protocolo SNMP, estatísticas de fluxos de pacotes, estatísticas coletadas através de sondas (*probes*), cabeçalhos dos pacotes e topologia das redes.

### 2.2.1 **Estatísticas obtidas com o SNMP**

O *Simple Network Management Protocol* (SNMP) (RFC 1157, 1990) é o protocolo padrão para gerência de redes TCP/IP definido pelo IETF (*Internet Engineering Task Force*). O SNMP é um padrão de gerência simples, que requer poucos recursos para ser implementado. Ele oferece um conjunto de operações que permitem o monitoramento de

diferentes dispositivos de rede como roteadores, servidores, switches, impressoras entre outros. As informações que podem ser coletadas utilizando o protocolo SNMP variam de estatísticas como a quantidade de bytes que ingressaram uma determinada interface a indicadores como a temperatura interna de um equipamento. O SNMP é definido como um protocolo da camada de aplicação e utiliza o protocolo de transporte UDP para efetuar a troca de dados entre agentes e gerentes.

As ações do protocolo SNMP envolvem basicamente três componentes: o dispositivo gerenciado, o agente e o gerente. O dispositivo gerenciado é um nó da rede que contém um agente SNMP, responsável por reunir as informações sobre as operações do nó e disponibilizá-las ao gerente, no formato previsto pelo protocolo SNMP. O gerente, por sua vez, é responsável por se comunicar com os agentes e buscar estas informações nos nós gerenciados. Um único gerente pode se comunicar com vários agentes simultaneamente.

Na Figura 2.1, são apresentadas algumas das interações entre o gerente e um agente, previstas no protocolo SNMP. A interação *GET request* é utilizada pelo gerente para requisitar uma determinada informação ao agente que, por sua vez, utiliza a interação *GET response* para retornar esta informação ao gerente. Os *TRAPs* são espécies de alarmes que os agentes enviam ao gerente, reportando alguma ocorrência especial e também fazem parte das operações de monitoramento. Entre as situações que podem ser reportadas, temos reinicialização do elemento de rede gerenciado, queda de enlace, (re)estabelecimento do enlace, entre outras. A interação *SET request* é uma operação de controle que possibilita ao gerente alterar valores de objetos de gerência no agente. Com a operação *SET*, o gerente pode, por exemplo, desabilitar uma interface de um switch.

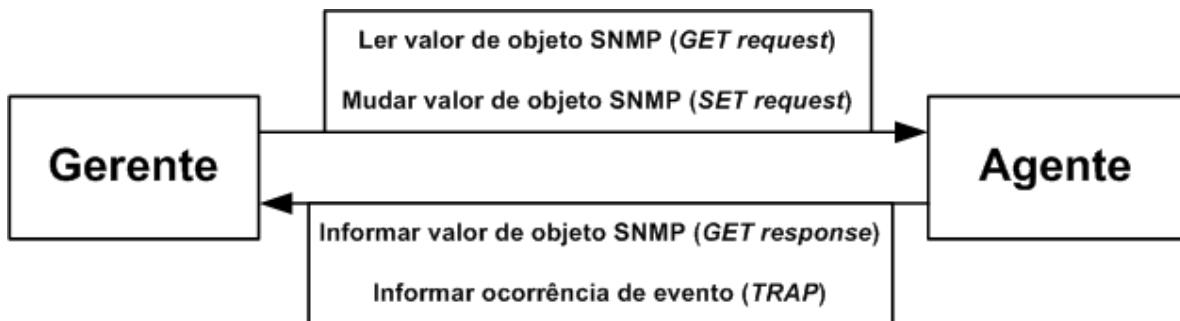


Figura 2.1 - Interações previstas no protocolo SNMP entre o gerente e o agente.

As informações que podem ser obtidas através do protocolo SNMP ficam armazenadas em bases de dados no dispositivo gerenciado, denominadas MIB (*Management Information Base*). Elas contêm variáveis ou objetos de gerência, responsáveis por armazenar as estatísticas de funcionamento do equipamento. O objeto de gerência *ipInReceives*, por exemplo, é responsável por armazenar a informação de quantos pacotes IP foram recebidos pelo elemento de rede analisado. As informações na MIB são organizadas de maneira hierárquica. Para requisitar as informações contidas em um determinado objeto, a requisição enviada pelo gerente deve conter o identificador deste objeto, que representa o caminho que deve ser percorrido na hierarquia de informações da MIB, até que o objeto desejado seja atingido. A Figura 2.2 mostra a árvore de informações de uma MIB. Para requisitar, por exemplo, o objeto *ipInReceives*, que é o terceiro objeto do grupo *ip*, é necessário passar o identificador de objeto 1.3.6.1.2.1.4.3.

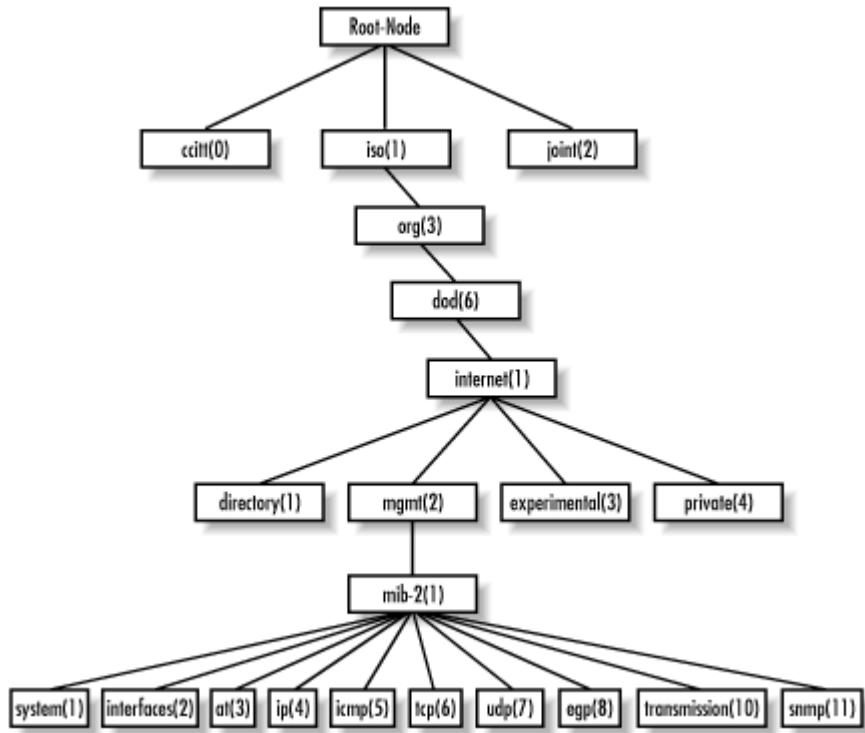


Figura 2.2 - Informações da MIB dispostas de forma hierárquica (MAURO; SCHMIDT, 2001).

## 2.2.2 Fluxos de pacotes

No monitoramento de fluxos de pacotes, cada interface de um roteador é monitorada e os pacotes coletados são agrupados utilizando os seguintes dados presente no cabeçalho do pacote:

- Endereços IP de origem e destino;
- Campo protocolo do cabeçalho IP;
- Portas de origem e destino;
- Campo de tipo de serviço do cabeçalho do protocolo IP;

O roteador coleta estes dados, mantendo-os em sua memória. Em intervalos determinados de tempo, o conjunto de dados armazenado na memória é exportado para um coletor, que faz o armazenamento definitivo dos registros dos fluxos IP. Além das informações já citadas, os registros de fluxos contêm também os *timestamps* do início e término do fluxo, as *flags* TCP observadas no monitoramento, a quantidade de *bytes* e a quantidade de pacotes que trafegavam neste fluxo. Com estes dados, é possível calcular, por exemplo, qual foi a taxa de transmissão em *bits/s* entre duas aplicações de dois nós da rede no fluxo monitorado. É possível também levantar os serviços utilizados em cada fluxo através das portas de origem e destino e quantas conexões TCP foram estabelecidas. O cruzamento das informações de fluxos monitorados em diferentes roteadores pode prover ao administrador de rede uma visão panorâmica de todo o sistema.

Este processo de coleta de dados de fluxo está implementado no protocolo NetFlow, desenvolvido pela Cisco Systems (RFC 3954, 2004). A versão 9 deste protocolo foi utilizada pelo grupo de trabalho IPFIX (*IP Flow Information eXport*) do IETF, visando a especificação do modelo de informação para descrição de fluxos IP e do protocolo IPFIX, que define como deve ser feita a transferência de dados entre os exportadores e os coletores de dados.

Em situações de tráfego intenso, a tarefa de manter todos os dados dos fluxos ocupa grande parte dos recursos de armazenamento e processamento dos roteadores. Para evitar este problema, o NetFlow prevê a configuração de uma taxa de amostragem, na qual pode ser definida a quantidade de pacotes que devem ser coletados em relação ao total que realmente trafegou pela interface observada. Porém, a definição correta desta taxa de

amostragem ainda é um desafio. Uma taxa de amostragem baixa pode ser interessante em situações de sobrecarga do roteador, pois preserva os recursos. Contudo, em situações de normalidade, a taxa de amostragem baixa subutiliza os recursos disponíveis e coleta menos dados do que realmente poderia ser coletado, diminuindo a eficácia da análise (ESTAN et al., 2004).

Outra dificuldade neste método de medição é definir o momento em que um fluxo foi finalizado. O protocolo NetFlow utiliza quatro regras, visando atender este requisito:

1. Um fluxo é considerado finalizado quando há o fim de uma conexão TCP identificado através de *flags* como FIN ou RST;
2. Um fluxo é considerado finalizado quando não é observado um pacote pertencente a ele a mais de 15 segundos. É importante mencionar que este valor é configurável;
3. Um fluxo é considerado finalizado 30 minutos depois de ter sido criado;
4. Caso não haja mais memória suficiente para novos fluxos, todos os fluxos armazenados na memória do roteador são considerados finalizados e são exportados.

A aplicação destas regras unidas à amostragem de pacotes pode não ser eficiente, causando erros nas estatísticas levantadas sobre os fluxos monitorados. Durante a amostragem de pacotes, por exemplo, os pacotes de um fluxo podem não ser coletados mesmo que ele esteja ativo, fazendo com que a regra 2 seja aplicada. Na próxima vez que um pacote deste fluxo for incluído na amostragem, o fluxo será contabilizado novamente de maneira equivocada, já que o sistema o reconhecerá como um novo fluxo.

### **2.2.3 Estatísticas coletadas por sondas (*network probes*)**

As sondas (*network probes*) são ferramentas de medição ativa do estado das operações da rede. Estas ferramentas enviam pacotes especiais entre dois pontos da rede que permitem a coleta de métricas fim-a-fim como atraso, variação do atraso (*jitter*), capacidade de transmissão dos enlaces, largura de banda disponível e taxas de perdas.

A principal ferramenta de sonda é o *ping*, que envia pacotes ICMP ao destino e permanece esperando o seu retorno para coletar métricas como taxa de perdas e atraso.

Outra ferramenta bastante usada é o *traceroute*, que envia pacotes UDP e espera pacotes de resposta dos *gateways*, de forma que possa montar o caminho percorrido entre dois pontos e calcular os atrasos desta rota.

Ferramentas como o *pathchar* (PATHCHAR, 2010) permitem a coleta de uma variedade maior de métricas do que as oferecidas pelo *ping* e o *traceroute*. Assim como o *traceroute*, o *pathchar* envia pacotes UDP e coleta, trabalhando o campo TTL (*time to live*) dos pacotes, respostas em cada *hop*. Porém, diferente do *traceroute*, o *pathchar* combina as medições coletadas nos *hops* anteriores e no *hop* atual para inferir métricas como atraso, vazão, capacidade de transmissão e taxas de perda em cada ponto percorrido durante o trajeto.

A principal vantagem das sondas é que elas possibilitam a realização de medições em caminhos que atravessam diferentes domínios de gerência, atendendo a uma necessidade recorrente no monitoramento de desempenho dentro de ambientes heterogêneos como a Internet. Por outro lado, os diagnósticos oferecidos por essas ferramentas podem ser superficiais, já que os percursos realizados pelos pacotes na rede podem não refletir a realidade, distorcendo os resultados das medições e inferências realizadas sobre estas.

#### **2.2.4 Cabeçalhos de pacotes**

Os cabeçalhos dos pacotes que trafegam pela rede também podem ser utilizados para a detecção de anomalias. Em geral, os pacotes são coletados e analisados a partir do monitoramento de dispositivos de rede que operam em modo promíscuo (COLE et al., 2005). A inspeção dos cabeçalhos dos pacotes pode oferecer informações úteis, como o comportamento e a utilização de cada protocolo na rede. Diferentes sistemas de detecção de anomalias podem focar na análise de protocolos pertencentes às diferentes camadas. Logo, há sistemas que analisam somente os protocolos da camada de enlace, como o Token Ring e Ethernet, da camada de rede, como o IP, da camada de transporte, como o TCP, o UDP e o RTP e da camada de aplicação como o HTTP, o SMTP e o DNS (KRÜGEL et al., 2002) (MAHONEY; CHAN, 2002).

Utilizada principalmente em sistemas que têm como foco principal a segurança, a inspeção dos cabeçalhos dos pacotes pode, por exemplo, revelar más formações que

indicam possíveis anomalias, como pacotes com TTL pequeno, pacotes contendo endereços IP privados, pacotes com *flags* TCP inválidas e pacotes excessivamente pequenos. Termos encontrados no *payload* dos pacotes também podem ser úteis para evidenciar possíveis intrusões na rede monitorada.

### 2.2.5 Topologia de redes

Utilizada com menos frequência que as outras fontes de dados citadas, a topologia da rede monitorada também pode ser útil na detecção de anomalias. Uma de suas principais aplicações está relacionada à localização da origem das anomalias, onde a correlação dos alertas gerados por vários elementos da rede utiliza a topologia da rede como um de seus parâmetros de entrada para tomadas de decisão. A topologia da rede permite que sejam levantadas as dependências entre os equipamentos monitorados. A partir destas dependências, os métodos de localização de anomalias podem verificar qual foi o caminho de propagação da anomalia pela rede.

Com o aumento da utilização das redes sem fio, principalmente aquelas baseadas na ausência de controle centralizado como as redes *ad hoc*, as informações sobre a topologia das redes se tornaram importantes para as atividades de gerência. A composição destas redes é modificada frequentemente e o acompanhamento destas variações pode ser um elemento chave na detecção de comportamentos inadequados (ESTEVEZ-TAPIADOR et al., 2004).

## 2.3 Detecção de anomalias

Os primeiros sistemas de detecção de anomalias propostos utilizavam o método baseado nas assinaturas das anomalias. Este método é bastante eficaz e apresenta reduzidas taxas de alarmes falsos. Porém, as redes evoluíram rapidamente e a incapacidade dos sistemas baseados em assinaturas de se adaptarem a novos cenários e tipos de anomalias que surgiam com frequência fez com que os pesquisadores passassem a buscar outra solução.

Em 1987, Denning (1987) apresentou um trabalho pioneiro que propunha a detecção de anomalias utilizando não mais as características das anomalias a serem identificadas, e sim a caracterização do comportamento normal do tráfego. O objetivo do método baseado

na caracterização do comportamento normal era tornar os sistemas independentes das características das anomalias que poderiam ser detectadas.

Ainda hoje, os sistemas de detecção são classificados nestas duas categorias. Esta seção apresentará a descrição, as vantagens e as desvantagens destas duas categorias de sistemas de detecção de anomalias, detalhando algumas soluções utilizadas na detecção baseada no comportamento normal da rede, que é o foco neste trabalho. As soluções que serão detalhadas se baseiam em técnicas de aprendizado, processamento de sinais, especificação e mineração de dados.

### **2.3.1 Detecção baseada nas assinaturas das anomalias**

Na detecção baseada em assinaturas, as anomalias são modeladas de forma que suas principais características sejam levantadas e seja construída uma assinatura, que será armazenada em uma base de informações do sistema. Durante o monitoramento do tráfego de rede, o sistema busca comportamentos que tenham os mesmos atributos das assinaturas contidas na base de informações. Caso sejam encontradas situações semelhantes às descritas nas assinaturas, alarmes são enviados ao administrador de rede. Este método é bastante usado em sistemas de detecção de intrusão.

A principal vantagem deste método é que ataques presentes na base de informações podem ser detectados de maneira eficaz, com baixas taxas de falsos positivos. A assinatura reúne uma série de eventos específicos para definir um ataque, permitindo que o problema seja facilmente diagnosticado. Outra importante vantagem reside no fato de que este tipo de sistema começa a proteger a rede logo após a sua implantação, pois não há período de treinamento ou aprendizado sobre o funcionamento da rede.

A primeira desvantagem é que os sistemas de detecção baseados em assinaturas não têm a capacidade de detectar ataques com características desconhecidas da sua base. Além disso, sistemas que se baseiam neste método consomem muitos recursos da equipe de administração de redes, já que a base que contém as assinaturas precisa ser constantemente atualizada com os novos ataques criados e novos elementos que foram inseridos no ambiente monitorado (PATCHA; PARK, 2007) (COLE et al., 2005).

### **2.3.2 Detecção baseada na caracterização do comportamento normal**

A detecção baseada na caracterização do comportamento normal compara as informações coletadas da rede às características das atividades consideradas normais. O perfil de normalidade é obtido após um estudo do comportamento prévio da rede. Uma situação é considerada anômala quando o seu grau de desvio em relação ao perfil de normalidade é significativo (COLE et al., 2005) (ESTEVEZ-TAPIADOR et al., 2004) (PATCHA; PARK, 2007) (PROENÇA JUNIOR, 2005) (THOTTAN; JI, 2003).

A principal vantagem deste método é a capacidade de detectar anomalias desconhecidas. Como a operação de detecção é baseada no conceito de normalidade e não na busca por comportamentos conhecidos e considerados anormais, um ataque com características desconhecidas possivelmente será detectado, simplesmente por fazer com que as atividades da rede se desviem de sua normalidade.

A primeira desvantagem é a necessidade de um período de treinamento para que os perfis de normalidade sejam construídos. Outra desvantagem é a alta porcentagem de falsos alarmes encontrados nestes sistemas. Muitas vezes as variações de comportamento são naturais nas redes e podem ser encaradas como desvios de comportamento por estes sistemas. A caracterização do tráfego é também uma questão de difícil solução. Normalmente, as estatísticas obtidas no monitoramento das redes apresentam características não estacionárias, que dificultam o estabelecimento de um padrão de operação. Uma modelagem de comportamento de baixa qualidade vai implicar em um sistema ineficiente. Por fim, um invasor pode treinar o sistema gradualmente, de forma que as atividades maliciosas sejam incorporadas ao perfil e consideradas normais.

As próximas subseções detalharão soluções utilizadas para criação dos sistemas baseados na caracterização do comportamento normal da rede. Teremos detalhes sobre as técnicas de aprendizado, processamento de sinais, especificação e mineração de dados.

#### **2.3.2.1 Detecção baseada em técnicas de aprendizado**

Esta solução aplica técnicas de aprendizado sobre o histórico de dados reais coletados da rede para gerar os perfis de operações normais do tráfego. Após a geração do perfil do comportamento normal do tráfego, são determinados os limites de tolerância dos

desvios em relação a este perfil. Estes limites são construídos, por exemplo, a partir da análise de grandes quantidades de diferentes níveis de desvio padrão do tráfego real em relação à estimativa gerada. Quando a movimentação da rede se desvia do comportamento definido no perfil de operações normais, ultrapassando os limites de tolerância, uma anomalia é detectada.

Dentre as principais técnicas de aprendizado utilizadas, temos os algoritmos estatísticos e os modelos de inteligência artificial como as redes neurais e bayesianas. Geralmente, os perfis de comportamento normal definidos são específicos para cada hora do dia e dia da semana, diferenciando dias úteis, fins de semana e feriados (JIANG; PAPAVASSILIOU, 2004) (LI; MANIKOPOULOS, 2003) (SOULE et al., 2005) (WU; ZHANG, 2003).

As principais vantagens desta solução são a capacidade de detectar anomalias desconhecidas e de se adaptar a mudanças no ambiente que está sendo monitorado. Entretanto, a grande quantidade de dados necessária para a geração de um novo perfil de operações normais pode representar um problema.

### **2.3.2.2 Detecção baseada em especificação**

Como na detecção baseada em técnicas de aprendizado, na detecção baseada em especificação o objetivo também é definir um perfil de operações normais para a rede. A diferença principal é que na detecção baseada em especificação, este perfil é construído por um ser humano. O modelo de operações normais é definido a partir de um conjunto de especificações manuais que traduzem o comportamento normal da rede.

Essas especificações são desenvolvidas com o auxílio de ferramentas apropriadas, como máquinas de estados finitos (SEKAR et al., 2002). A principal desvantagem desta abordagem é que o desenvolvimento destas especificações é lento e complexo. Mudanças na rede analisada podem acarretar mudanças nas especificações, tornando a capacidade de adaptação do sistema bastante reduzida. Entretanto, a taxa de falsos alarmes tende a ser menor, já que essas especificações podem lidar melhor com as situações que parecem anomalias, mas são apenas variações naturais do tráfego de rede (ESTEVEZ-TAPIADOR et al., 2004).

### **2.3.2.3 Detecção baseada em processamento de sinais**

A detecção baseada em processamento de sinais utiliza técnicas de processamento de sinais para detectar anomalias em séries temporais que reúnem estatísticas coletadas de fluxos de dados ou de objetos SNMP. A partir destas técnicas, os sinais considerados normais são separados dos sinais anômalos presentes nas séries analisadas. Algoritmos de *wavelet*, transformadas de Fourier e algoritmos estatísticos como ARIMA (*Autoregressive Integrated Moving Average*) são algumas das técnicas utilizadas para analisar de maneira *on line* os dados coletados da rede (BARFORD et al., 2002) (LAKHINA et al., 2004) (THOTTAN; JI, 2003) (WU; SHAO, 2005).

A detecção baseada em processamento de sinais é capaz de detectar anomalias desconhecidas e de se adaptar rapidamente às evoluções do ambiente monitorado. Ao contrário das soluções baseadas em aprendizado, não é necessário um período de treinamentos extenso que consuma um grande conjunto de dados coletados da rede.

### **2.3.2.4 Detecção baseada em mineração de dados**

As técnicas de mineração de dados são capazes de lidar com grandes quantidades de dados, buscando padrões consistentes entre estes dados e formando subconjuntos que respeitem determinadas regras de associação. As soluções que utilizam a mineração de dados têm um objetivo parecido com as soluções que utilizam processamento de sinais: separar em dois subconjuntos os dados considerados normais e os dados considerados anômalos durante o processamento das informações coletadas da rede (PATCHA; PARK, 2007).

Técnicas relacionadas à mineração de dados como PCA (*Principal Component Analysis*) (LAKHINA et al., 2004), PSO-KM (*K Means based on Particle Swarm Optimization*) (XIAO et al., 2006) e CSI-KNN (*Combined Strangeness and Isolation measure K-Nearest Neighbors*) (KUANG; ZULKERNINE, 2008) são utilizadas com bons resultados para identificar em grandes conjuntos de dados coletados da rede, quais são os dados anômalos e quais são os dados normais.

## 2.4 *Localização de anomalias*

A utilização de sistemas de detecção de anomalias que analisam individualmente os elementos da rede gera conjuntos muito grandes de alarmes, os quais são frequentemente redundantes. Em redes de comunicações, um único problema pode causar a geração de múltiplos alarmes. Eles são resultantes do fato que um único problema normalmente se propaga afetando vários dispositivos da rede, causando a geração de vários alertas simultâneos para a mesma situação. A análise destes alertas depende de fatores como as dependências entre os dispositivos monitorados, os serviços disponíveis na rede e a presença simultânea de outros problemas.

Em uma situação de ataque a um servidor, por exemplo, os enlaces e dispositivos da rede percorridos durante a realização do ataque apresentarão variações do tráfego, causando a geração de alarmes para cada um destes elementos, todos reportando o mesmo problema. Os métodos de localização de anomalias visam solucionar este problema, possibilitando a construção de sistemas que sejam capazes de analisar a rede como um todo, reunindo os alertas gerados de maneira isolada para diferentes elementos em um único relatório, que traga uma visão panorâmica do problema (LI et al., 2008). Eles podem ser definidos como orientados a eventos ou baseados em janelas. Na primeira categoria, o alerta é analisado no momento em que é gerado, levando em conta os alertas anteriores. Após processar o novo alerta, o sistema volta a aguardar pelo próximo, repetindo esse ciclo até que o problema seja diagnosticado. No segundo grupo, são determinados intervalos de tempo, conhecidos também como janelas de tempo, onde todos os alertas gerados são analisados conjuntamente (STEINDER; SETHI, 2004b).

A localização de anomalias se divide em três fases (STEINDER; SETHI, 2004b):

- Detecção do problema: é a fase na qual são gerados os diferentes alertas. Diferentes métodos para detecção de anomalias foram abordados na seção 2.3;
- Elaboração de uma visão panorâmica do problema: esta fase é responsável pela correlação dos alertas gerados na primeira fase;
- Fase de testes: fase na qual há a confirmação da origem do problema, realizada a partir da verificação dos resultados obtidos na segunda fase;

Diferentes estratégias têm sido abordadas para executar os dois últimos passos, como será mostrado nas subseções a seguir. Entre elas, temos os sistemas especialistas, as propostas focadas na modelagem do ambiente monitorado e as soluções baseadas em modelos de propagação dos problemas.

#### **2.4.1 Sistemas especialistas**

Os sistemas especialistas são programas que tentam reproduzir o comportamento de especialistas humanos diante de um determinado domínio de problemas. Em geral, o conhecimento do especialista é modelado na forma de regras, que são armazenadas em bases de dados e consultadas quando é necessária a tomada de decisão (GIARRATANO; RILEY, 2004).

Na localização de anomalias, as ações que são tomadas por um administrador de redes para tratamento dos alertas são modeladas, formando a base de conhecimento utilizada pelo sistema. Para realizar a correlação de alarmes, as regras que compõem a base de conhecimento do sistema podem ser divididas em duas categorias. A primeira inclui regras aplicáveis em diferentes sistemas, que representam conhecimentos genéricos sobre redes variadas e dão respostas aproximadas quando se deparam com um problema. A segunda categoria incorpora regras específicas para a rede em análise. Estas regras são utilizadas para execução da fase de testes, na qual se busca encontrar as razões exatas para o problema, a partir das respostas encontradas com a aplicação prévia das regras genéricas pertencentes à primeira categoria (STEINDER; SETHI, 2004b).

A principal desvantagem dos sistemas baseados em regras é a falta de capacidade de adaptação a problemas novos, que não sejam de conhecimento da base do sistema. A inclusão constante de regras para atender o surgimento de novas possibilidades de problemas pode tornar muito difícil a manutenção destes sistemas, sendo outro ponto de desvantagem.

Para suprir estas lacunas da abordagem baseada em regras, outras informações são adicionadas às soluções propostas. Representações do modelo e das dependências do sistema em análise podem ser incluídas na base de conhecimento, de maneira a melhorar a qualidade dos diagnósticos e conferir capacidade de lidar com problemas novos.

## **2.4.2 Sistemas baseados na modelagem da rede**

Estes sistemas realizam a correlação dos alertas baseados em representações formais das redes monitoradas, focando principalmente nos relacionamentos entre os elementos que compõem a rede. Muitos destes sistemas podem ser definidos como orientados a eventos, nos quais a análise se inicia a partir da geração de um alerta e, conforme outros eventos vão ocorrendo, percorre todo o modelo através de suas ramificações. Desta forma, as possíveis causas do problema são localizadas. Geralmente, modelam a rede através de abordagem orientada a objetos ou utilizando grafos. As classes ou vértices representam os nós da rede e os relacionamentos ou arestas mostram que há conexão entre dois nós (STEINDER; SETHI, 2004b).

A fase de testes é executada durante o trajeto pelo modelo. Conforme ele é percorrido, os respectivos elementos da rede passam por verificações a fim de determinar suas situações. A causa do problema é encontrada quando um elemento com problemas não é dependente de qualquer outro elemento que também esteja apresentando dificuldades. Nas redes onde o teste do nó monitorado pode ser executado automaticamente, esta abordagem representa uma alternativa interessante.

A escolha deste método é indicada quando as relações entre os elementos da rede analisada são triviais.

## **2.4.3 Sistemas baseados em modelos de propagação das anomalias**

Modelos de propagação de anomalias geralmente são representados por grafos de causalidade. Os grafos de causalidade representam a ocorrência da anomalia, indicando quais alarmes podem ser gerados e como eles são relacionados. Cada alarme é representado por um vértice do grafo. As arestas do grafo podem representar a probabilidade de relação entre os alertas registrados nos vértices ou até mesmo a probabilidade dos mesmos serem independentes. A especificação prévia de como os elementos da rede estão relacionados e como as falhas podem se propagar através deles é utilizada para a construção destes grafos. A qualidade desta especificação tem influência direta na eficiência do sistema de localização de anomalias. Quando ocorre a anomalia, o sistema de localização compara o conjunto de alertas gerados com os diferentes cenários possíveis de propagação definidos

nos grafos de causalidade. Quando a comparação retorna uma resposta positiva, é encontrada a causa do problema (STEINDER; SETHI, 2004b).

Diferentes técnicas, geralmente relacionadas à teoria de grafos, são utilizadas para analisar os modelos de propagação de anomalias como o algoritmo dividir-para-conquistar, as gramáticas livres de contexto, técnicas baseadas em *codebook* e redes de crença.

Estes sistemas oferecem resultados mais precisos que os baseados na modelagem da rede. Porém, a definição dos padrões de propagação de anomalias é complexa e pode levar o sistema a ignorar anomalias não previstas no momento da modelagem.

## 2.5 ***Tomografia de redes***

A Internet apresenta algumas particularidades quando analisada sob a perspectiva da gerência de redes. Ela cresceu de maneira desordenada, caracterizando um ambiente aberto, com controle descentralizado e que apresenta grande complexidade. Portanto, os paradigmas de gerência convencionais, que foram desenvolvidos para redes compostas por porções cooperativas e domínios de gerência comuns, não são apropriados para a Internet. A tomografia de redes foi criada para suprir esta lacuna, trazendo um conjunto de técnicas que visam possibilitar o monitoramento de desempenho da Internet. Este método realiza inferências a partir de conjuntos incompletos de informações, normalmente coletados através de sondas. A rede é observada como um todo, evitando a análise individual de elementos, que é comum nos métodos apresentados anteriormente neste trabalho (COATES et al., 2002) (THOTTAN; JI, 2003).

As técnicas de tomografia de redes se dividem em duas categorias. No primeiro grupo, as inferências relacionadas aos enlaces são realizadas a partir de informações coletadas no nível dos caminhos fim-a-fim. Na tomografia no nível de enlace, como é denominada esta categoria, parâmetros de desempenho de um enlace como largura de banda disponível, taxas de perdas e atrasos são estimados a partir de medições em roteadores de borda ou até mesmo *hosts*. Na segunda categoria, denominada tomografia no nível fim-a-fim, estima-se parâmetros de desempenho de caminhos fim-a-fim a partir de medições nos enlaces. Matrizes de tráfego são preenchidas com medições realizadas nas interfaces dos roteadores, que posteriormente são utilizadas para realização das estimativas

de tráfego para cada caminho fim-a-fim, os quais podem incluir diferentes enlaces (COATES et al., 2002).

A aplicação das técnicas de tomografia é útil para a detecção de anomalias em ambientes heterogêneos e com controle descentralizado, como nos trabalhos de Agrawal et al. (2007) e de Nguyen e Thiran (2007). Maiores detalhes sobre estes trabalhos serão apresentados na seção 2.6.

## 2.6 **Trabalhos relacionados**

Esta seção apresenta, de maneira breve, a variedade de soluções que são construídas combinando técnicas e conceitos abordados ao longo deste capítulo. Todas estas soluções foram propostas em publicações coletadas em bases de importantes organizações como o IEEE e ACM. Primeiramente, são apresentadas soluções propostas na área de detecção de anomalias. Logo após, trazemos alguns trabalhos relacionados à localização de anomalias. Por fim, diferentes propostas para a tomografia de redes também são apresentadas.

As anomalias em redes de computadores têm sido estudadas por diferentes autores há quase três décadas, quando os primeiros trabalhos na área, como o de Anderson (1980), começaram a ser publicados. Levantamentos dedicados a apresentar diferentes conceitos e paradigmas da detecção de anomalias podem ser encontrados nos trabalhos de Estevez-Tapiador et al. (2004) e Lim e Jones (2008). Os dois trabalhos apresentam taxonomias para classificar os diferentes sistemas de detecção de anomalias. O trabalho apresentado por Estevez-Tapiador et al. (2004) classifica uma lista de sistemas ligados a projetos de pesquisa, enquanto o trabalho de Lim e Jones (2008) traz produtos comerciais. Características e soluções relacionadas aos sistemas de localização de anomalias são apresentadas por Steinder e Sethi (2004b) em seu trabalho. Por fim, os principais conceitos da tomografia de redes são apresentados por Coates et al. (2002).

O trabalho de Thottan e Ji (2003) propõe um sistema de detecção de anomalias que busca realizar a correlação do comportamento de diferentes objetos SNMP, visando diminuir a taxa de falsos positivos. Os dados de cada objeto SNMP são organizados em séries temporais modeladas a partir da aplicação de um processo Auto-regressivo. Para detectar mudanças de comportamento nestas séries temporais, é aplicado um teste de hipótese baseado no método GLR (*Generalized Likelihood Ratio*). Os resultados deste teste

são organizados em vetores, que são correlacionados posteriormente com base nas características dos objetos SNMP.

O trabalho de Barford et al. (2002) emprega técnicas de processamento de sinais em séries temporais, que são coletadas em MIBs e em registros de fluxos de dados. Para analisar as estatísticas do tráfego de rede coletadas nestas duas fontes, os autores desenvolveram o sistema IMAPIT (*Integrated Measurement Analysis Platform for Internet Traffic*). O IMAPIT aplica algoritmos de *wavelet* para decompor as séries temporais em conjuntos com diferentes frequências. O conjunto com as baixas frequências é formado por eventos de longa duração. O conjunto com as médias e altas-frequências traz os eventos de curta duração. Desta forma, anomalias que são caracterizadas por eventos de longa duração, por exemplo, podem ser mais facilmente visualizadas no conjunto de baixas-frequências. Cada um destes conjuntos de frequências é analisado por um algoritmo denominado *deviation score*, que detecta as anomalias.

Lakhina et al. (2004) analisam estatísticas do tráfego de rede coletadas em registros de fluxos de pacotes. Este trabalho propõe uma técnica para diagnosticar anomalias, oferecendo não só a indicação de ocorrência, mas também a localização da origem desta anomalia. Toda a análise dos dados é baseada em uma técnica de mineração de dados conhecida como PCA, capaz de separar as estatísticas coletadas em dois subespaços: o subespaço normal e o subespaço anômalo. Um trabalho mais recente de Ringberg et al. (2007) mostrou que é difícil aplicar esta técnica em ambientes de produção, já que o ajuste dos parâmetros leva a grandes variações na taxa de falsos positivos.

O trabalho de Roughan et al. (2004) apresenta uma proposta que explora a união de duas fontes de dados para melhorar o desempenho do sistema: o protocolo de gerência SNMP e o protocolo de roteamento BGP (*Border Gateway Protocol*). A partir da correlação entre desvios de comportamento encontrados nestas duas fontes de dados, foi alcançada uma diminuição na taxa de falsos positivos. A análise dos dados coletados nos objetos SNMP foi realizada com o método Holt Winters e a análise dos dados do protocolo BGP foi realizada com o método EWMA (*Exponentially Weighted Moving Average*).

As propostas que utilizam redes de Bayes também são comuns, como o trabalho de Kline et al. (2008), que propõe um algoritmo chamado S3. O primeiro componente do

algoritmo S3 aplica *wavelets* para detectar mudanças de comportamento em séries temporais, que são formadas pelas contagens de pacotes nos fluxos de entrada e saída da rede. O segundo componente busca correlações entre as séries temporais formadas pelos fluxos de entrada e saída. Esta ação é baseada na premissa de que há simetria entre os tráfegos de entrada e saída em situações normais. O último componente utiliza uma rede de Bayes para identificar a ocorrência da anomalia. As redes de Bayes são modelos gráficos capazes de representar relações de causa entre diferentes variáveis. Com este fim, as redes bayesianas, como também são conhecidas, são aplicadas na verificação da relação entre os resultados encontrados com a aplicação dos dois primeiros componentes do algoritmo S3.

Shon e Moon (2007) propõem o uso de uma ferramenta comumente utilizada para reconhecimento de padrões e classificação, as SVM (*Support Vector Machines*). Primeiramente, os pacotes são coletados da rede e filtrados em tempo real pela ferramenta denominada PTF (*Passive TCP/IP Fingerprinting*), que permite que pacotes mal formados sejam identificados e descartados. No conjunto de pacotes filtrados pelo PTF são aplicados dois processos. O primeiro visa determinar o perfil dos pacotes normais utilizando uma técnica de mineração de dados conhecida como SOFM (*Self-Organized Feature Map*). O segundo processo utiliza algoritmos genéticos para selecionar quais campos dos pacotes apresentam maior probabilidade de evidenciar a ocorrência de anomalias. O resultado da execução destes processos é inserido na SVM para que seja efetuado o aprendizado e, posteriormente, sejam detectadas anomalias.

Em geral, os trabalhos de detecção de anomalias não trazem inovações na maneira de coletar informações da rede. O trabalho de Androuliakis et al. (2009) é uma exceção. Segundo os autores, as informações necessárias para a detecção de anomalias estão contidas em uma pequena porção dos fluxos de pacotes. Eles propõem uma amostragem oportunista que busca coletar informações apenas dos fluxos que possam trazer evidências de ocorrências de anomalias. Os métodos de amostragem propostos se dividem em dois grupos. O primeiro procura fluxos de pacotes com poucos pacotes, pois eles normalmente contêm anomalias causadas por *worms* e escaneamento de portas. O segundo grupo busca fluxos com muitos pacotes. Foram trabalhadas anomalias com diferentes causas como *flash crowds*, ataques de negação de serviço e *worms* e os resultados obtidos com a utilização de amostragem oportunista foram melhores que os obtidos com a amostragem comum.

Mahoney e Chan (2002) apresentaram dois sistemas que utilizam o cabeçalho dos pacotes como fonte de dados para a detecção de anomalias. O primeiro sistema é o PHAD (*Packet Header Anomaly Detector*), que monitora 33 campos dos cabeçalhos dos pacotes nas camadas de enlace, rede e transporte. O segundo sistema é o ALAD (*Application Layer Anomaly Detector*) que monitora as conexões abertas com aplicações em servidores, onde os dados analisados são o endereço IP da origem e do destino da conexão, a porta TCP de destino e os flags TCP. Em ambos os sistemas, há um período de treinamento no qual os valores considerados normais para os campos são definidos através de análise estatística. Depois, durante o monitoramento, para cada desvio de comportamento detectado é assinalado um *anomaly score*, utilizado na geração dos alarmes.

Krügel et al. (2002) apresentaram um sistema de detecção de anomalias que analisa os dados do cabeçalho relacionados às camadas de rede e transporte e o *payload* com os dados das aplicações. Somente dados de quatro aplicações são analisados: HTTP, DNS, SMTP e FTP. As requisições destas aplicações são analisadas segundo o tipo da requisição, o tamanho e a distribuição do *payload*. Um *anomaly score* é gerado através do cálculo da média ponderada destas três variáveis e a detecção de anomalias é executada com a comparação entre o *anomaly score* e um limiar determinado pelo administrador do sistema.

O trabalho de Kim e Reddy (2008) se preocupa em detectar anomalias geradas por um agente malicioso que domina a rede e a utiliza para atacar outras redes. Para alcançar este intuito, são inspecionados pacotes coletados nos roteadores de saída da rede. A proposta se baseia no princípio de que o tráfego de saída de um único domínio administrativo apresenta um comportamento bem definido ao longo do tempo. Primeiramente, é construído um sinal com o resultado das análises de correlação entre os endereços e portas de destino dos pacotes coletados sequencialmente. No segundo passo, este sinal é processado utilizando algoritmos de *wavelet*, a fim de separar sinais anômalos de sinais considerados normais. Por fim, são detectadas as anomalias a partir da análise dos resultados do segundo passo, utilizando limiares pré-definidos.

No trabalho de Steinder e Sethi (2004a), é proposto um sistema de localização de anomalias orientado a eventos, que é capaz de trabalhar com dados incompletos ou inconsistentes sobre o sistema analisado. Ele utiliza raciocínio Bayesiano baseado em redes de crença, as quais representam o modelo de propagação de anomalias.

Primeiramente, a solução proposta utiliza uma heurística para a aplicação do algoritmo de atualização das crenças. A partir dos resultados desta aplicação, é utilizado o algoritmo para identificação da explicação mais provável para a anomalia.

Saaman e Karmouch (2008) aplicam inicialmente um processo auto-regressivo em dois ciclos para realizar a detecção de desvios de comportamento nos dados coletados em objetos SNMP. A partir daí, para detectar a causa raiz do problema, é utilizada uma base onde há registros de anomalias com suas causas raízes. A partir desta base, são construídas hipóteses que serão manipuladas segundo a teoria de Dempster-Shafer para determinar qual é a causa mais provável do problema.

No trabalho de Reali e Monacelli (2009), é apresentada uma proposta baseada na abordagem *codebook* para localizar anomalias. A solução representa o modelo de propagação das anomalias em uma matriz que cruza os problemas e seus respectivos sintomas. Esta matriz é resultado da otimização de um grafo de causalidade, onde o número de sintomas necessários para se detectar um problema é minimizado através de heurísticas criadas pelos autores. Na solução proposta, alarmes são gerados quando são violados os limiares de indicadores de desempenho de um nó, como volume de tráfego e nível de utilização de recursos. O conjunto de alarmes gerados para diferentes nós é comparado com os modelos de propagação através da medida de similaridade calculada pela distância de Hamming.

O trabalho de Tang et al. (2009) se distingue dos apresentados até o momento por ter como objetivo a localização de anomalias em redes *overlay*. Estas redes são altamente escalonáveis, impedindo o uso de mapas sintoma-anomalia estáticos. O framework denominado AIR (*Active Integrated Fault Reasoning*) realiza uma alternância balanceada entre medições ativas e passivas, na busca pelas causas-raiz das anomalias. O monitoramento ativo é disparado somente se o monitoramento passivo não é suficiente para alcançar o resultado pretendido.

No trabalho de Agrawal et al. (2007), é apresentada uma proposta de tomografia de redes onde anomalias ocorridas em nível de enlace, como taxas excessivas de perdas ou altos atrasos, são detectadas a partir de medições passivas no nível dos caminhos fim-a-fim. Poucos pontos de monitoramento são estabelecidos em pontos estratégicos, normalmente

mais próximos da borda da rede, a fim de monitorar o comportamento no nível de caminhos fim-a-fim e inferir problemas no nível de enlace. Algoritmos gulosos são utilizados para minimizar a quantidade necessária de caminhos monitorados na realização das inferências.

No trabalho de Nguyen e Thiran (2007), o foco é o cálculo das taxas de perda em enlaces utilizando medições realizadas no nível fim-a-fim. O trabalho é baseado na premissa que perdas causadas por congestionamentos ocorrem em rajadas, acarretando altas variâncias nos índices de perdas em enlaces congestionados. É proposto um algoritmo que combina as taxas de perda fim-a-fim, construindo um sistema de equações lineares que relaciona as variâncias dos enlaces a serem descobertas com as covariâncias das taxas de perda fim-a-fim. A equação linear é então resolvida e os enlaces são classificados conforme a variância encontrada. Os enlaces que apresentam baixa variância, e consequentemente baixo nível de congestionamento, são eliminados, resultando em outro sistema de equações lineares menor. Este sistema é calculado, obtendo as taxas de perdas a partir das variâncias.

### **3 Caracterização de tráfego: modelo BLGBA e DSNS**

A caracterização do tráfego é fundamental para a identificação de ocorrências de anomalias. Sem definir o comportamento normal dos dados analisados, é impossível determinar quando a rede está se comportando de maneira não esperada. Este capítulo apresenta como é realizada a caracterização do tráfego em nossa proposta de detecção de anomalias.

A primeira dificuldade encontrada na caracterização do tráfego é a falta de consenso sobre qual modelo é capaz de executar esta tarefa de maneira eficiente. O tráfego de rede apresenta um nível considerável de ruídos, além de fatores que tornam o seu comportamento dinâmico, como a variação do nível de utilização das redes em diferentes períodos do dia. Somente com o domínio do padrão de operações da rede será possível realizar um diagnóstico de qualidade caso ocorram anomalias (ABUSINA et al., 2005) (HAJJI, 2005) (PROENÇA JUNIOR, 2005).

Neste trabalho, a caracterização de tráfego é realizada a partir da aplicação do modelo BLGBA (*Baseline* para Gerenciamento de *Backbone* Automático) no histórico de informações coletadas da rede. O resultado da aplicação do BLGBA consiste de perfis de comportamento normal denominados DSNS (*Digital Signature of Network Segment*). O DSNS pode ser definido como o conjunto básico de informações que constituem o perfil de operações normais dos dados observados em um objeto SNMP. O modelo BLGBA e o DSNS foram propostos por Proença Junior (2005).

O comportamento do tráfego é formado por ciclos diários, distintos para diferentes dias da semana (BARFORD et al., 2002). Esta característica se torna mais visível quando são comparados os tráfegos dos dias úteis com os fins de semana ou feriados. Os mecanismos de caracterização do tráfego devem estar preparados para lidar com estas situações. Em nossa proposta, a aplicação do modelo BLGBA gera DSNS específicos para cada dia da semana, a fim de diminuir a quantidade de erros decorrentes da análise conjunta de dias com comportamentos diferentes, como dias úteis e feriados. Além disso, cada

segundo do dia também é analisado individualmente, para que o DSNS resultante respeite a variação do tráfego ao longo do dia.

O histórico de dados analisado para geração do DSNS é representado por uma matriz  $M$  de dimensões  $I \times N$ , onde  $I = \frac{86400}{coleta}$  e  $N =$ número de dias analisados. O termo *coleta* define o intervalo usado para busca de dados na MIB. Portanto, se as coletas forem realizadas de 5 em 5 segundos, o conjunto de amostras para um dia conterá o total  $\frac{86400}{5} = 17280$  amostras, já que um dia tem 86400 segundos. Desta forma, cada elemento  $m_{i,j}$  da matriz representa um valor coletado no instante  $i$  do dia  $j$ .

$$M = \begin{bmatrix} m_{1,1} & \cdots & m_{1,N} \\ \vdots & \vdots & \vdots \\ m_{I,1} & \dots & m_{I,N} \end{bmatrix} \quad (3.1)$$

O algoritmo do modelo BLGBA é baseado em uma variação do cálculo da *moda*, que leva em consideração as frequências das classes inferiores e da classe modal. O algoritmo analisa a matriz  $M$  da equação (3.1) linha a linha. Os elementos são separados em frequências, utilizando cinco classes e baseando-se na diferença entre o maior e o menor elemento de cada linha, representados por  $gr_i$  e  $sm_i$ . Esta diferença, dividida por 5, forma a amplitude  $amp_i$  entre as classes, mostrada em (3.2):

$$amp_i = \frac{(gr_i - sm_i)}{5} \quad (3.2)$$

O próximo passo é obter os limites  $Lim_k$  de cada uma das classes. Estes limites são calculados em (3.3), onde  $Ck$  representa a  $k$ -ésima classe ( $k = 1...5$ ).

$$Lim_k = sm_i + (amp_i * k) \quad (3.3)$$

O cálculo tem o propósito de obter o elemento que representa 80% das amostras analisadas em cada linha. O  $Bl_i$  será definido como o maior elemento inserido na classe com frequência acumulada maior ou igual a 80%. O objetivo é obter o elemento que estaria acima da maioria das amostras, respeitando o limite de 80%. O valor de 80% foi escolhido no trabalho de Proença Junior (2005) após a realização de testes de regressão linear, Bland e Altman e análise de resíduos em segmentos de rede da UEL e da UNICAMP. O  $Bl_i$  é, portanto, o representante escolhido pelo BLGBA para cada linha analisada. Ao final da

análise das  $\frac{86400}{coleta}$  linhas da matriz  $\mathbf{M}$ , teremos um  $Bl_i$  para cada linha formando o DSNS resultante.

A Figura 3.1 e a Figura 3.2 ilustram na forma de histogramas a movimentação diária de um fim de semana e de 5 dias úteis, respectivamente, no servidor *proxy* da UEL, na semana de 29/03/2009 a 04/04/2009. O objeto SNMP é o *ipInReceives*, que contabiliza quantos pacotes foram entregues pela camada de enlace à camada de rede do elemento. Ao apresentar estes gráficos, temos como objetivo mostrar um exemplo real da diferença de comportamento dos dados coletados em um objeto para diferentes momentos do dia e dias da semana. No domingo, por exemplo, o tráfego se mantém, na maior parte do tempo, abaixo dos 45 pacotes/s. No sábado, o tráfego é um pouco maior, atingindo os 100 pacotes/s em vários momentos. Durante a semana, no horário comercial, temos níveis de tráfego próximos a 600 pacotes/s. O comportamento do tráfego ao longo de um dia é claramente influenciado pelo regime de horas dos funcionários da universidade. Com a entrada dos funcionários em torno das 8h da manhã, o tráfego cresce e atinge níveis que são mantidos até as 12h, horário em que se inicia o período de almoço. A partir das 13h, o tráfego volta a crescer um pouco, mantendo os níveis que são alcançados até as 18h. Após este horário, com o fim do expediente, os níveis de tráfego voltam a cair. Porém, como há aulas à noite, os níveis de tráfego entre 18h e 22h são significativamente mais altos que os encontrados no período entre 0h e 8h da manhã. O modelo BLGBA é capaz de lidar com este requisito, já que temos estimativas diferentes para cada dia da semana e cada instante onde há uma coleta dentro de um dia.

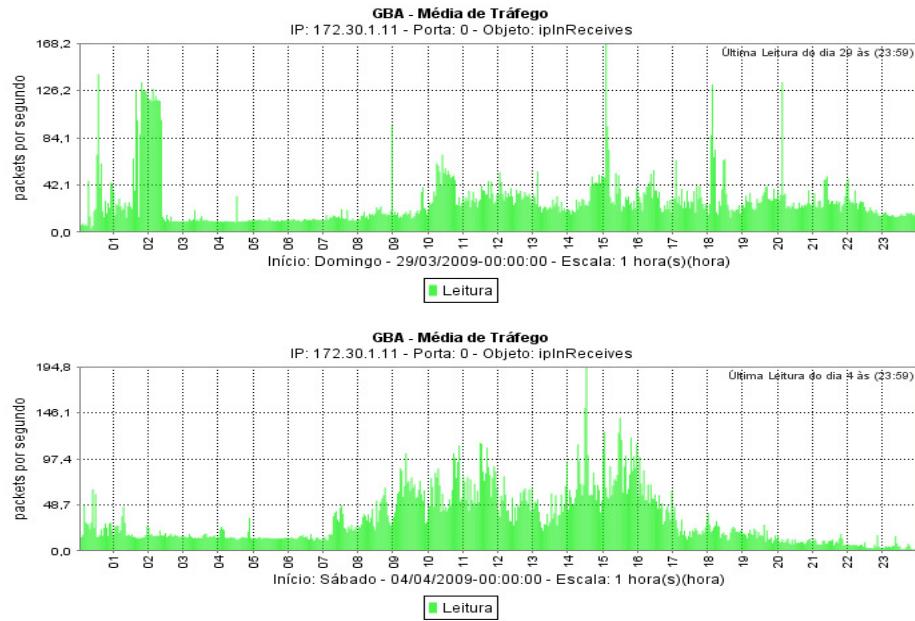


Figura 3.1 - Monitoramento de fim de semana no servidor Proxy da UEL, objeto *ipInReceives*.

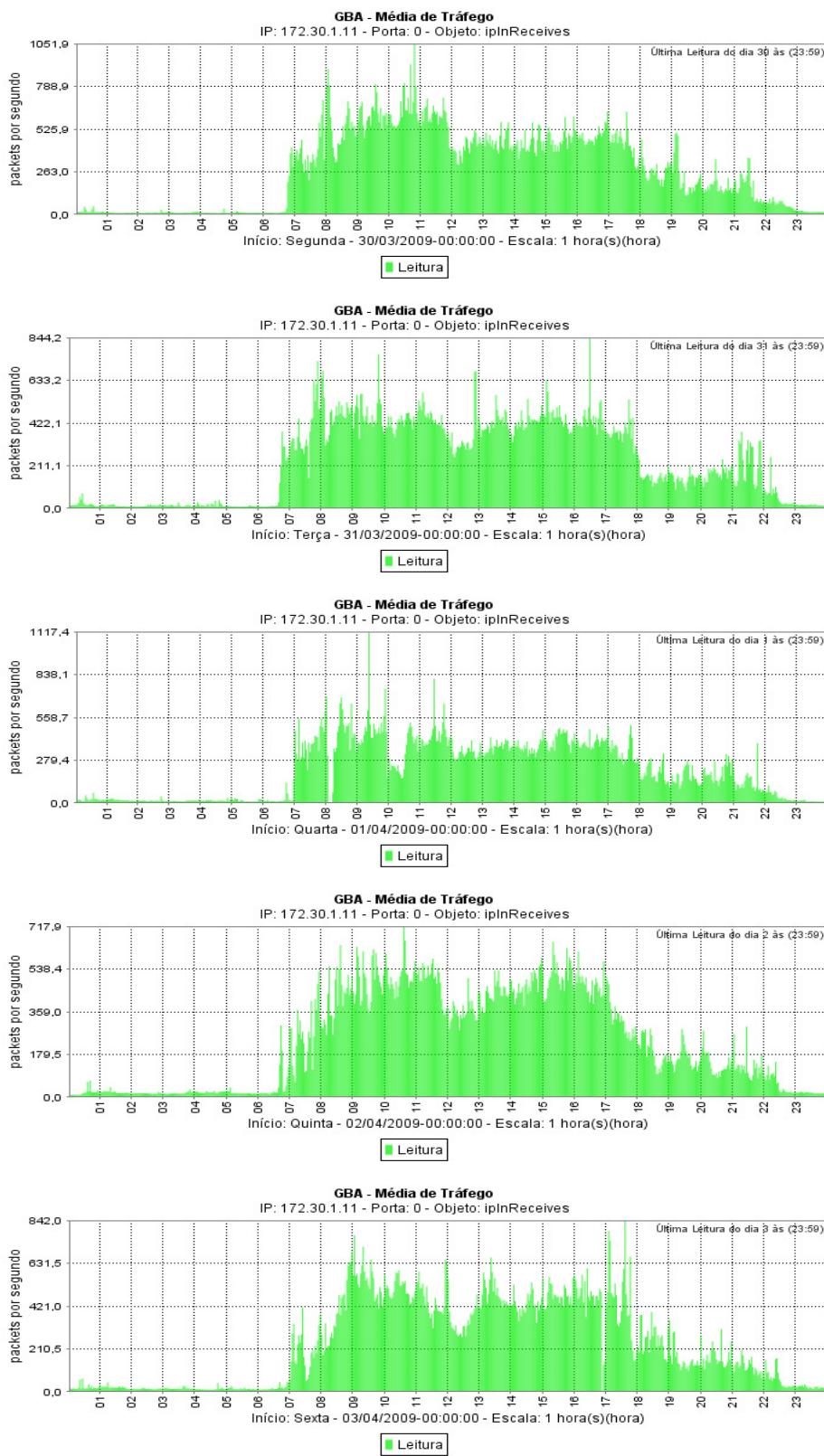


Figura 3.2 - Monitoramento de dias úteis no servidor Proxy da UEL, objeto *ipInReceives*.

A Figura 3.3 e a Figura 3.4, que complementam a Figura 3.1 e a Figura 3.2, respectivamente, apresentam o tráfego real e os DSNS, gerados a partir da aplicação do BLGBA. O tráfego real é apresentado nas cores verde e vermelho, enquanto o DSNS é apresentado na cor azul. O DSNS está bem ajustado ao tráfego real, acompanhando todas as variações causadas pelas mudanças ligadas ao horário de expediente dos funcionários da universidade. O DSNS foi gerado após a análise de 12 semanas de tráfego. O trabalho de Proença Junior (2005) determinou, após a realização de testes de regressão linear, Bland e Altman e de análise de resíduos, que este é o número de semanas considerado ótimo para geração de DSNS.

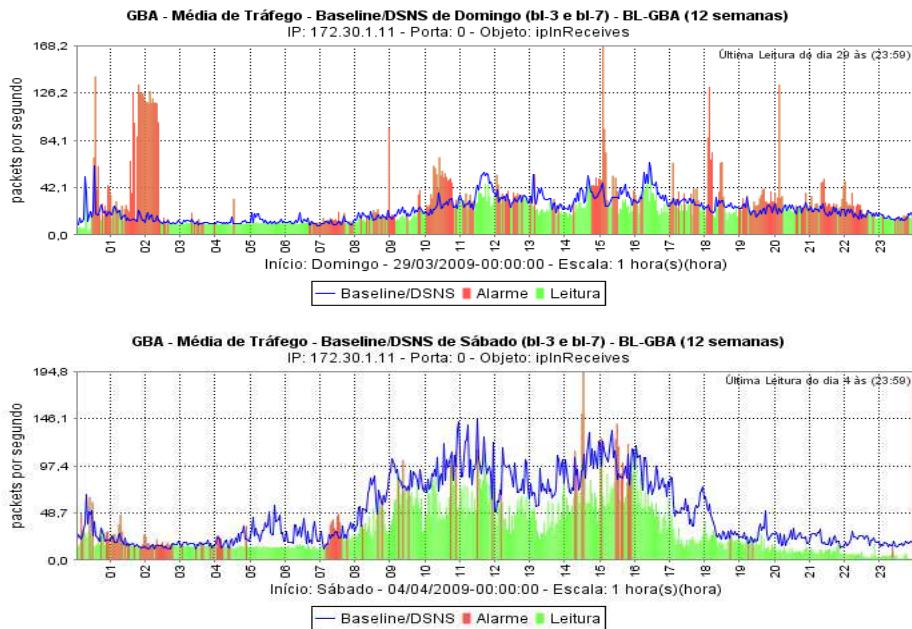


Figura 3.3 - Monitoramento de fim de semana com os respectivos DSNS no servidor Proxy da UEL, objeto *ipInReceives*.

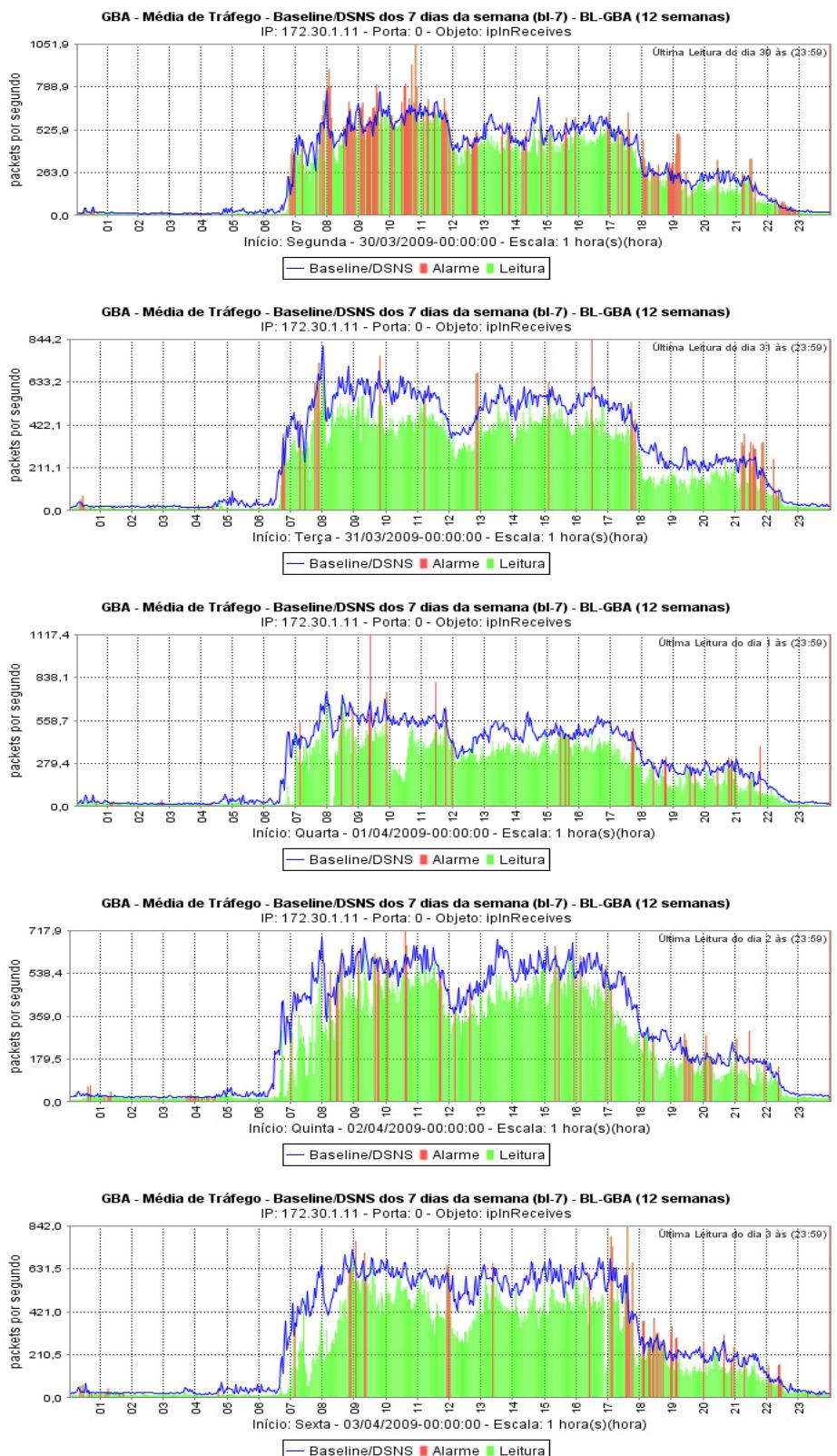


Figura 3.4 - Monitoramento de dias úteis com os respectivos DSNS no servidor

Proxy da UEL, objeto *ipInReceives*.

## 4 Detecção de anomalias em três níveis

Este capítulo traz os detalhes acerca da solução de detecção de anomalias em três níveis, que forma o núcleo da proposta desta tese. Primeiramente, é apresentada a abordagem utilizada em nossa proposta para determinar quais desvios de comportamento são considerados anomalias. É uma solução parametrizada, que permite que sejam estudadas anomalias com diferentes características, atendendo a diferentes políticas de gerência. O segundo tópico apresentado trata de uma visão geral da arquitetura em três níveis proposta para detecção de anomalias. Após a seção com a visão geral da arquitetura, teremos seções que detalham cada um dos módulos que compõem a solução proposta.

### 4.1 Definição parametrizada de anomalia

A noção de anomalia, em geral, é construída sobre elementos subjetivos (SOULE et al., 2005). Quando o objetivo é detectar anomalias relacionadas apenas a eventos que são incluídos nos *logs* da rede, como falhas em equipamentos ou ataques que causam a interrupção dos serviços, não há dúvidas sobre quais situações devem ser identificadas pelo sistema de detecção de anomalias. Porém, há desvios de comportamento que não resultam em entradas nos *logs* de eventos do sistema, mas seriam de interesse do administrador de rede e por isso deveriam ser detectados (ROUGHAN et al., 2004). Para estes casos, a definição de quais desvios de comportamento devem ser considerados anomalias depende da política de gerência da rede, que pode ter como objetivo realizar um monitoramento mais rígido, no qual a grande maioria dos desvios de comportamento deve ser reportada independente da gravidade de suas consequências.

Segundo Soule et al. (2005), uma abordagem comum é entregar a definição do conjunto de anomalias a ser utilizado na avaliação do sistema a um administrador de rede experiente. Com o auxílio de gráficos e outras ferramentas de gerência, o administrador aponta quais eventos devem ser considerados anômalos em um conjunto de dados coletado de uma rede real. O sistema de detecção é aplicado então sobre o conjunto de dados avaliado pelo administrador, para que seja aferida a sua capacidade de atender aos objetivos

da política seguida pelo administrador ao analisar os eventos. O problema desta abordagem é que ela depende totalmente do conhecimento do administrador, que pode cometer erros durante a sua análise. Como alternativa a esta abordagem, Lakhina et al. (2004) substituem o papel do administrador presente no cenário de Soule et al. (2005) pela aplicação de métodos como o EWMA (*Exponentially Weighted Moving Average*) e a análise de Fourier. Os desvios identificados por estas técnicas formam, portanto, a base de anomalias que é utilizada para avaliar o sistema de detecção de anomalias.

Trabalhos como os de Al-Kassasbeh e Adda (2009) recorrem à geração artificial de anomalias. Em um ambiente de rede controlado, são gerados eventos através da injeção de tráfego, nos quais os parâmetros dos testes são definidos para verificar o comportamento do sistema perante anomalias com diferentes características. A vantagem desta abordagem é que ela permite que o detector de anomalias seja avaliado sob diferentes condições. A principal desvantagem é que estes testes não são realizados em ambientes reais, fazendo com que algumas situações avaliadas não encontrem correspondência no mundo real.

Neste trabalho, decidiu-se traduzir de maneira objetiva a análise visual que os administradores de rede realizam utilizando os gráficos gerados pela ferramenta GBA (Gerenciamento de *Backbone* Automático) (PROENÇA JUNIOR, 2005). O GBA foi desenvolvido como ferramenta para auxiliar a gerência e o aprendizado na área de redes de computadores, tendo a sua primeira versão finalizada em 1998. Desde então, o GBA tem passado por diversas melhorias, encontrando-se atualmente na sexta versão. Além de módulos para coleta de dados em MIBs SNMP e para geração de DSNS, o GBA possui um módulo para geração de gráficos.

A definição da anomalia é parametrizada, permitindo que sejam realizados testes utilizando anomalias com diferentes características. Um gráfico de movimento diário produzido pela ferramenta GBA é composto por duas séries de dados: as leituras do objeto SNMP e o DSNS correspondente. Cada uma destas séries é formada por 600 pontos. Cada ponto representa a média de 144 segundos, já que um dia possui 86400 segundos. No algoritmo apresentado na Tabela 4.1, são apresentados os passos que definem se um desvio de comportamento é anomalia ou não.

Tabela 4.1 - Algoritmo utilizado para definir se um desvio de comportamento deve ser considerado como anomalia.

---

#### **ALGORITMO: DEFINIR DESVIO DE COMPORTAMENTO COMO ANOMALIA**

---

**Entrada:** séries de dados com dados do objeto SNMP e do DSNS, contidas no eixo y de um gráfico da ferramenta GBA;

**Saída:** conjunto de desvios de comportamento definidos como anomalias;

**Notação:**

$Y_{\text{objeto}}$  : série de dados que representa os valores coletados no objeto SNMP;

$Y_{\text{DSNS}}$  : série de dados que representa os valores do DSNS;

```

01. PROCEDIMENTO definirAnomalia( $\alpha$  ,  $\gamma$ )
02. INICIO
03.         PARA  $i=1\dots totalAmostras$  FAÇA
04.             SE ( $Y_{\text{objeto}}(i) \geq Y_{\text{DSNS}}(i) + (Y_{\text{DSNS}}(i) * \alpha)$ ) ENTÃO
05.                 incrementa contadorViolações;
06.             SENÃO
07.                 contadorViolações := 0;
08.             FIM SE;
09.             SE (contadorViolações >  $\gamma$ ) ENTÃO
10.                 define desvio como anomalia;
11.                 contadorViolações := 0;
12.             FIM SE;
13.         FIM PARA;
14. FIM PROCEDIMENTO;

```

---

O algoritmo apresentado define uma anomalia como um conjunto de violações consecutivas. Cada vez que um valor da série de dados do objeto SNMP ultrapassa o DSNS adicionado de um fator de tolerância  $\alpha$ , é contabilizada uma violação. Se ocorrerem mais que  $\gamma$  violações consecutivas, o desvio de comportamento é definido como uma anomalia. Os valores de  $\alpha$  e  $\gamma$  são passados como parâmetros, permitindo que sejam definidas anomalias com diferentes características conforme se modifica os valores destes parâmetros.

## 4.2 **Arquitetura da solução proposta**

Esta seção apresenta a arquitetura da solução proposta, identificando seus principais objetivos e detalhando a responsabilidade de cada elemento e a interação entre eles. A Figura 4.1 ilustra a divisão em três níveis utilizada pelo sistema para coletar os dados da rede e processá-los até chegar a uma conclusão sobre a ocorrência de anomalias. Inicialmente, temos os dados coletados dos objetos SNMP, que se encontram em um estado

natural. Estes dados são apenas contagens de pacotes ou *bytes*, que analisadas fora do contexto onde estão inseridas não conseguem mostrar se há anomalias no tráfego da rede. Os três níveis de análise trazem informações sobre o contexto onde os dados coletados estão inseridos, possibilitando a análise do comportamento da rede e a consequente detecção das anomalias.

No primeiro nível de análise, temos a caracterização do tráfego e a comparação dos dados do objeto SNMP com o DSNS. Caso os dados coletados do objeto SNMP não estejam coerentes como o perfil de comportamento normal, são gerados alarmes de primeiro nível. Porém, isto ainda não é suficiente para determinar se um elemento de rede está atravessando problemas ou não. O primeiro nível de análise trabalha os objetos SNMP individualmente e não tem, portanto, uma visão abrangente do que está ocorrendo no elemento de rede.

O segundo nível é o responsável por analisar os alarmes de primeiro nível gerados para diferentes objetos SNMP em um único elemento de rede. A partir da correlação de alarmes de primeiro nível de diferentes objetos SNMP, será determinado qual é o comportamento da anomalia no elemento de rede analisado, gerando um alarme de segundo nível. Diferente do primeiro nível de análise, o qual analisa o objeto individualmente, o segundo nível de análise trabalha com conjuntos de objetos de um mesmo elemento de rede, sendo capaz de determinar se está ocorrendo uma anomalia neste elemento ou não. Por outro lado, o segundo nível de análise ignora as relações do elemento de rede analisado com outros elementos de rede, o que o torna incapaz de determinar como a anomalia está se propagando pela rede.

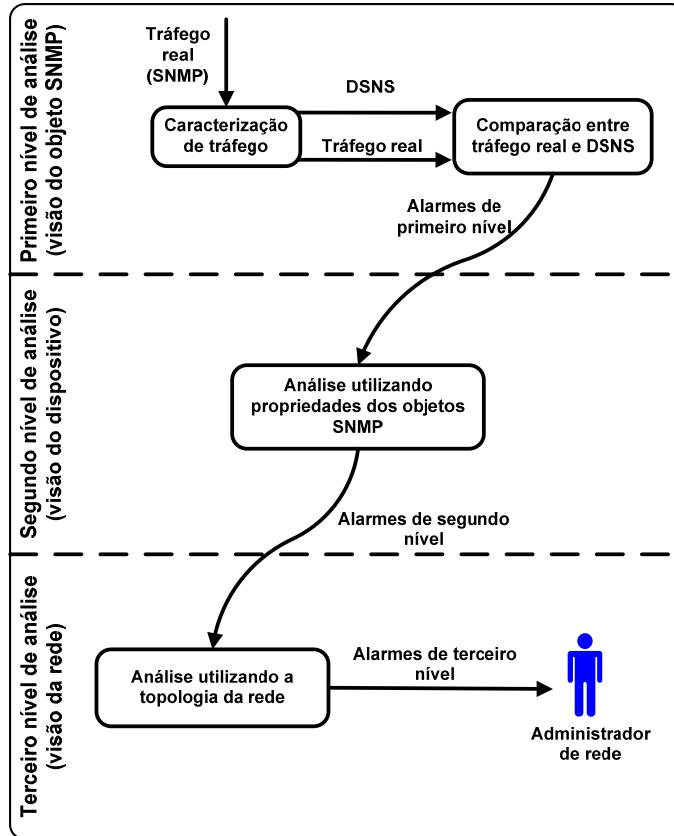


Figura 4.1 - Fluxo da informação e níveis de análise no sistema proposto.

O terceiro nível de análise traz a visão panorâmica de como a anomalia está se propagando pela rede. Neste nível, os alarmes de segundo nível emitidos pelos diferentes elementos da rede são analisados em conjunto com informações da topologia, mapeando a propagação da anomalia pela rede. Um alarme de terceiro nível é gerado apresentando o mapa da propagação da anomalia pela rede.

Conforme se percorre os níveis de análise, são adicionadas ao processo mais informações do contexto monitorado. Desta forma, é possível partir da análise individual de dados em estado bruto como os coletados nos objetos SNMP e chegar a alarmes de terceiro nível contendo a conclusão sobre o comportamento do problema em toda a rede.

A arquitetura do sistema proposto é apresentada na Figura 4.2. Nela podemos observar a interação entre os diversos módulos do sistema.

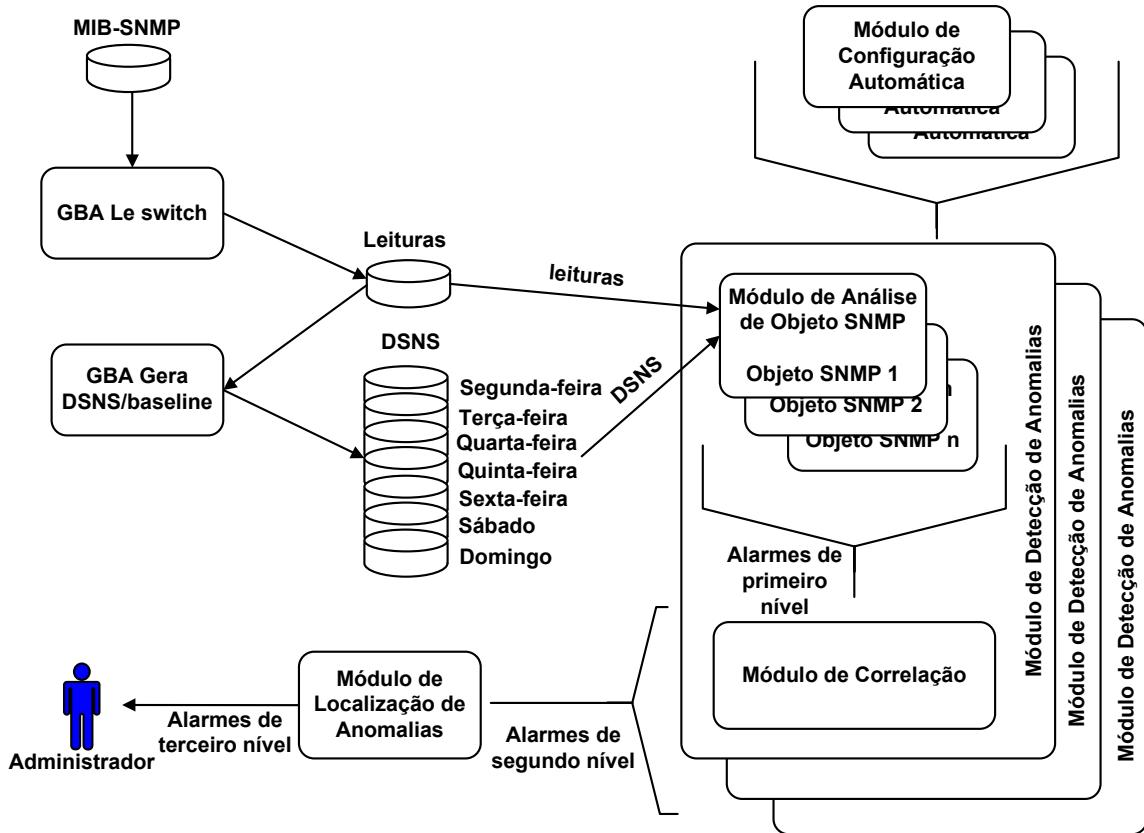


Figura 4.2 - Visão geral do modelo proposto para tratamento de anomalias.

Tanto a coleta de dados quanto a geração dos perfis de comportamento, denominados DSNS (*Digital Signature of Network Segment*) são realizados por módulos da ferramenta GBA.

A coleta dos dados é realizada pelo módulo denominado *GBA Le switch*, que coleta as observações em tempo real e as armazena em disco. O módulo denominado *GBA Gera DSNS/baseline* é o responsável pela caracterização de tráfego e armazenamento em disco dos DSNS. É importante mencionar que é gerado um DSNS específico para cada dia da semana e para cada objeto SNMP. Maiores detalhes da geração do DSNS estão presentes no capítulo 3.

Para cada objeto SNMP monitorado em um elemento de rede teremos uma instância do Módulo de Análise de Objeto SNMP, que executa o primeiro nível de análise. Este módulo, que está inserido dentro do Módulo de Detecção de Anomalias, realiza a comparação entre as leituras reais e o DSNS, a fim de descobrir desvios de comportamento

no objeto SNMP. Estes desvios serão reportados ao sistema através de alarmes de primeiro nível.

Para cada um dos  $n$  dispositivos monitorados teremos uma instância do Módulo de Correlação, que também está inserido no Módulo de Detecção de Anomalias e corresponde ao segundo nível de análise do sistema. Ele é encarregado de combinar os alarmes do primeiro nível de cada um dos objetos monitorados e indicar se o dispositivo atravessa uma anomalia ou não. Caso seja confirmada a anomalia no dispositivo de rede, é gerado um alarme de segundo nível.

As instâncias do Módulo de Detecção de Anomalias são acompanhadas por instâncias do Módulo de Configuração Automática. Este módulo é responsável por selecionar valores para os parâmetros de sensibilidade do sistema que sejam adequados à política de gerência da rede.

A ocorrência de uma anomalia acarretará a geração de alarmes do segundo nível redundantes nos diferentes dispositivos de rede que forem afetados pelo problema. O Módulo de Localização de Anomalias, que corresponde ao terceiro nível de análise do sistema, é responsável por agrupar estes alarmes de segundo nível e verificar como a anomalia está se propagando pela rede. Desta forma, o sistema poderá oferecer ao administrador uma visão panorâmica do problema. A mensagem gerada pelo sistema ao administrador de rede, além de indicar a ocorrência da anomalia, deverá conter também um mapa com a propagação do problema, cujo objetivo é facilitar a solução do mesmo. Esta mensagem é conhecida como alarme de terceiro nível.

### 4.3 **Módulo de Detecção de Anomalias**

O Módulo de Detecção de Anomalias reúne em seu escopo o Módulo de Análise de Objeto SNMP e o Módulo de Correlação, sendo responsável pelas seguintes tarefas:

- Análise de cada objeto SNMP monitorado em determinado equipamento;
- Correlação dos alarmes do primeiro nível e a consequente geração dos alarmes do segundo nível;
- Envio dos alarmes de segundo nível ao Módulo de Localização de Anomalias;

#### 4.3.1 Módulo de Análise de Objeto SNMP

O Módulo de Análise de Objeto SNMP é responsável por comparar o DSNS aos dados reais coletados em um objeto SNMP, a fim de identificar a ocorrência de desvios de comportamento e gerar alarmes quando estes desvios forem significativos. Os alarmes gerados, denominados alarmes do primeiro nível, são enviados ao Módulo de Correlação.

As primeiras soluções comerciais que geravam alarmes para notificar o administrador de redes sobre desvios de comportamento operavam de maneira bem simples. Nestes sistemas, limiares constantes eram determinados pelo próprio administrador de rede ou através de estatísticas simples como a média, sem considerar as variações do tráfego em horários diferentes do dia ou diferentes dias da semana. Toda vez que o tráfego superava estes limiares, alarmes eram gerados (HAJJI, 2005). O problema desta abordagem é que flutuações ocorridas no tráfego faziam com que o limiar fosse superado várias vezes em um curto espaço de tempo, causando a geração de alarmes em excesso.

O esquema de geração de alarmes do protocolo RMON (*Remote Network Monitoring*) (RFC 1757, 1995), por sua vez, diminui a quantidade de alarmes gerados, utilizando uma ferramenta denominada mecanismo de histerese. Os alarmes são gerados nas seguintes condições, aplicando um limite inferior e um limite superior:

- É gerado um alarme para o limite superior se o valor atual coletado da rede é maior que este limite e o último evento que causou a geração de um alarme foi relacionado ao limite inferior;
- É gerado um alarme para o limite inferior se o valor atual coletado da rede é menor que este limite e o último evento que causou a geração de um alarme foi relacionado ao limite superior.

Ao condicionar a geração de alarmes à alternância dos eventos no limite superior e no limite inferior, o mecanismo desenvolvido para o RMON diminuiu o impacto das flutuações do tráfego na quantidade de alarmes gerados. A Figura 4.3 ilustra o funcionamento do mecanismo de histerese para geração de alarmes do RMON. No ponto indicado por (a), há a geração de um alarme para o limite inferior. No ponto (b), por sua vez, não há geração de alarme, já que o último alarme gerado também foi para o limite inferior. No ponto (c), onde o limite superior é ultrapassado, um novo alarme é gerado. O

ponto (d) não tem geração de alarme, pois o evento anterior também envolveu violação do limite superior. No ponto (e), como o último alarme foi gerado para o limite superior, há a geração de um novo alarme para o limite inferior. Observa-se nesta situação que foram gerados apenas 3 alarmes. Se fossem gerados alarmes para toda violação de limite, teriam sido gerados 5 alarmes, ou seja, 66% a mais.

Como base para o desenvolvimento do Módulo de Análise de Objeto SNMP, foi utilizado o mecanismo de geração de alarmes do protocolo RMON. A aplicação deste mecanismo no Módulo de Análise de Objeto SNMP requer adaptações, já que ele utiliza o DSNS e não dois limites como o RMON. Isto representa uma vantagem, já que o DSNS acompanha os diferentes ciclos de comportamento do tráfego ao longo do dia, enquanto os limites utilizados no RMON são constantes. O objetivo é desenvolver um mecanismo de histerese semelhante ao do protocolo RMON, para evitar que sejam gerados alarmes toda vez que uma leitura superar o limite estabelecido no DSNS.

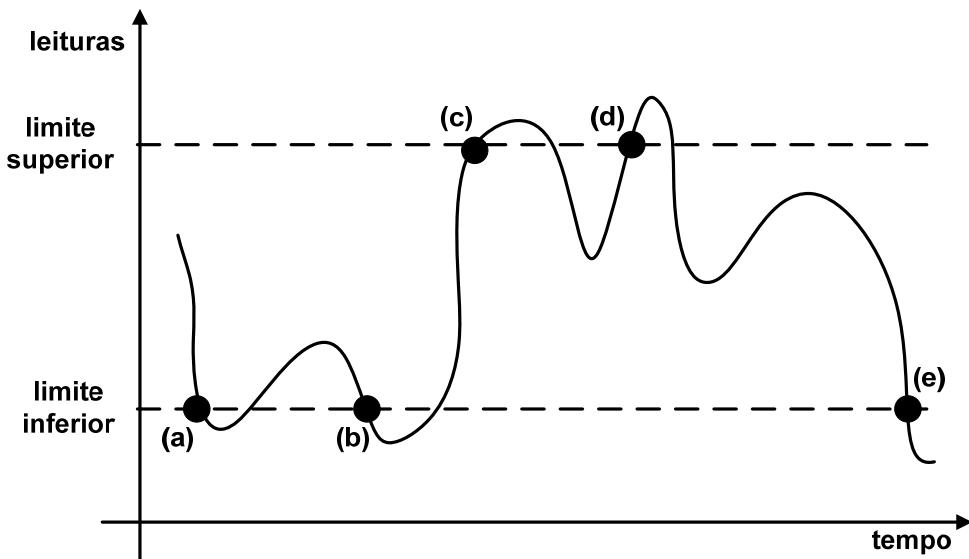


Figura 4.3 - Ilustração do mecanismo de histerese do RMON.

A Figura 4.4 ilustra a primeira proposta. Como não há dois limites, não é possível exigir a alternância entre eventos superiores e inferiores para a geração de alarmes. Portanto, para evitar a geração de alarmes em toda flutuação de tráfego acima do DSNS, foi criado um mecanismo de estabelecimento de novos limites. Quando uma leitura ultrapassa

pela primeira vez o valor determinado pelo DSNS, como no ponto (a) da Figura 4.4, é definido um novo limite com esta leitura e é gerado um alarme. Caso a leitura se situe abaixo do DSNS, como no ponto (b), nenhuma ação é tomada. Caso as próximas leituras sejam superiores ao DSNS e inferiores ao limite definido anteriormente como no ponto (c) da Figura 4.4, há a renovação do limite, mas não há a geração de um alarme. Para que um novo alarme seja gerado, uma nova leitura tem de ultrapassar o limite estabelecido no momento, assim como ocorre no ponto (d) da Figura 4.4.

A geração de alarmes utilizando esta primeira proposta mostrou dois problemas principais. Mesmo com o mecanismo de histerese, o número de alarmes gerados é muito alto. A aplicação deste algoritmo para o objeto *ipInReceives* do servidor *proxy* da UEL na semana de 29/03/2009 a 04/04/2009 gerou 2339 alarmes. Em uma semana de monitoramento deste objeto, são coletadas 60480 amostras, já que o intervalo de coleta é de 10 segundos. Considerando 2339 alarmes gerados, temos um alarme a cada 25 amostras coletadas em média, ou um alarme a cada 4min10s. O outro problema desta proposta é a impossibilidade de ajustar a sensibilidade do algoritmo, fato que impede que o sistema de detecção se adapte a diferentes políticas de gerência.

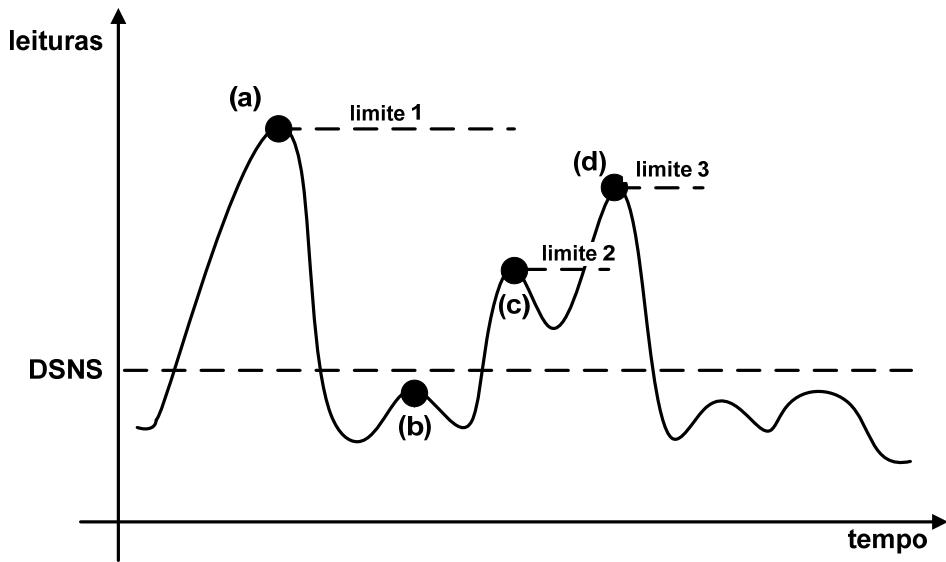


Figura 4.4 - Primeira proposta de mecanismo de histerese para o Módulo de Análise de Objeto SNMP.

A proposta final do algoritmo para o Módulo de Análise de Objeto SNMP busca solucionar estes dois problemas. Esta proposta final inclui uma regra adicional em relação à primeira proposta apresentada. Ao invés de gerar alarmes para toda a violação de limite ocorrida, a proposta final gera alarmes somente quando o número de violações supera um dado patamar, diminuindo o número de alarmes gerados. O número de violações toleradas pelo algoritmo é definido em um parâmetro denominado  $\delta$ . Quanto maior o  $\delta$ , maior o número de violações necessárias e menor a sensibilidade do algoritmo. Segundo na direção inversa, quanto menor o  $\delta$ , mais sensível é o algoritmo e mais alarmes são gerados. A inclusão do parâmetro  $\delta$  permite que haja uma calibragem do algoritmo para diferentes cenários e situações, atendendo ao requisito mencionado anteriormente.

A proposta final do algoritmo é orientada pela identificação de três eventos, que estão relacionados com a violação do DSNS, dos limites e do  $\delta$ . A Figura 4.5 mostra um diagrama de atividades que ilustra os passos do algoritmo. O primeiro passo do algoritmo consiste da comparação entre a leitura real realizada no objeto SNMP e o respectivo valor do DSNS. Caso a leitura real tenha um valor maior que o valor do DSNS, um evento do tipo 1 é gerado. Com a identificação do evento 1, a contagem de um intervalo de tempo de duração pré-definida é iniciada. Foi necessária a criação deste intervalo de tempo, denominado *intervalo de histerese*, no qual são contadas as violações de limites para a tomada de decisão em relação à geração dos alarmes. Quando o evento 1 é identificado, o novo limite passa a ser a leitura que ultrapassou o DSNS. Dentro do *intervalo de histerese*, toda vez que a leitura supera o DSNS, é renovado o limite. Se além de superar o DSNS, a leitura superar o limite corrente, é identificada a ocorrência de um evento 2. Quando a contagem de eventos 2 em um único *intervalo de histerese* é maior que o valor determinado em  $\delta$ , é gerado o evento 3 e um alarme.

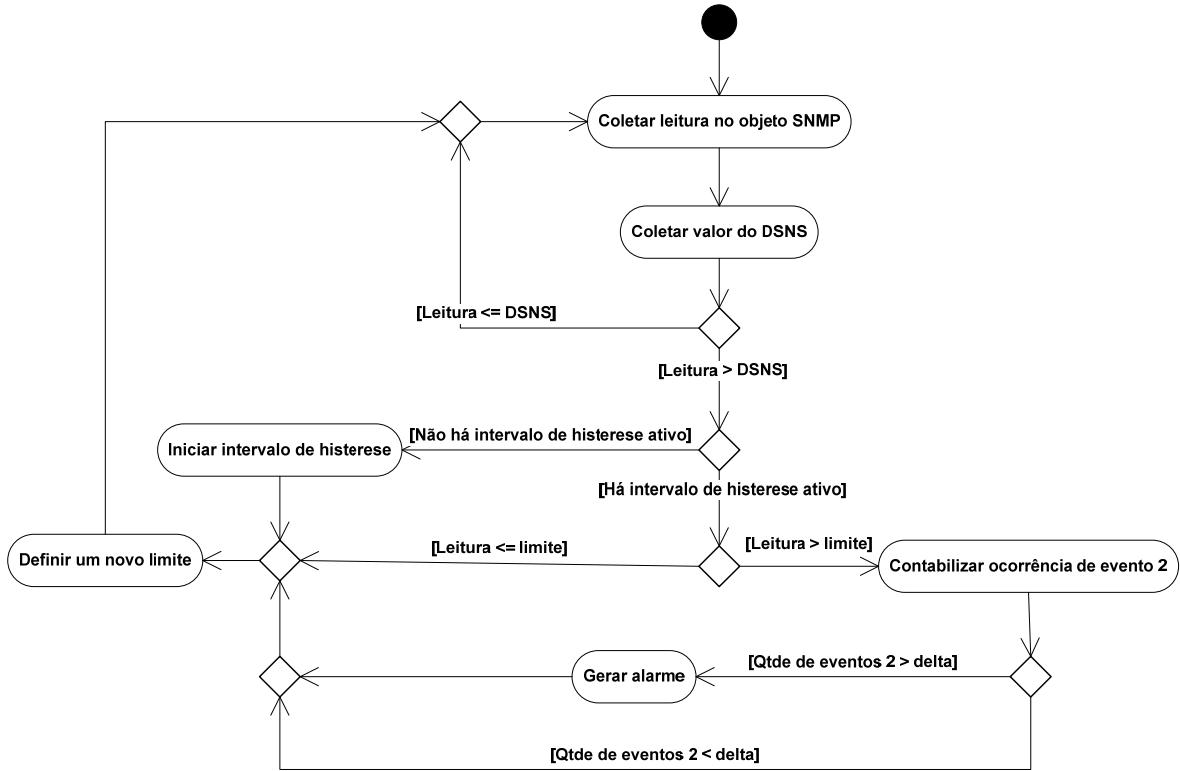


Figura 4.5 – Diagrama de atividades para o algoritmo do Módulo de Análise de Objeto SNMP.

Este algoritmo foi aplicado no objeto *ipInReceives* do servidor *proxy* da UEL na semana de 29/03/2009 a 04/04/2009 com diferentes valores de  $\delta$ . A Tabela 4.2 mostra os resultados para valores de  $\delta$  que variam de 1 a 13, já que para valores superiores a 13 não foi gerado nenhum alarme. Considerando o valor de  $\delta$  mais baixo aplicado, que define a configuração de maior sensibilidade para o algoritmo, já podemos observar uma forte diminuição no número de alarmes em relação aos 2339 alarmes gerados pela proposta inicial. Além disso, é possível observar como o aumento do  $\delta$  e a consequente diminuição da sensibilidade vão alterando os resultados obtidos.

Uma importante constatação obtida com estes resultados preliminares é que o administrador tem em suas mãos uma ferramenta que pode ser configurada com diferentes sensibilidades, apresentando comportamentos ajustáveis para as políticas de gerência que pretende aplicar em cada elemento de sua rede.

Tabela 4.2 - Alarmes gerados para o objeto *ipInReceives*, servidor Proxy da UEL de 29/03/2009 a 04/04/2009.

$\delta$	Quantidade de alarmes	Número médio de amostras entre dois alarmes	Intervalo médio de tempo entre dois alarmes
1	361	167	27 minutos
2	227	266	44 minutos
3	157	385	1 hora e 4 minutos
4	107	565	1 hora e 34 minutos
5	79	765	2 horas e 7 minutos
6	58	1042	2 horas e 53 minutos
7	39	1550	4 horas e 18 minutos
8	28	2160	6 horas
9	17	3557	9 horas e 52 minutos
10	9	6720	18 horas e 40 minutos
11	3	20160	56 horas
12	2	30240	84 horas
13	2	30240	84 horas

Por fim, é importante observar que os alarmes gerados pelo algoritmo construído para o Módulo de Análise de Objeto SNMP não devem causar a notificação imediata do administrador, já que eles não indicam a ocorrência de uma anomalia e sim de um desvio de comportamento em um objeto SNMP. Mesmo assim, a sinalização dos alarmes gerados fica disponível em arquivos de *logs* e na forma de gráficos referentes à movimentação da rede, para que os administradores possam fazer uma análise posterior mais precisa sobre os eventos ocorridos, caso seja necessário. Informações como momento de geração, quantidade e frequência dos alarmes e valores do tráfego real e do DSNS podem ser úteis no planejamento da rede para que situações anômalas futuras passíveis de prevenção sejam evitadas.

### 4.3.2 Módulo de Correlação

O Módulo de Correlação é responsável por reunir os diferentes alarmes de primeiro nível gerados pelas instâncias do Módulo de Análise de Objeto SNMP, verificando se eles

estão relacionados para emitir um alarme de segundo nível que agregue as informações presentes em cada um destes alarmes de primeiro nível.

Na detecção de anomalias utilizando o protocolo SNMP, a coleta de dados em vários objetos de gerência é fundamental, pois cada um deles oferece diferentes perspectivas do comportamento do elemento de rede e dos problemas identificados, melhorando o diagnóstico e diminuindo a taxa de falsos positivos. O real aproveitamento dos benefícios que podem ser trazidos pelo monitoramento de vários objetos está relacionado à capacidade do sistema em lidar com dados heterogêneos, que nem sempre apresentam relacionamentos triviais. Esta capacidade pode ser alcançada com a aplicação de técnicas de correlação dos alarmes gerados para cada um dos objetos monitorados (SIRIS; PAPAGALOU, 2006) (VALDES; SKINNER, 2001).

A correlação de alarmes ocorre quando dois ou mais alarmes apresentam relação de interdependência ou causalidade entre si. Constatada esta relação, é possível então aplicar métodos que reúnem sistematicamente estes alarmes em unidades únicas para tratamento e análise. Em resumo, a correlação de alarmes é a interpretação conceitual de múltiplos alarmes, que atribui novos significados aos grupos de alarmes analisados (JAKOBSON; WEISSMAN, 1993). Ela leva à diminuição do volume de alarmes que devem ser analisados e à melhora na qualidade das informações incluídas na notificação resultante (CHAO et al., 2001).

Alguns objetos SNMP reagem de modo diferente ao mesmo evento. Enquanto uns refletem com maior clareza a anomalia, outros não apresentam qualquer desvio de comportamento. Isto ocorre porque a anomalia pode se propagar de diferentes maneiras dentro do elemento de rede, dependendo se ele está recebendo, gerando ou roteando tráfego anômalo. Estes detalhes no comportamento dos objetos SNMP durante a ocorrência de uma anomalia devem ser considerados ao construirmos as regras de correlação dos alarmes. Os objetos SNMP apresentam relacionamentos entre si, que devem ser explorados para que o caminho da anomalia dentro do dispositivo de rede seja mapeado e a probabilidade de ocorrência de falsos positivos seja diminuída.

Nas duas próximas seções, é apresentada a proposta de correlação de objetos SNMP desenvolvida neste trabalho. Primeiramente, são detalhadas as características dos objetos

SNMP utilizados na proposta. Utilizando os conceitos apresentados sobre os objetos SNMP, é apresentado o grafo de dependências proposto para representar as relações entre estes objetos. Por fim, é apresentado o algoritmo que utiliza o grafo de dependências para analisar os alarmes de primeiro nível e gerar um alarme de segundo nível que traga um mapa de comportamento da anomalia dentro do dispositivo de rede.

#### **4.3.2.1 Objetos SNMP**

Existem diversas MIBs disponíveis para utilização. Cada uma delas apresenta suas particularidades e objetivos, que são materializados nos objetos de gerência catalogados. A MIB-II, apresentada no RFC 1213 (1991), é definida pelo IETF como o padrão para gerência de redes TCP/IP e Internet. Existem também MIBs proprietárias, disponibilizadas pelos fabricantes, que oferecem objetos específicos para seus equipamentos. Um dos objetivos desta tese é propor uma abordagem para detecção de anomalias que seja baseada em padrões de gerência de redes. Portanto, será utilizada apenas a MIB-II.

A MIB-II possui 220 objetos SNMP, que oferecem diversas medições relacionadas aos parâmetros de funcionamento das redes TCP/IP. Dois objetivos nos levaram a realizar um estudo sobre as características destes objetos, de forma a aplicá-los neste trabalho:

1. Escolher um subconjunto de objetos que sejam adequados à detecção de anomalias, já que a grande quantidade de objetos torna impossível o monitoramento de todos eles;
2. Levantar os relacionamentos entre os objetos SNMP escolhidos. Esses relacionamentos permitirão que os alarmes de segundo nível apresentem informações sobre o comportamento da anomalia no dispositivo de rede, de forma a apoiar o trabalho desempenhado pelo Módulo de Localização de Anomalias no tratamento destes alarmes.

Para atingir o primeiro objetivo, é necessário analisar quais são as necessidades do sistema de detecção de anomalias e quais objetos podem atendê-las. O Módulo de Detecção de Anomalias deve ter a capacidade de monitorar *switches*, roteadores e servidores. Portanto, é necessário monitorar objetos de gerência que são relacionados às camadas de enlace, de rede e de transporte, que estão fortemente ligadas ao funcionamento de *switches*, roteadores e servidores, respectivamente. Os objetos SNMP que trazem estatísticas da

camada de enlace pertencem ao grupo *interface*, os objetos da camada de rede pertencem ao grupo *ip*, e os objetos da camada de transporte pertencem aos grupos *tcp* e *udp*. Dentro de cada um destes grupos, foram priorizados os objetos que contabilizam, em diferentes situações, os PDUs (*Protocol Data Unit*) manipulados pelo elemento de rede monitorado. As quatro listagens a seguir trazem estes objetos e suas definições.

Na primeira lista são apresentados os objetos escolhidos do grupo *interface*:

- *ifInOctets*: quantidade total de bytes recebidos pela interface;
- *ifInUcastPkts*: quantidade total de pacotes de *unicast* entregues para as camadas superiores;
- *ifInNUcastPkts*: quantidade total de pacotes de *broadcast* e *multicast* entregues para as camadas superiores;
- *ifInDiscards*: quantidade de pacotes ingressantes descartados mesmo sem a presença de erros;
- *ifInErrors*: quantidade de pacotes ingressantes que contém erros;
- *ifInUnknownProtos*: quantidade de pacotes ingressantes descartados por estarem relacionados a um protocolo não suportado;
- *ifOutOctets*: quantidade total de bytes transmitidos pela interface;
- *ifOutUcastPkts*: quantidade de pacotes de *unicast* transmitidos;
- *ifOutNUcastPkts*: quantidade de pacotes de *broadcast* e *multicast* transmitidos;
- *ifOutDiscards*: quantidade de pacotes descartados antes de serem transmitidos mesmo sem conterem erros.
- *ifOutErrors*: quantidade de pacotes egressos que contém erros;

Os objetos do grupo *ip* são apresentados na segunda lista:

- *ipInReceives*: quantidade total de datagramas IP recebidos pelo dispositivo de rede, incluindo os datagramas com erro;
- *ipInHdrErrors*: quantidade total de datagramas descartados devido a erros no cabeçalho IP;
- *ipInAddrErrors*: quantidade total de datagramas descartados porque o endereço IP de destino contido no cabeçalho não era válido;

- *ipForwDatagrams*: quantidade de datagramas IP ingressantes que foram encaminhados ou roteados para outro destino.
- *ipInUnknownProtos*: quantidade de datagramas IP descartados por causa de um protocolo desconhecido ou não suportado;
- *ipInDiscards*: quantidade de datagramas IP que foram descartados mesmo sem apresentar erros. Deve ser observado que este contador não contabiliza os datagramas descartados enquanto esperavam a remontagem.
- *ipInDelivers*: quantidade de datagramas IP entregues com sucesso aos protocolos clientes do protocolo IP.
- *ipOutRequests*: quantidade de datagramas IP que saíram do dispositivo, sem contabilizar os datagramas encaminhados;
- *ipOutDiscards*: quantidade de datagramas IP que foram descartados na saída mesmo sem apresentar qualquer erro;
- *ipOutNoRoutes*: quantidade de datagramas IP descartados porque não foram encontradas rotas para sua transmissão ao destino.
- *ipReasmReqds*: quantidade de fragmentos IP recebidos que deveriam ser remontados no elemento de rede analisado;
- *ipReasmFails*: quantidade de falhas detectadas pelo algoritmo de remontagem.
- *ipFragOKs*: quantidade de datagramas IP que foram fragmentados com sucesso neste elemento de rede
- *ipFragFails*: quantidade de datagramas IP para os quais não foi possível aplicar a fragmentação;
- *ipFragCreates*: quantidade de fragmentos gerados no elemento de rede analisado;.

A terceira lista traz objetos do grupo *tcp*:

- *tcpInSegs*: quantidade de segmentos TCP recebidos, incluindo aqueles que apresentam erros.
- *tcpOutSegs*: quantidade de segmentos TCP enviados, excluindo aqueles que contém somente dados retransmitidos.

- *tcpRetransSegs*: quantidade de segmentos que contém um ou mais bytes que estão sendo retransmitidos.
- *tcpInErrs*: quantidade de segmentos recebidos que contém erros.
- *tcpOutRsts*: quantidade de segmentos enviados contendo o *flag RST*.

A última lista apresentada traz objetos do grupo *udp*:

- *udpInDatagrams*: quantidade total de datagramas UDP recebidos e entregues à camada superior com sucesso;
- *udpNoPorts*: quantidade de datagramas UDP recebidos para os quais não existia aplicação na porta de destino;
- *udpInErrors*: quantidade de datagramas UDP recebidos e não entregues por razões diferentes da ausência de aplicação na porta de destino;
- *udpOutDatagrams*: quantidade de datagramas enviados pelo elemento de rede;

Para atingir o segundo objetivo, de identificar as relações entre os objetos SNMP, é utilizada uma ferramenta denominada diagrama de Case, proposta por Case e Patridge (1989). Estes diagramas oferecem um eficiente instrumento de análise do posicionamento dos objetos SNMP em relação ao fluxo de dados que atravessa as camadas do elemento de rede. A partir deste posicionamento, é possível também levantar os relacionamentos entre os objetos estudados. A Figura 4.6 traz o diagrama de Case para o grupo *interface*, a Figura 4.7 apresenta o diagrama de Case para o grupo *ip*, o diagrama de Case para o grupo *tcp* é apresentado na Figura 4.8 e, por fim, a Figura 4.9 mostra o diagrama de Case para o grupo *udp*.

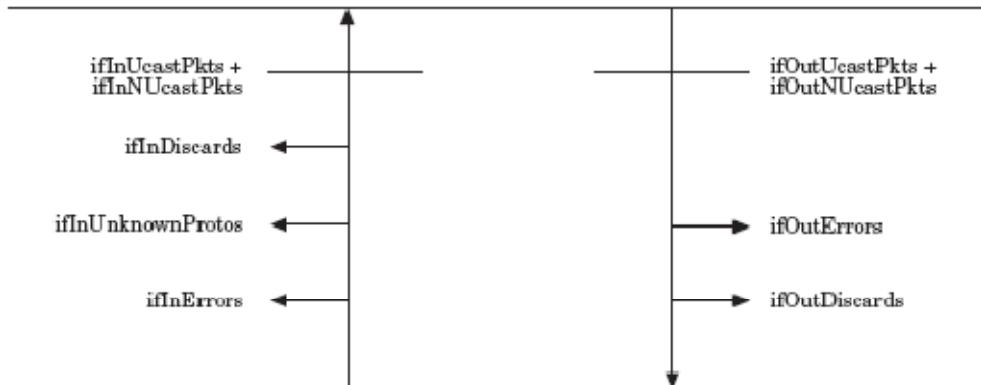


Figura 4.6 - Diagrama de Case para o grupo *interface*.



Figura 4.7 - Diagrama de Case para o grupo *ip*.

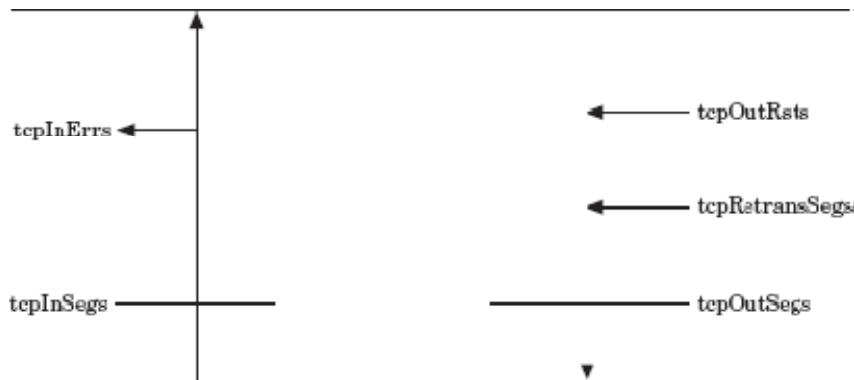


Figura 4.8 - Diagrama de Case para o grupo *tcp*.

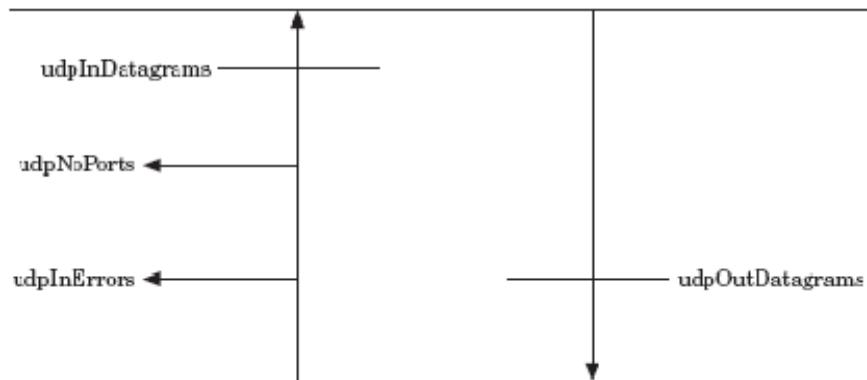


Figura 4.9 - Diagrama de Case para o grupo *udp*.

Os diagramas de Case possuem três tipos de setas horizontais:

1. Setas de subtração, que possuem orientação para a esquerda e representam objetos que contabilizam PDU's que estão sendo retirados do fluxo. Por exemplo, temos o objeto *udpNoPorts* na Figura 4.9;
2. Setas de adição, que possuem orientação para a direita e representam objetos que contabilizam PDU's que estão sendo adicionados ao fluxo. Por exemplo, temos o objeto *ipReasmOKs* na Figura 4.7;
3. Setas de filtro, que não possuem orientação e representam objetos que apenas realizam a contagem de PDU's nas transições entre as camadas. Por exemplo, temos o objeto *tcpInSegs* na Figura 4.8.

Para aplicação do diagrama de Case em nosso trabalho, foi realizada também uma classificação das setas verticais, que não está presente na definição original do diagrama: há as setas verticais que representam os fluxos principais de dados e as setas que representam fluxos secundários, originadas de desvios em relação aos fluxos principais. Na Figura 4.7, que traz o diagrama de Case para a camada de rede do protocolo TCP/IP, podem ser notados dois fluxos de dados principais, representados pelas duas setas verticais maiores. Na área onde temos os objetos *ipReasmReqds*, *ipReasmFails* e *ipReasmOks*, nota-se a presença de uma seta vertical menor, que define um fluxo de dados secundário.

#### 4.3.2.2 Grafo de Dependências

A partir dos diagramas de Case, é proposta a construção de um grafo de dependências entre objetos SNMP. O objetivo é representar os relacionamentos entre os objetos analisados, permitindo a correlação dos alarmes de primeiro nível. O modelo de correlação apresentado nesta seção é todo baseado no grafo de dependências dos objetos, que constitui, portanto, o núcleo do Módulo de Correlação.

Um grafo  $G$  é uma estrutura de dados definida por  $G=(V,A)$ , onde  $V$  representa o conjunto de vértices do grafo, e  $A$  o conjunto de arestas que interligam os vértices respeitando uma determinada relação entre eles. Cada objeto SNMP analisado é representado por um vértice. O relacionamento entre dois objetos é representado por uma aresta. O grafo de dependências do Módulo de Correlação é um grafo direcionado. Nesta categoria de grafos, as arestas exprimem uma relação unidirecional entre dois vértices e são

representadas por pares ordenados  $(x, y)$ . No grafo de dependências proposto, um par ordenado  $(x, y)$  define que uma anomalia pode se propagar do objeto SNMP representado pelo vértice  $x$  ao objeto SNMP representado pelo vértice  $y$ .

Na Tabela 4.3 é apresentado um algoritmo proposto, que realiza a tradução de diagramas de Case para grafos de dependência. O algoritmo tem como principal objetivo sistematizar a criação do grafo de dependências a partir dos diagramas de Case.

No algoritmo, primeiramente, são analisadas as setas verticais que definem os fluxos principais de dados. Estas setas são percorridas respeitando a sua direção, e cada uma das setas horizontais é tratada de acordo com a sua classificação: adição, subtração ou filtro. Conforme as setas horizontais são encontradas, os vértices e arestas do grafo de correlação vão sendo criados. Após analisar os fluxos principais, segue a análise dos fluxos secundários, também definidos no diagrama por setas verticais. Como todo fluxo secundário se inicia a partir de uma seta horizontal, quando a análise destes fluxos é iniciada, assume-se que a última seta de filtro encontrada é a seta que deu origem ao fluxo secundário, mesmo que esta seta represente uma subtração para o fluxo principal onde ela originou-se.

Tabela 4.3 - Algoritmo para tradução dos diagramas de Case em grafos de dependências.

---

#### ALGORITMO: TRANFORMAR DIAGRAMA DE CASE EM GRAFO DE DEPENDÊNCIAS

---

**Entrada:** Diagrama de Case;

**Saída:** Grafo de Dependências;

##### Notação

`tipo(seta_horizontal)`: função que retorna o tipo da seta horizontal passada como parâmetro, se ela é uma seta horizontal de adição, subtração ou filtro;

`criarOuBuscarVertice(seta_horizontal)`: função que busca vértice criado para representar uma seta horizontal e o retorna. No caso de não encontrar cria um novo vértice;

`coletarVérticeCorrespondente(seta_horizontal)`: função que busca vértice que corresponde a determinada seta horizontal passada como parâmetro;

`coletarVérticesCorrespondentes(conjunto de setas horizontais)`: função que busca conjunto de vértices que correspondem a determinado conjunto de setas horizontais passado como parâmetro;

`criarArestaDirecionada(vértice x, vértice y)`: função que cria aresta direcionada partindo do vértice  $x$  para o vértice  $y$ ;

`criarArestasDirecionadas(conjunto de vértices x, vértice y)`: função que cria

---

---

aresta direcionada partindo de cada vértice contido no conjunto x para o vértice y;

fimSetaVertical(seta vertical): função que responde se a seta vertical já foi inteiramente analisada;

tratarSetaFiltroSubtração(seta horizontal): função que dispara a criação de vértices e arestas no grafo para a seta horizontal do tipo filtro ou subtração recebida;

tratarSetaAdição(seta horizontal): função que dispara a criação de vértices e arestas no grafo para a seta horizontal do tipo adição recebida;

proximaSetaEncontrada: variável que armazena a próxima seta encontrada que deve ser analisada no momento;

conjuntoUltimasSetasAdição: variável que armazena o conjunto com as últimas setas horizontais de adição encontradas, desde a última vez em que se encontrou uma seta de filtro;

ultimaSetaFiltro: variável que armazena a última seta de filtro encontrada;

setaHorizontalInicioFluxoSecundario: variável que armazena a seta que inicia um fluxo secundário.

---

```
01. /*programa principal*/
02. ENQUANTO existem setas verticais
03.     SE fluxoPrincipal ENTÃO
04.         setaHorizontal := proximaSetaEncontrada;
05.         FAÇA
06.             SE (tipo(setaHorizontal) = adição) ENTÃO
07.                 tratarSetaAdição(setaHorizontal);
08.                 conjuntoUltimasSetasAdição := setaHorizontal;
09.             SE (tipo(setaHorizontal) = filtro) ENTÃO
10.                 tratarSetaFiltroSubtração(setaHorizontal);
11.                 ultimaSetaFiltro := setaHorizontal;
12.                 conjuntoUltimasSetasAdição := null;
13.             SE (tipo(setaHorizontal) = subtração) ENTÃO
14.                 tratarSetaFiltroSubtração(setaHorizontal);
15.             ENQUANTO NÃO (fimSetaVertical(fluxoPrincipal));
16.             SENÃO
17.                 SE fluxoPrincipal já analisado ENTÃO
18.                     //a seta horizontal que dá inicio ao fluxo secundário
19.                     /*trabalha como uma seta de filtro até que uma próxima seta
20.                      * de filtro seja encontrada*/
21.                     ultimaSetaFiltro := setaHorizontalInicioFluxoSecundario;
22.                     FAÇA
23.                         setaHorizontal := proximaSetaEncontrada;
24.                         SE (tipo(setaHorizontal) = adição) ENTÃO
25.                             tratarSetaAdição(setaHorizontal);
```

---

---

```

26.           conjuntoUltimasSetasAdição := setaHorizontal;
27.           SE (tipo(setaHorizontal) = filtro) ENTÃO
28.               tratarSetaFiltroSubtração(setaHorizontal);
29.               ultimaSetaFiltro := setaHorizontal;
30.               conjuntoUltimasSetasAdição := nulo;
31.               SE (tipo(setaHorizontal) = subtração) ENTÃO
32.                   tratarSetaFiltroSubtração(setaHorizontal);
33.               ENQUANTO NÃO(fimSetaVertical(fluxoSecundário));


---


34. /*rotina de tratamento das setas encontradas no diagrama de Case*/
35. PROCEDIMENTO tratarSetaFiltroSubtração (setaHorizontal)
36.     //vêrtice para objeto representado pela seta de adição;
37.     v1 := criarOuBuscarVertice(setaHorizontal);
38.     v2 := coletarVerticeCorrespondente(ultimaSetaFiltro);
39.     v3 := coletarVerticesCorrespondentes(conjuntoUltimasSetasAdição);
40.     criarArestaDirecionada(v2, v1);
41.     criarArestasDirecionadas(v3, v1);
42. FIM PROCEDIMENTO;


---


43. /*rotina de tratamento das setas encontradas no diagrama de Case*/
44. PROCEDIMENTO tratarSetaAdição (setaHorizontal)
45.     //vêrtice para objeto representado pela seta de adição;
46.     v1 := criarOuBuscarVertice(setaHorizontal);
47. FIM PROCEDIMENTO;

```

---

O grafo de dependências resultante da tradução dos diagramas de Case para os grupos de objetos *interface*, *ip*, *tcp* e *udp* é apresentado na Figura 4.10. Ele reúne em seu escopo os possíveis caminhos de propagação das anomalias ao longo dos objetos SNMP, mapeando os possíveis comportamentos delas no elemento de rede analisado. Quando o Módulo de Correlação recebe os alarmes de primeiro nível, ele utiliza o grafo de dependências para verificar se a anomalia está ocorrendo ou não. Quando a anomalia é detectada, o caminho de propagação da anomalia pelo grafo de dependências é utilizado para definir um mapa de comportamento da anomalia dentro do dispositivo de rede. Este diagnóstico é oferecido como informação adicional contida no alarme de segundo nível, que auxiliará na construção do mapa de propagação do problema em toda a rede.

A análise mais profunda dos possíveis caminhos de propagação de uma anomalia em um elemento de rede se inicia no levantamento dos possíveis comportamentos de um elemento de rede que esteja participando de uma anomalia. Basicamente, são três os comportamentos possíveis:

- Destino do tráfego anômalo: o elemento de rede está sendo atacado ou recebendo um volume anormal de requisições. Um servidor Web que esteja atravessando uma anomalia do tipo *flash crowd*, por exemplo.
- Origem do tráfego anômalo: o elemento de rede é o ponto de origem do volume anormal de pacotes ou requisições. Um computador que tenha sido dominado e esteja sendo usado para atacar outros computadores vai ser a origem de uma anomalia.
- Encaminhando o tráfego anômalo: um roteador ou um firewall pode receber tráfego anômalo e encaminhar para outros elementos da rede.

A partir destas três possíveis situações para o comportamento de um elemento de rede perante uma anomalia, são identificados três diferentes tipos de fluxos de dados que podem apresentar anomalias nos elementos de rede: fluxo de entrada, fluxo de saída e fluxo de encaminhamento. Na Figura 4.10, os objetos são classificados segundo estes fluxos. A Figura 4.11 mostra como estes diferentes fluxos de dados se comportam em relação às camadas do conjunto de protocolos TCP/IP. O *fluxo de entrada* atravessa todas as camadas até alcançar a camada de aplicação, onde os dados transportados serão utilizados. O *fluxo de saída* segue o caminho inverso, tendo sua origem na camada de aplicação, que está enviando os dados para a rede. O *fluxo de encaminhamento* chega até a camada de rede, onde é definido o encaminhamento dos dados para outro ponto da rede, em um processo característico de equipamentos que realizam roteamento. Esta divisão por fluxos será utilizada principalmente para classificar o comportamento da anomalia no elemento de rede. Se a anomalia é detectada no fluxo de entrada, o elemento de rede está recebendo tráfego anômalo, comportando-se como destino deste tráfego. Se a anomalia é detectada no fluxo de saída, o elemento de rede está se comportando como origem de tráfego anômalo. Por fim, se detectarmos anomalias no fluxo de encaminhamento, saberemos que o elemento está encaminhando um tráfego anômalo, que tem origem e destino interconectados com o elemento sob análise.

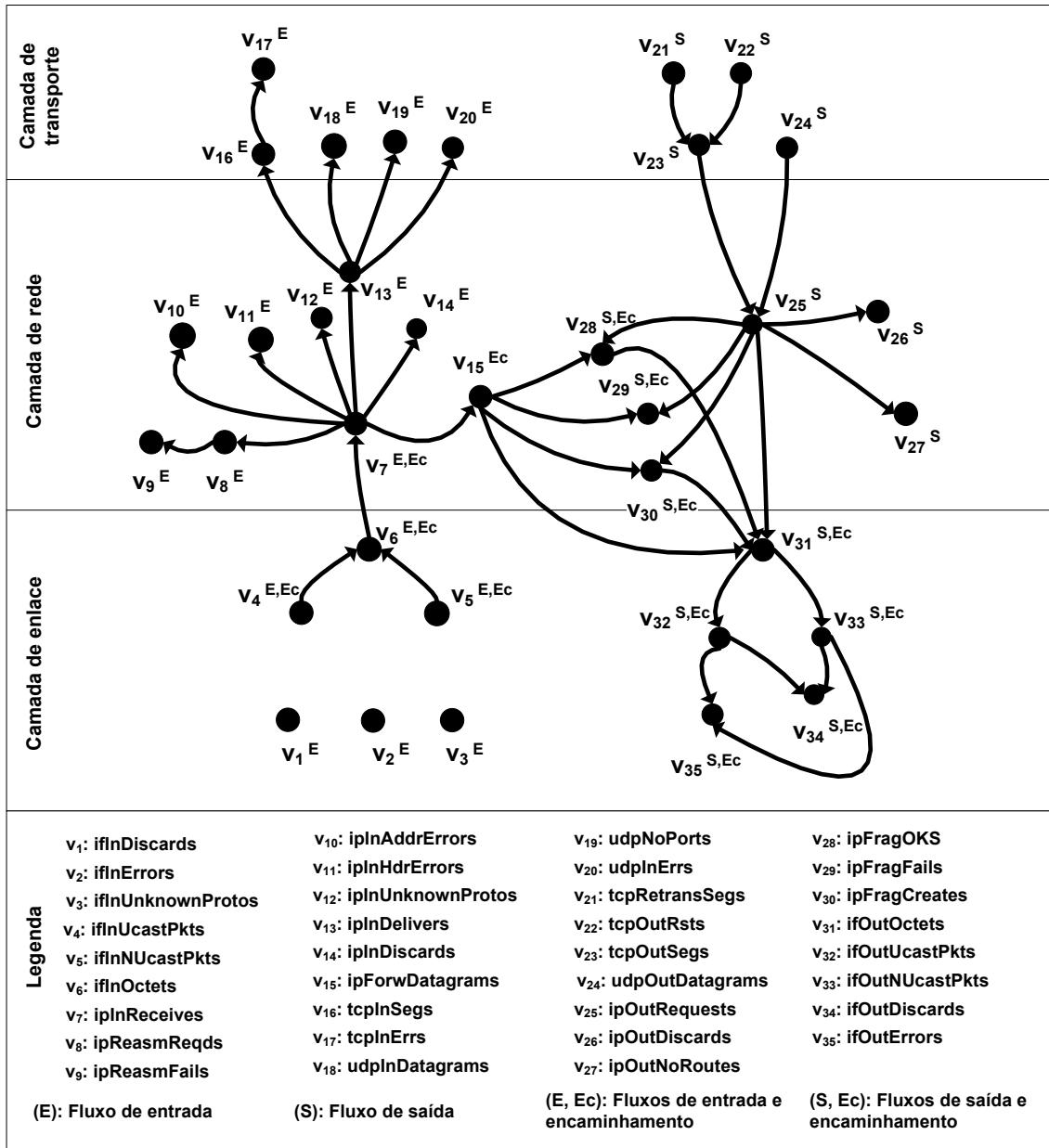


Figura 4.10 - Grafo de dependências.

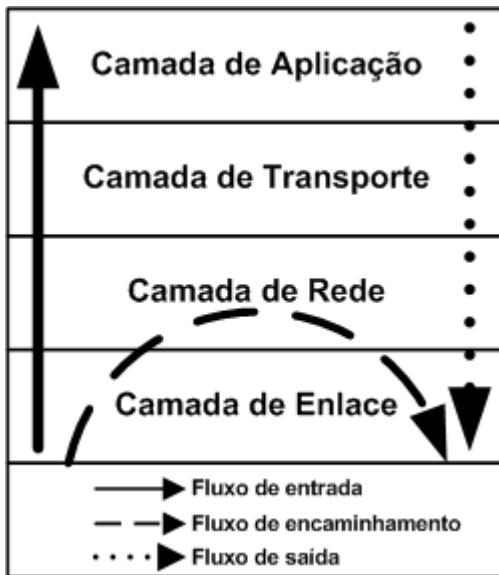


Figura 4.11 - As camadas do protocolo TCP/IP e os fluxos de dados.

#### 4.3.2.3 Algoritmo de Correlação

O objetivo do algoritmo apresentado nesta seção é verificar a ocorrência da anomalia e mapear a sua propagação no elemento de rede monitorado, através da correlação dos alarmes de primeiro nível. A correlação será baseada no grafo de dependências, que traz as relações entre os objetos SNMP e o posicionamento destes objetos com relação aos fluxos de dados que percorrem o elemento de rede.

Com base na direção de propagação destes fluxos e no posicionamento dos objetos, podemos definir alguns objetos como pontos iniciais e outros como pontos finais dos possíveis caminhos de propagação das anomalias. Na correlação dos alarmes de primeiro nível, um alarme de segundo nível só é gerado caso os objetos relacionados aos alarmes de primeiro nível formem um caminho no grafo de dependências que se inicie em um objeto definido como inicial e se encerre em um objeto definido como final. Assim sendo, nossa proposta exige que haja um indício concreto de ocorrência da anomalia para que o alarme de segundo nível seja criado: a propagação de tráfego anômalo por caminhos pré-definidos completos de propagação de anomalias.

A Figura 4.12 traz as possibilidades de propagação da anomalia no fluxo de entrada e os respectivos objetos iniciais e finais para este fluxo. Os objetos iniciais são o *ifInUcastPkts* e o *ifNUcastPkts*, que monitoram a entrada de *bytes* em uma determinada

interface do dispositivo. A partir daí, o tráfego anômalo pode se propagar de duas formas. Na primeira, o elemento de rede processa os pacotes com poucos erros ou descartes e a anomalia reflete no caminho composto pelos objetos *ipInReceives*, *ipInDelivers* e *tcpInSegs* ou *udpInDatagrams*. Estes dois últimos objetos são considerados como pontos finais. Há também a possibilidade do tráfego anômalo conter grandes quantidades de pacotes com erros ou o elemento já estar passando por sobrecarga, descartando os pacotes. Neste caso, a propagação da anomalia pode se encerrar já na camada de rede, refletindo no *ipInReceives* e em objetos de erros como o *ipInAddrErrors* ou *ipInDiscards*, que são os objetos finais. O tráfego anômalo pode se propagar até a camada de transporte onde apresentaria os erros, tendo objetos relacionados a erros como o *tcpInErrs* e *udpNoPorts* como objetos finais.

A Figura 4.13 traz as possibilidades de propagação de anomalias para o fluxo de saída, com seus respectivos objetos iniciais e finais. Os objetos iniciais para a propagação da anomalia pelo caminho principal do fluxo de saída são o *tcpOutsSegs* e o *udpOutDatagrams*. O *tcpRetransSegs* e o *tcpOutRsts* são considerados pontos iniciais também, mas refletem apenas anomalias onde há muitos pacotes retransmitidos ou contendo o flag RST. No caso de não haver erros ou descartes em excesso durante o processamento do tráfego anômalo, os objetos finais são *ifOutUCastPkts* e *ifOutNUcastPkts*, que monitoram os pacotes entregues à rede. Caso haja dificuldade da interface em entregar estes pacotes, por sobrecarga ou outros erros, a anomalia vai se refletir em objetos relacionados a erros e descartes. Para contemplar estas situações, os objetos *ifOutDiscards* e *ifOutErrors* são definidos como pontos finais. A propagação da anomalia no fluxo de saída pode se encerrar ainda na camada de rede, caso haja erros de fragmentação ou descarte de pacotes. Estes casos são atendidos pela definição dos objetos *ipOutDiscards*, *ipOutNoRoutes* e *ipFragFails* como objetos finais.

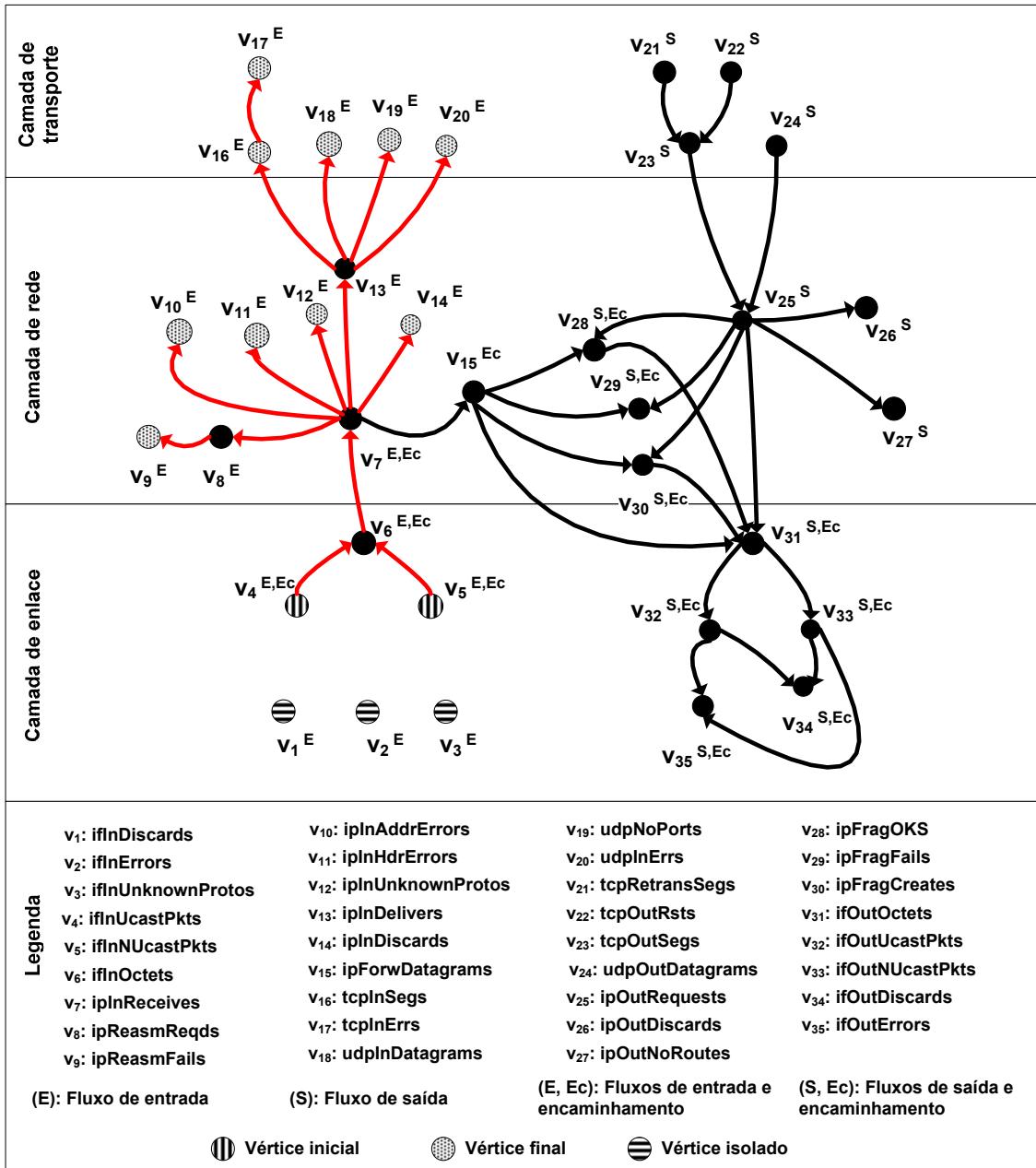


Figura 4.12 - Caminhos de propagação de anomalias no fluxo de entrada do grafo de dependências.

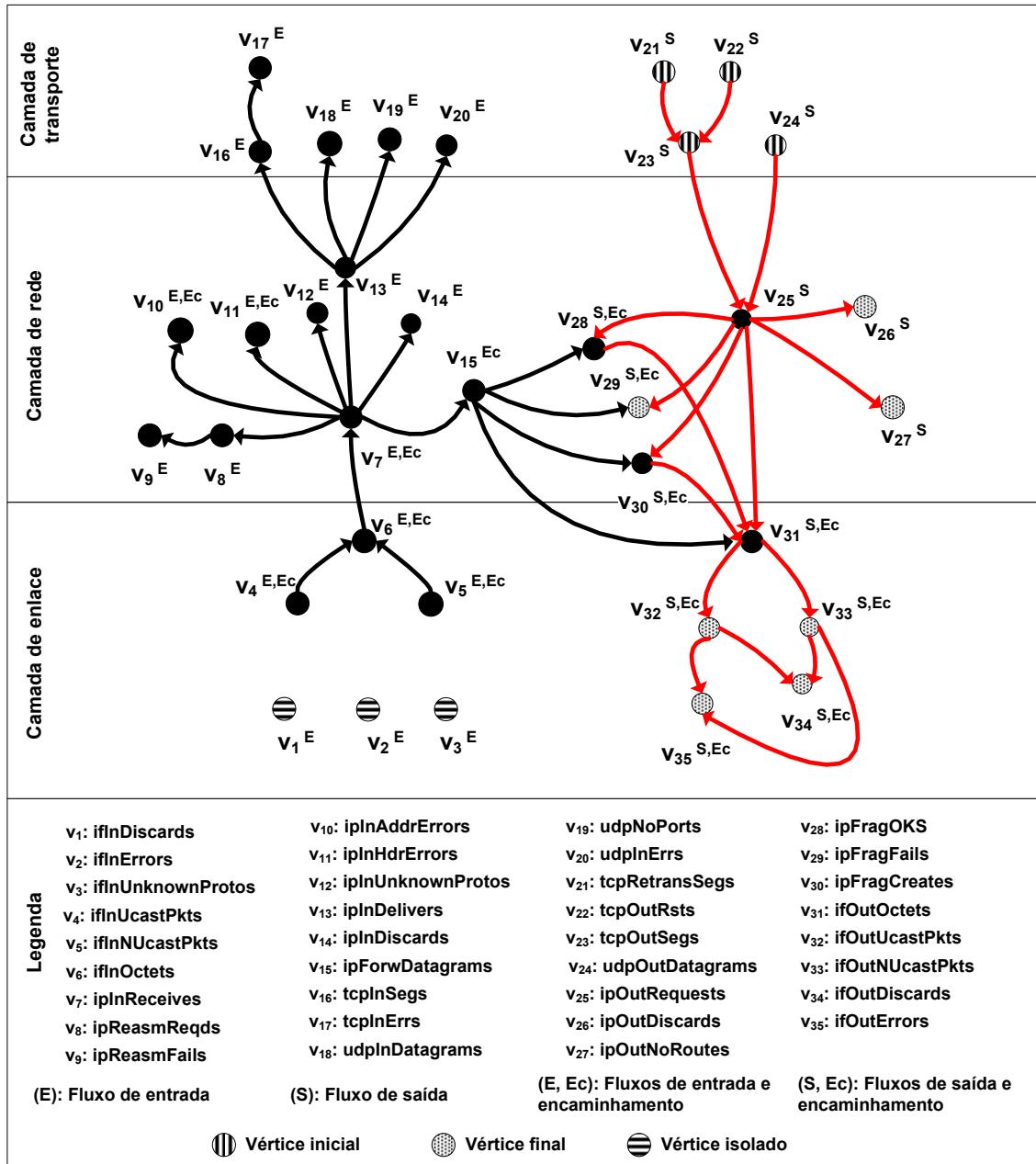


Figura 4.13 - Caminhos de propagação de anomalias no fluxo de saída do grafo de dependências.

A Figura 4.14 traz em destaque as possibilidades de propagação da anomalia no fluxo de encaminhamento e os objetos iniciais e finais. Neste caso não há envolvimento da camada de transporte. Os objetos iniciais são os mesmos do fluxo de entrada. Caso o processamento do tráfego anômalo não traga erros ou descartes de pacotes em excesso, os objetos finais são o *ifOutUcastPkts* e *ifOutNUCastPkts*. Caso existam muitos pacotes com

erros de cabeçalho ou endereços IP inválidos, o comportamento dos objetos *ipInHdrErrors* e *ipInAddrErrors* serão afetados, por isso, eles são considerados objetos finais. Outro objeto considerado final na camada de rede está relacionado a erros na fragmentação de pacotes, o *ipFragFails*. Por fim, temos os objetos finais que contemplam as situações de erro na camada de enlace: *ifOutDiscards* e *ifOutErrors*.

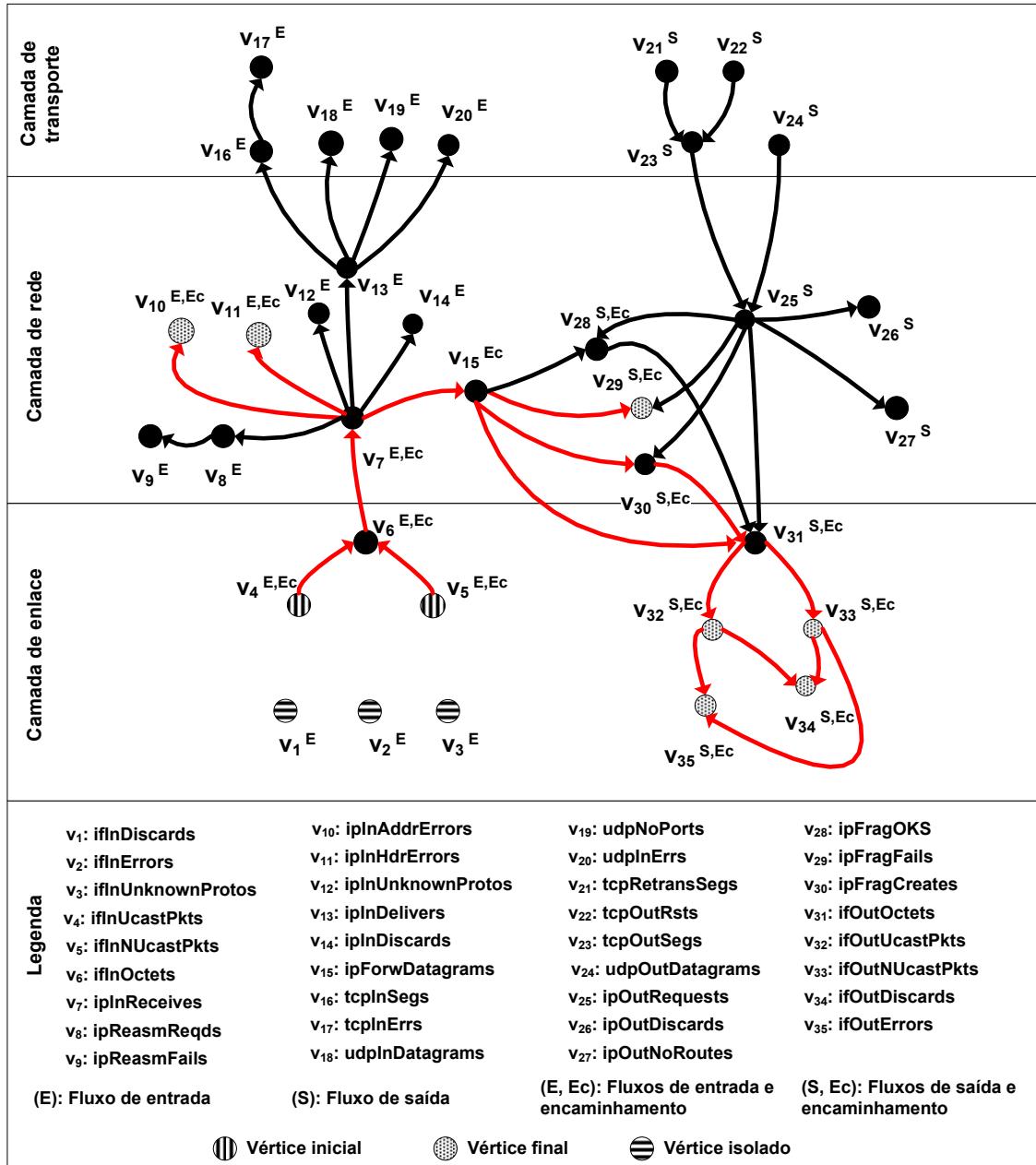


Figura 4.14 - Caminhos de propagação de anomalias no fluxo de encaminhamento do grafo de dependências.

O estudo do posicionamento dos objetos SNMP e dos possíveis caminhos de propagação das anomalias levou à definição da estratégia de correlação baseada nas relações de dependência entre os objetos. Outra questão a ser tratada é a estratégia de análise dos alarmes de primeiro nível segundo o momento em que eles foram gerados. No algoritmo proposto é utilizada a técnica baseada em janelas (STEINDER; SETHI, 2004b). O monitoramento de cada objeto SNMP é dividido em janelas fixas de 5 minutos, denominadas janelas de correlação. O tempo de 5 minutos foi escolhido por ser normalmente utilizado em outros sistemas de detecção de anomalias como os propostos por Barford et al. (2002), Soule et al. (2005) e Thottan e Ji (2003). Todos os alarmes de primeiro nível gerados na mesma janela de correlação são analisados conjuntamente. Em suma, a janela de correlação tem como seu maior objetivo estabelecer um referencial de tempo para a execução da correlação no Módulo de Correlação.

O algoritmo de correlação, que reúne os alarmes do primeiro nível e verifica a ocorrência da anomalia por meio do grafo de dependências, foi construído com base no algoritmo de busca em profundidade em grafos (GERSTING, 2002). A diferença é que, enquanto no algoritmo de busca em profundidade caminha-se pelo grafo por meio de vértices adjacentes, no algoritmo empregado no Módulo de Correlação caminha-se pelo grafo por meio de vértices correlacionados. Dois vértices são considerados correlacionados quando são adjacentes e há alarmes gerados para ambos os objetos SNMP na mesma janela de tempo de 5 minutos.

O algoritmo do Módulo de Correlação é apresentado na Tabela 4.4. O procedimento `buscaProfundidade()` é recursivo e encarregado de percorrer o grafo de dependências, buscando o caminho seguido pela suposta anomalia, verificando sua ocorrência e preparando o mapa a ser apresentado ao administrador de rede. É importante lembrar que a situação é considerada anômala quando o processo de busca sobre o grafo de dependências parte de um objeto inicial e alcança um objeto final.

Tabela 4.4 - Algoritmo de busca em profundidade para o Módulo de Correlação.

---

### ALGORITMO: BUSCA EM PROFUNDIDADE

---

**Entrada:** alarmes de primeiro nível;

**Saída:** alarmes de segundo nível;

**Notação:**

$O_i$ : conjunto de objetos definidos como iniciais;

$O_f$ : conjunto de objetos definidos como finais;

$O_a$ : conjunto de objetos que apresentaram alarme de primeiro nível na mesma janela de cinco minutos;

*Pilha*: pilha utilizada na busca em profundidade

$C(o)$ : função que retorna todos os objetos que são adjacentes ao objeto  $o$  e possuem alarmes de primeiro nível gerados na mesma janela de cinco minutos que o objeto  $o$ ;

PROGRAMA PRINCIPAL

```

01. INICIO
02.     PARA cada  $o \in (O_i \cap O_a)$  FAÇA
03.         buscaProfundidade( $o$ );
04.     FIM PROGRAMA PRINCIPAL;
=====
05. PROCEDIMENTO buscaProfundidade( $o$ )
06. INICIO
07.     marcar  $o$  como visitado;
08.     empilhar  $o$  em Pilha;
09.     SE ( $o \in O_f$ ) ENTÃO
10.         anomalia detectada;
11.         PARA cada  $(o' \in C(o))$  FAÇA
12.             INICIO
13.                 SE  $o'$  não está marcado ENTÃO
14.                     buscaProfundidade( $o'$ );
15.                 FIM PARA;
16.             desempilhar Pilha;
17.         FIM PROCEDIMENTO;
```

---

## 4.4 Módulo de Configuração Automática

Administradores de redes podem apresentar opiniões distintas sobre quais desvios de comportamento devem ser considerados anomalias. Um primeiro grupo de administradores pode estar interessado em detectar até mesmo os menores desvios, de forma a coibir e detectar qualquer indício de mau uso dos recursos disponíveis na rede. Um segundo grupo pode desejar detectar apenas desvios maiores, que realmente apresentem fortes impactos na qualidade dos serviços prestados. A correta configuração dos parâmetros de sensibilidade do sistema de detecção pode fazer com que ele opere de acordo com a

política de gerência do administrador. Conforme os valores dos parâmetros são alterados, a sensibilidade do sistema é trabalhada, modificando as taxas de detecção, de falsos positivos e as situações consideradas como anomalias. Porém, não é possível delegar esta tarefa de calibragem dos parâmetros aos administradores, já que as redes atuais formam sistemas complexos, compostos por vários equipamentos, onde cada um deles apresenta uma série de objetos de gerência para serem monitorados. Configurar o Módulo de Detecção de Anomalias para cada elemento a ponto de fazê-lo operar dentro da política de gerência desejada para o administrador demandaria muito tempo. Como solução para este problema, apresentamos nesta seção uma proposta desenvolvida para a configuração automática dos parâmetros das instâncias do Módulo de Análise de Objeto SNMP.

O núcleo da solução reside em analisar diferentes combinações de valores de parâmetros aplicando-os em semanas anteriores de monitoramento para definir qual se ajusta melhor aos objetivos do administrador de redes. Cabe ao administrador da rede informar quais anomalias gostaria de ter detectado, a fim de que o sistema possa identificar qual é o comportamento ideal a ser seguido. Na proposta apresentada, o sistema escolhe os parâmetros buscando o melhor equilíbrio entre duas métricas bastante utilizadas para avaliação de sistemas de detecção: a taxa de falsos positivos e a taxa de detecção. Os alarmes gerados por sistemas de detecção de anomalias podem ser classificados em quatro categorias: falso positivo, positivo verdadeiro, negativo verdadeiro e falso negativo (ESTEVEZ-TAPIADOR et al., 2004). A Figura 4.15 apresenta como esta divisão é feita. O falso positivo ocorre quando o sistema de detecção de anomalias identifica como anômala uma situação normal. O positivo verdadeiro ocorre quando uma situação anômala é identificada corretamente. Na mesma direção, o negativo verdadeiro ocorre quando uma situação normal é identificada como tal. O falso negativo ocorre quando o sistema identifica uma situação como normal quando, de fato, ela se trata de uma anomalia.

Destas classificações, são geradas as duas métricas que utilizaremos para avaliar quais valores devem ser aplicados aos parâmetros de sensibilidade do sistema. A primeira métrica é a taxa de falsos positivos, que é identificada em (4.1) por  $F$ . Ela é calculada a partir da divisão entre a quantidade de alarmes que receberam a classificação de falso positivo pelo total de alarmes gerados.

$$F = \frac{\# \text{alarmes\_falsos\_positivos}}{\# \text{total\_alarmes}} \quad (4.1)$$

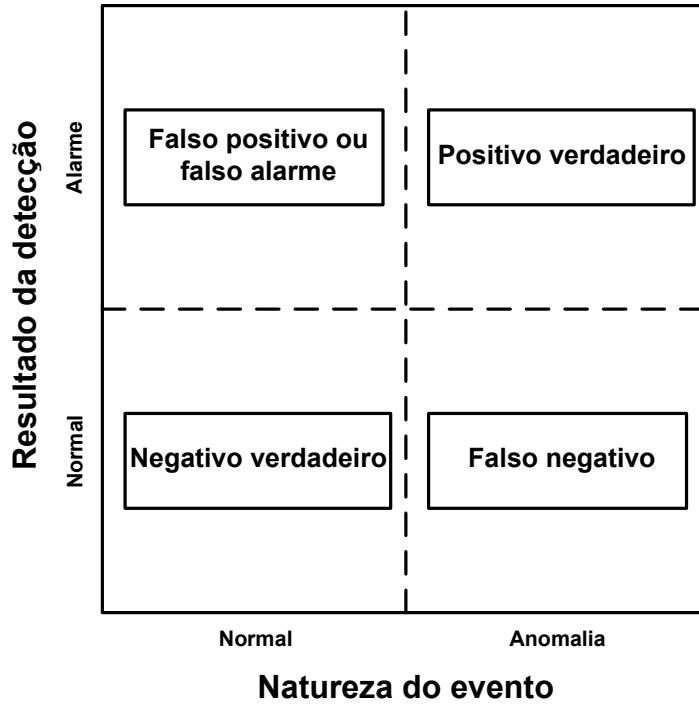


Figura 4.15 - Possíveis classificações para alarmes.

A segunda métrica é a taxa de detecção, identificada por  $T$  em (4.2). Ela é calculada a partir da divisão da quantidade de anomalias detectadas por positivos verdadeiros pelo total de anomalias ocorridas.

$$T = \frac{\# \text{anomalias\_detectadas}}{\# \text{total\_anomalias}} \quad (4.2)$$

Para avaliar qual é o melhor conjunto de valores para os parâmetros, utilizamos uma métrica denominada eficiência ( $E$ ), apresentada em (4.3), que soma a taxa de detecção ( $T$ ) e o complemento da taxa de falsos positivos ( $1-F$ ). O pior valor que pode ser obtido para  $E$  é  $E=0$ . Neste caso, temos  $T=0$ , indicando que nenhuma das anomalias ocorridas foi detectada, e  $F=1$ , indicando que todos os alarmes gerados pelo sistema são falsos positivos. O melhor valor que pode ser obtido para  $E$  é  $E=2$ . Neste caso, temos  $T=1$ , indicando que todas as anomalias ocorridas foram detectadas e  $F=0$ , indicando que nenhum dos alarmes gerados é falso positivo.

$$E = T + (1 - F) \quad (4.3)$$

O Módulo de Análise de Objeto SNMP possui dois parâmetros de sensibilidade: o tamanho do intervalo de histerese e o valor do  $\delta$ . Para estabelecer qual é o melhor conjunto de valores para os parâmetros no monitoramento da semana  $w_n$ , o algoritmo testa a aplicação de diferentes valores de parâmetros nas semanas que vão de  $w_1$  a  $w_{n-1}$ , constituindo um período de treinamento. O algoritmo utiliza os registros de anomalias inseridos pelo administrador de rede para calcular a taxa de detecção e a de falsos positivos para cada conjunto de parâmetros testado no período de treinamento. A configuração que apresentar os melhores resultados para o índice de eficiência é escolhida para ser aplicada na semana  $w_n$ .

A Figura 4.16 detalha os passos de execução do algoritmo. Primeiramente são formados dois conjuntos com os possíveis valores para histerese e  $\delta$ , denominados  $H$  e  $D$ , respectivamente. No passo número 2, estes dois conjuntos são combinados, formando pares com um valor para intervalo de histerese e um valor de  $\delta$ . Estes diferentes pares formam o conjunto  $P$ . No passo 3, são gerados alertas de segundo nível utilizando cada um dos pares de valores definidos em  $P$ . Estes alarmes de segundo nível vão formar o conjunto  $S$ . O quarto passo utiliza o conjunto de anomalias  $L$  inserido pelo administrador de redes para avaliar os diferentes pares de valores para os parâmetros, calculando os valores de eficiência para cada um. O quinto passo classifica os diferentes resultados de eficiência obtidos. No passo 6, os parâmetros que geraram o melhor resultado de eficiência são recolhidos, para que possam ser aplicados na análise futura da rede.

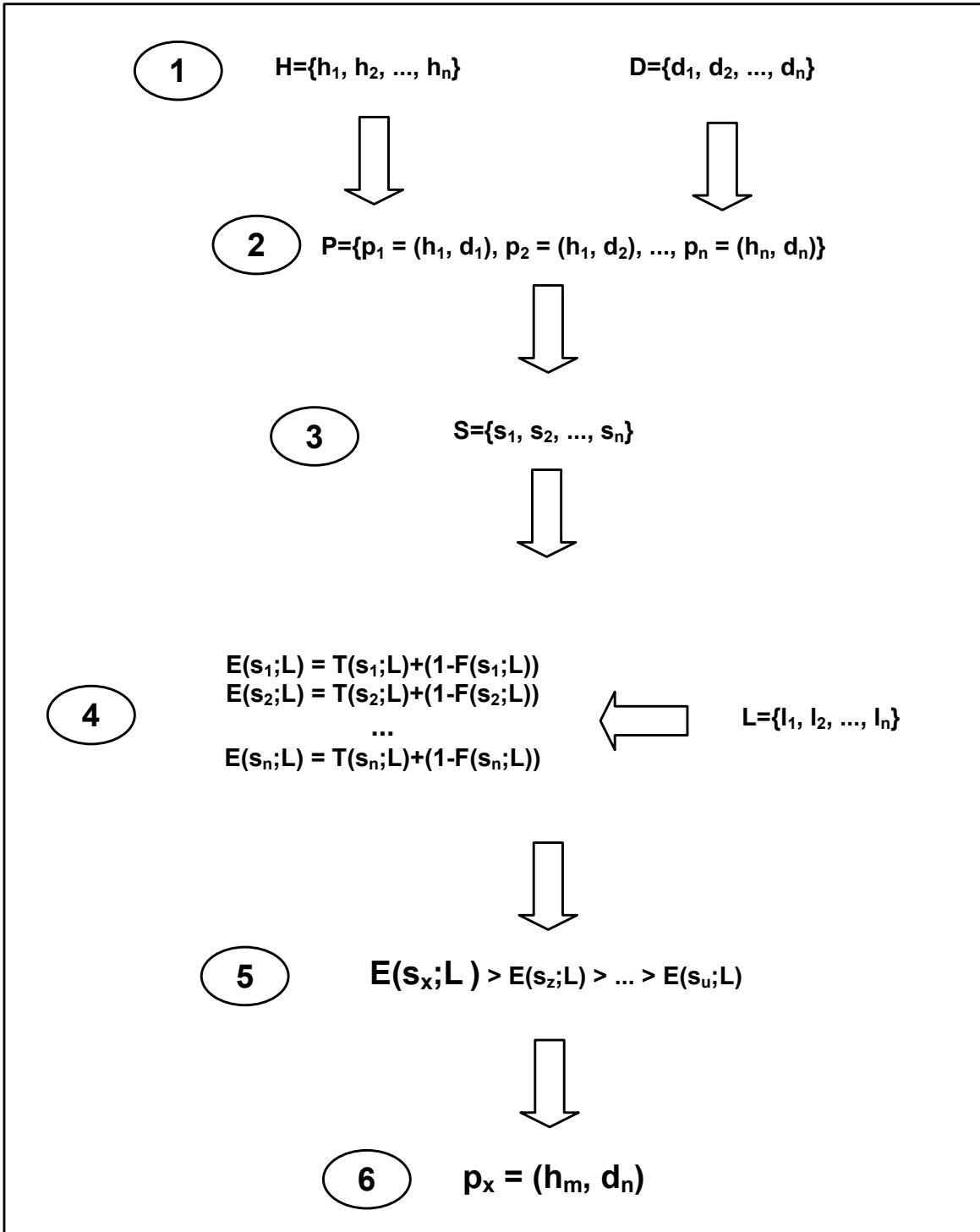


Figura 4.16 - Passos do algoritmo de configuração automática de parâmetros.

#### **4.5 Módulo de Localização de Anomalias**

No passado, as redes de computadores eram mais simples e formadas por poucos elementos. A gerência destas redes era realizada por administradores experientes, que recorriam a ferramentas automatizadas em situações pontuais. Atualmente, para atender a crescente demanda por serviços, as redes têm se tornado sistemas heterogêneos e complexos, impossibilitando uma abordagem de gerência baseada apenas na experiência de administradores humanos (SAAMAN; KARMOUCH, 2008).

Sistemas capazes de analisar informações coletadas da rede e informar o administrador sobre problemas que estejam ocorrendo se tornaram essenciais. Muitas soluções existentes analisam os diferentes elementos de rede um a um, gerando um conjunto muito grande de alarmes, que muitas vezes são redundantes. Em uma situação de ataque a um servidor, por exemplo, os enlaces e dispositivos da rede que fazem parte do caminho de acesso ao dispositivo atacado apresentarão variações de tráfego. Este fato, provavelmente, causará a geração de alarmes para cada um destes elementos, todos reportando o mesmo problema. É necessário, portanto, desenvolver sistemas que sejam capazes de analisar a rede como um todo, reunindo os alertas gerados de maneira isolada para diferentes elementos em um único relatório que traga uma visão panorâmica do problema (LI et al., 2008). Nesta seção, é apresentada a proposta para o Módulo de Localização de Anomalias. Ele será o responsável pelo agrupamento dos alarmes de segundo nível em um alarme de terceiro nível, que será enviado ao administrador e apoiará a busca pela origem e solução do problema.

O Módulo de Localização de Anomalias vai cruzar dois tipos de informações para mapear a propagação da anomalia pela rede:

- 1. Classificação da anomalia presente no alarme de segundo nível:** conforme apresentado na seção 4.3.2, o alarme de segundo nível aponta se a anomalia ocorreu no fluxo de entrada, de saída ou de encaminhamento. Esta classificação será utilizada para determinar o papel de cada elemento de rede na propagação do tráfego anômalo;
- 2. Dados sobre a topologia da rede:** a solução de localização de anomalias proposta recorre a uma abordagem determinística, ou seja, assume que as

dependências entre os elementos de rede são conhecidas. A topologia da rede será modelada em um grafo, no qual cada equipamento é representado como um vértice e as arestas representam os enlaces que conectam estes equipamentos;

A partir do cruzamento entre os dados sobre o comportamento da anomalia em cada elemento de rede e as conexões entre estes elementos, podemos inferir qual é o caminho de propagação da anomalia pela rede.

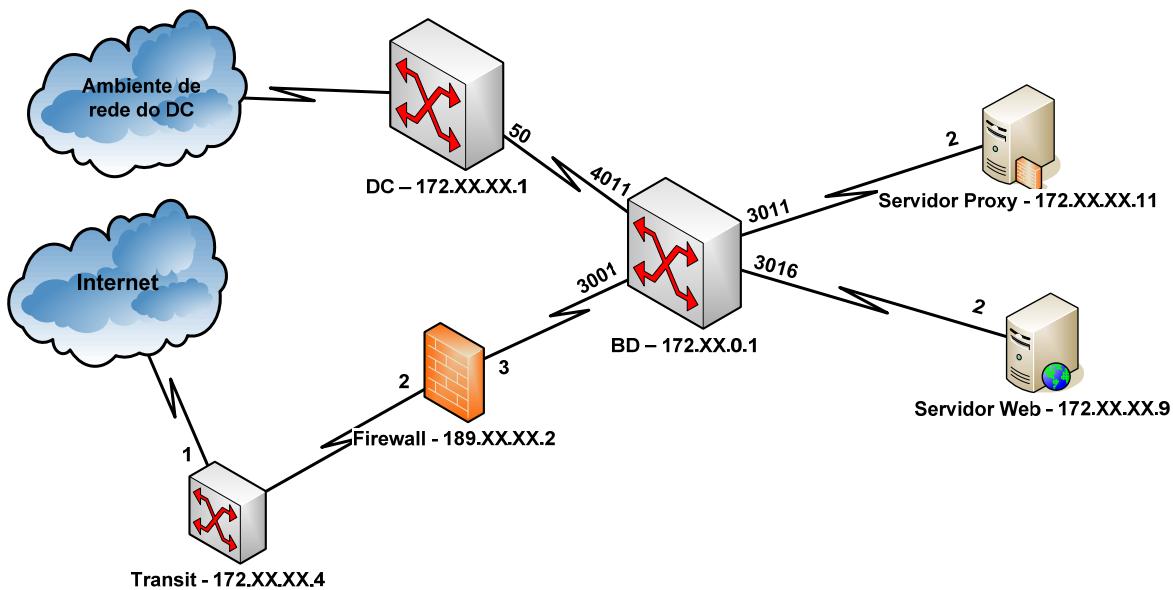


Figura 4.17 - Ambiente de rede monitorado na UEL.

A Figura 4.17 mostra o cenário de rede da UEL para o qual foi construído o grafo da topologia apresentado na Figura 4.18. Cada equipamento é representado como um vértice. Há equipamentos que possuem mais de uma interface, como é o caso do *switch Black Diamond* (endereço IP: 172.XX.0.1) e do *firewall* (endereço IP: 189.XX.XX.2). Por isso, são colocados atributos nas arestas, que identificam quais interfaces são usadas pelo enlace representado pela aresta. Estes atributos são formatados como um par ordenado  $(m,n)$ .  $m$  representa a interface utilizada pelo equipamento que está na origem da aresta, enquanto  $n$  representa a interface utilizada pelo equipamento que está no final da aresta. Como exemplo, temos os equipamentos com endereços IP 172.XX.0.1 e 172.XX.XX.1. Há uma aresta que tem sua origem em 172.XX.0.1 e seu final em 172.XX.XX.1. O atributo desta aresta é (4011,50). Isto indica que a conexão entre estes dois equipamentos é estabelecida

através da interface 4011 do equipamento com endereço IP 172.XX.0.1 e da interface 50 do equipamento com endereço IP 172.XX.XX.1.

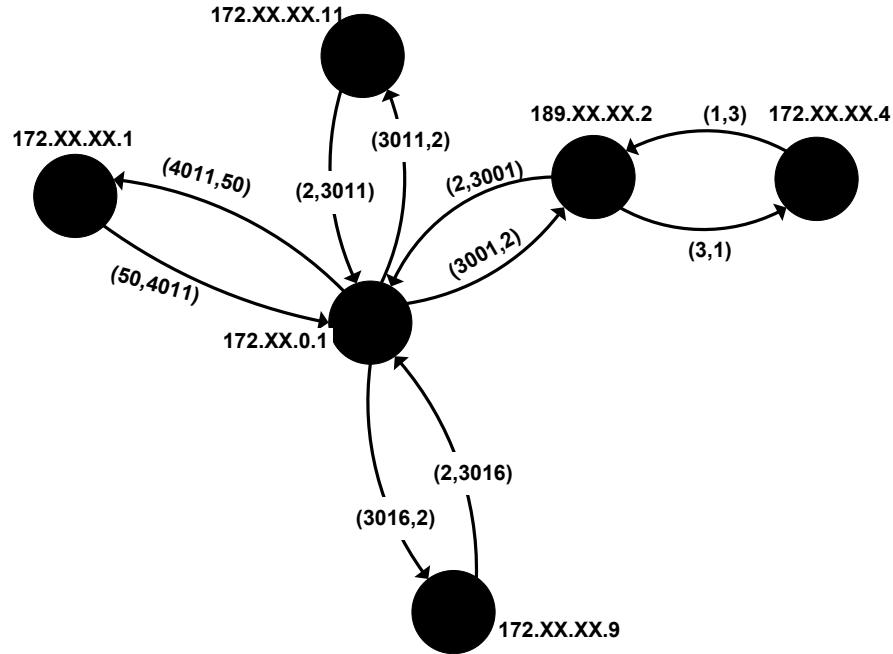


Figura 4.18 - Grafo que modela a topologia da rede monitorada.

Com base neste grafo, é realizada uma análise dos alertas de segundo nível gerados na mesma janela de cinco minutos. Os alarmes de segundo nível são analisados aos pares. Suponhamos um conjunto de alarmes de segundo nível  $S = \{s_1, s_2, \dots, s_n\}$ . Para este conjunto, são formados pares de alarmes  $J_{i,j}$ , combinando todos os alarmes:

$$J_{1,2} = \{s_1 \ s_2\}, \quad J_{1,n} = \{s_1 \ s_n\}, \quad J_{2,1} = \{s_2 \ s_1\}, \quad J_{2,n} = \{s_2 \ s_n\}, \quad J_{n,1} = \{s_n \ s_1\} \quad \text{e} \\ J_{n,2} = \{s_n \ s_2\}.$$

Cada par  $J_{i,j}$  é analisado para verificar se a anomalia se propagou do elemento de rede  $el_i$ , relacionado ao alarme  $s_i$ , para o elemento de rede  $el_j$ , relacionado ao alarme  $s_j$ . Certificamos que a anomalia se propagou entre os dois elementos de rede quando as seguintes condições são atendidas:

1. Há no grafo uma aresta direcionada que parte do vértice  $el_i$  e termina no vértice  $el_j$ ;
2. As classificações dos alarmes  $s_i$  e  $s_j$  atendem uma das afirmações a seguir:

- a. O alarme  $s_i$  aponta uma anomalia no fluxo de saída e o alarme  $s_j$  aponta uma anomalia no fluxo de entrada;
- b. O alarme  $s_i$  aponta uma anomalia no fluxo de saída e o alarme  $s_j$  aponta uma anomalia no fluxo de encaminhamento;
- c. O alarme  $s_i$  aponta uma anomalia no fluxo de encaminhamento e o alarme  $s_j$  aponta uma anomalia no fluxo de entrada;
- d. O alarme  $s_i$  aponta uma anomalia no fluxo de encaminhamento e o alarme  $s_j$  aponta uma anomalia no fluxo de encaminhamento;

Conforme os pares de alarmes são analisados, é construído um grafo que representará o mapa de propagação da anomalia pela rede. Quando é certificada a propagação da anomalia de  $el_i$  para  $el_j$ , são criados os vértices para estes dois equipamentos e uma aresta direcionada que representa a propagação da anomalia. Caso já existam vértices criados para os elementos de rede em questão na análise em curso, eles são utilizados, ao invés de serem criados novos vértices. Ao final da análise, o grafo com os caminhos de propagação da anomalia é incluído em um alarme de terceiro nível, que é enviado ao administrador de rede com a visão panorâmica do problema.

## **5 Implementação e resultados**

Este capítulo apresenta detalhes da implementação do sistema de detecção de anomalias na rede da UEL e os resultados obtidos a partir de testes realizados sobre situações reais de anomalia. Primeiramente, é apresentado o ambiente de rede real monitorado na UEL. Logo após, são apresentados detalhes sobre a implementação do modelo proposto no capítulo 4. São mostrados detalhes sobre a ferramenta GBA, a qual dispõe de funcionalidades básicas para a gerência de redes como coleta de dados em MIBs e geração de gráficos e foi utilizada como base para a construção da ferramenta de detecção de anomalias. A integração entre os módulos legados da ferramenta GBA e os novos módulos desenvolvidos neste trabalho também faz parte desta primeira parte do capítulo. Outra questão importante que é tratada é a formatação do modelo de detecção de anomalias proposto, necessária para sua utilização em ambiente real.

O restante do capítulo traz resultados obtidos a partir de testes realizados com dados coletados na rede da Universidade Estadual de Londrina. Foram utilizadas as métricas de taxa de detecção e de falsos positivos para avaliar o Módulo de Detecção de Anomalias e o Módulo de Configuração Automática. São apresentados ainda casos de anomalias que ocorreram na rede da UEL, detalhando o comportamento do sistema de detecção de anomalias e os benefícios trazidos com a sua aplicação.

### **5.1 Ambiente de rede monitorado**

A proposta de detecção de anomalias apresentada nesta tese foi desenvolvida utilizando observações realizadas em um ambiente de rede real: a rede da Universidade Estadual de Londrina (UEL). A Figura 5.1 mostra os elementos da rede da UEL que foram monitorados e como eles interagem. Segue a lista dos elementos com seus respectivos detalhes:

- Switch Black Diamond (BD), endereço IP 172.XX.0.1: é o switch que concentra as principais ligações do núcleo da rede da UEL. Ele está conectado

com os principais servidores, os *switches* que dão acesso às sub-redes das unidades da universidade e o firewall que se conecta com a Internet;

- Switch Transit, endereço IP 172.XX.XX.4: este switch opera como um intermediário entre os *links* de Internet contratados pela UEL e o firewall da rede da universidade;
- Firewall, endereço IP 189.XX.XX.2: é um firewall do tipo filtragem de pacotes que opera na camada de rede. Sua função é delimitar o perímetro de defesa da rede da universidade perante a Internet;
- Servidor *proxy*, endereço IP 172.XX.XX.11: é o servidor responsável por controlar o acesso dos 5000 computadores da universidade à Internet. Ele opera como *cache* das páginas mais acessadas e impede o acesso a conteúdos não autorizados;
- Servidor *web*, endereço IP 172.XX.XX.9: é o principal servidor *web* da UEL. Por meio deste servidor, são disponibilizados conteúdos importantes, como a divulgação de resultado do vestibular;
- Switch DC, endereço IP 172.XX.XX.1: é o switch que interconecta a sub-rede do Departamento de Computação (DC) da UEL ao núcleo da rede da universidade;

Uma propriedade importante deste cenário é que ele reúne equipamentos com características de operação diferentes, que podem se concentrar na camada de enlace, de rede ou de transporte do protocolo TCP/IP. Primeiramente, temos os switches, que concentram as suas operações na camada de enlace. Da mesma forma, temos os servidores, que ao estabelecer conexões com seus clientes, concentram suas operações na camada de transporte e o firewall, que concentra suas operações na camada de rede. Todos estes equipamentos necessitam de ações direcionadas a estas características durante o monitoramento e detecção de anomalias. Apresentando esta diversidade, o cenário escolhido traz os requisitos necessários para que possamos desenvolver um sistema de detecção e localização de anomalias que atinja os nossos objetivos.

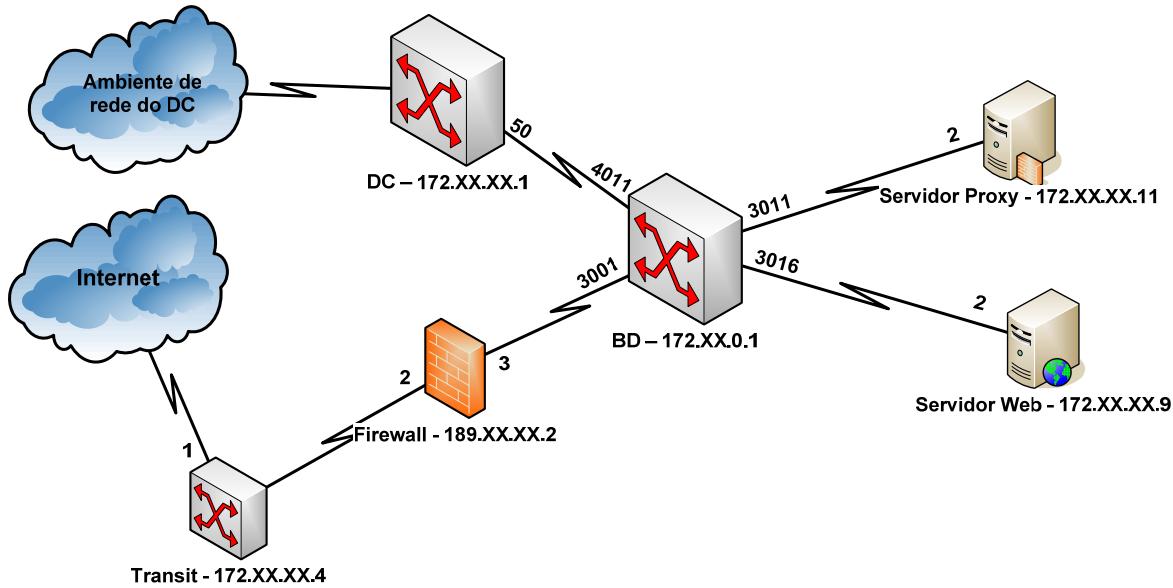


Figura 5.1 - Elementos de rede monitorados para o desenvolvimento e testes do sistema de detecção de anomalias.

## 5.2 Desenvolvimento com a ferramenta GBA

A proposta para detecção de anomalias apresentada neste trabalho foi implementada dentro do contexto da ferramenta GBA (Gerenciamento de *Backbone* Automático), que atualmente se encontra na versão 6.0 e foi inicialmente desenvolvida por Proença Junior (2005). A Figura 5.2 mostra os componentes que formam a ferramenta GBA, que são detalhados a seguir:

- GBA Le switch: é um programa executável para plataforma Windows escrito em C++. Ele é responsável por coletar as informações em tempo real de MIBs e gravá-las em arquivos com formato binário de extensão “.les”. Para acessar as MIBs, o GBA Le switch utiliza a biblioteca NET-SNMP (NET-SNMP, 2010);
- GBA Gera baseline: assim como o GBA Le switch, é um programa executável para plataforma Windows escrito em C++. Este módulo gera DSNS utilizando o histórico de leituras da rede armazenado pelo GBA Le switch. Os DSNS são armazenados em arquivos com formato binário de extensão “.bln”.
- GBA EJB Leituras: é um componente escrito em Java, que utiliza a tecnologia EJB (*Enterprise Java Beans*) versão 3 (EJB, 2010) (PANDA et al., 2007). Ele é

o responsável por acessar os arquivos de leituras gerados pelo GBA Le switch e transformá-los em objetos Java que são transmitidos para outros módulos;

- GBA EJB Baselines: é um componente escrito em Java, que utiliza a tecnologia EJB, assim como o GBA EJB Leituras. Ele é responsável por acessar os arquivos de DSNS gerados pelo GBA Gera Baseline e transformá-los em objetos Java que são transmitidos para outros módulos;
- GBA Gera Grafico: é um módulo que também utiliza a tecnologia EJB e a linguagem Java. Ele gera gráficos que podem conter as leituras e os valores de DSNS para diferentes períodos de tempo.

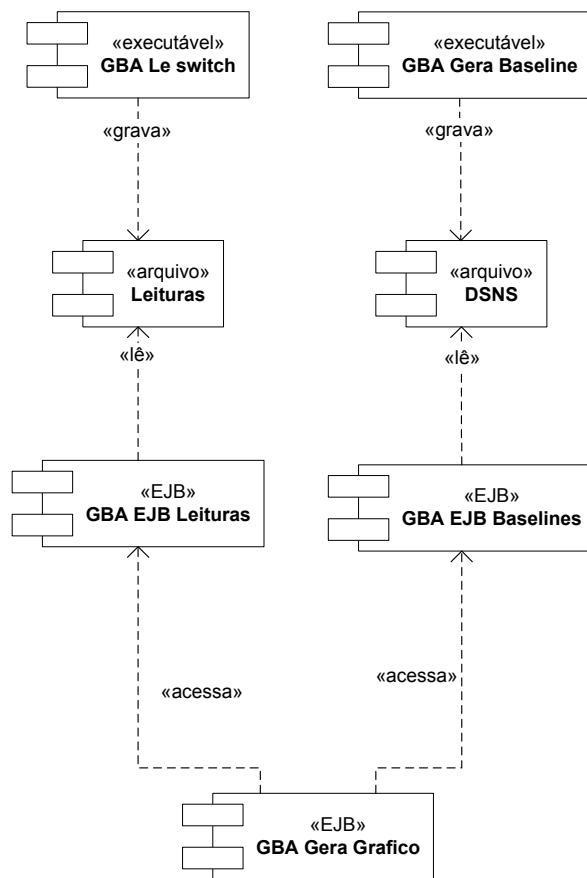


Figura 5.2 - Diagrama de componentes da ferramenta GBA versão 6.0.

No GBA Le switch, a interface com o usuário é feita por meio de arquivos de configuração. O GBA Gera Baseline e o GBA Gera Grafico podem ser acessados por meio de interface disponível na WEB. As páginas WEB e os componentes EJB dos módulos

GBA EJB Leituras, GBA EJB Baselines e GBA Gera Graficos são implantados em um servidor de aplicação GlassFish, hospedado na Universidade Estadual de Londrina (UEL).

O EJB versão 3 é uma tecnologia presente na plataforma Java, desenvolvida pela Sun Microsystems. Ele funciona como um *framework* para o desenvolvimento de aplicações distribuídas, que ficam hospedadas em servidores de aplicação com suporte a tecnologia EJB como é o caso do GlassFish (EJB, 2010) (PANDA et al., 2007). Ao utilizar EJB no desenvolvimento da ferramenta GBA, construímos uma solução distribuída, na qual cada módulo opera como um serviço.

Os módulos que compõem o sistema de detecção de anomalias também foram construídos utilizando a linguagem Java e a tecnologia EJB. A Figura 5.3 apresenta os módulos que compõem o sistema e como eles interagem com os módulos já existentes do GBA. A seguir, detalhamos cada um dos módulos:

- GBA Analise Objeto SNMP: componente que compara as leituras e os DSNS, gerando alarmes de primeiro nível quando um desvio de comportamento é detectado. O GBA Analise Objeto SNMP acessa o componente GBA EJB Leituras para obter os dados coletados na rede e o componente GBA EJB Baselines para obter os dados dos DSNS gerados pelo GBA Gera Baseline;
- GBA Correlação: componente que reúne os alarmes de primeiro nível gerados para diferentes objetos SNMP e os analisa para gerar o alarme de segundo nível. O componente GBA Correlação acessa o componente GBA Analise Objeto SNMP para obter os alarmes de primeiro nível;
- GBA Localização: componente que reúne os alarmes de segundo nível a fim de gerar um alarme de terceiro nível que contém o mapa de propagação da anomalia pela rede. Ele acessa o componente GBA Correlação para obter os alarmes de segundo nível.
- GBA Cliente: este componente é um programa executável escrito em Java, capaz de acessar componentes EJB. Através deste programa cliente, o usuário envia as configurações desejadas para os componentes GBA Correlação e GBA Localização. O GBA Cliente também produz relatórios

com os alarmes gerados e executa a rotina de configuração automática dos parâmetros do sistema.

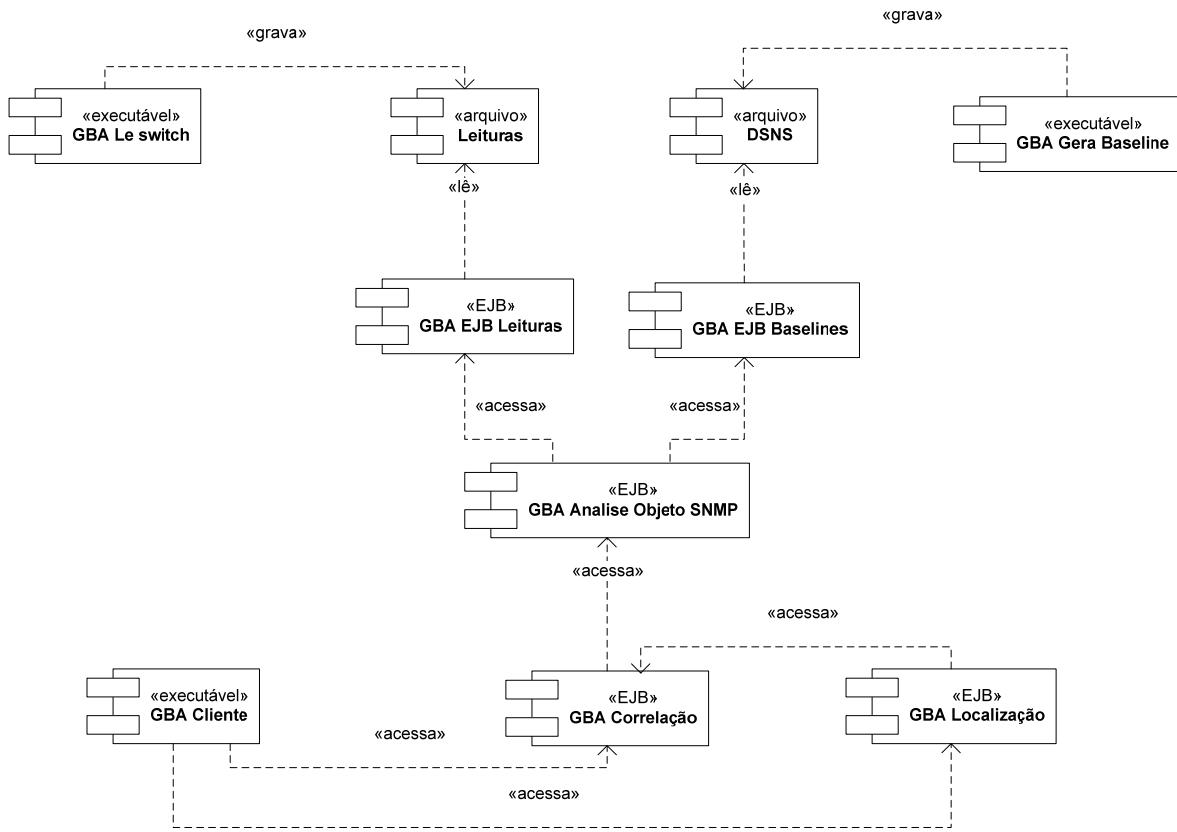


Figura 5.3 - Diagrama de componentes da ferramenta GBA incluindo a solução de detecção de anomalias.

### 5.3 **Ajustes no modelo para implantação na rede da UEL**

Esta seção apresenta a formatação dos grafos de dependências do Módulo de Detecção de Anomalias e do Módulo de Localização de Anomalias que foi realizada para adaptá-los ao ambiente de rede da UEL. No caso do grafo de dependências do Módulo de Detecção de Anomalias, ajustes foram realizados para evitar a utilização de objetos com erros de coleta e a sobrecarga dos equipamentos monitorados. No caso do grafo de dependências do Módulo de Localização de Anomalias, foi realizada a formatação para a topologia de rede da UEL.

O grafo de dependências do Módulo de Detecção de Anomalias, apresentado na Figura 4.10, contém 35 objetos SNMP. Ele inclui todos os objetos listados nos diagramas

de Case para os grupos *interface*, *ip*, *tcp* e *udp*. Porém, ele não deve ser aplicado em sua totalidade para o monitoramento de todos os elementos de rede, por duas razões principais:

- Não é recomendável monitorar 35 objetos simultaneamente, por questões de desempenho. O excesso de tráfego gerado pelos pacotes ligados ao protocolo SNMP pode causar congestionamentos, prejudicando as operações da rede;
- Os objetos que são monitorados em cada elemento de rede devem estar relacionados com a função desempenhada pelo elemento. Na rede da UEL, por exemplo, temos switches, servidores e um firewall. Os switches realizam a comutação de pacotes na camada de enlace. Portanto, não é necessário monitorar objetos relacionados às camadas de rede e transporte nestes equipamentos. No caso do firewall, que realiza a filtragem e encaminhamento de pacotes na camada de rede, não há necessidade de monitorar objetos ligados ao protocolo TCP. No caso dos servidores, não é necessário monitorar objetos que estão relacionados ao encaminhamento de pacotes como o *ipForwDatagrams*.

Portanto, o grafo de dependências proposto com 35 objetos SNMP deve ser utilizado como uma referência, da qual é escolhida uma instância com uma quantidade menor de objetos. Neste trabalho, focamos os nove objetos SNMP que se posicionam nas transições entre as camadas do protocolo TCP/IP, sendo representados nos diagramas de Case como setas de filtro. Desta forma, monitoramos objetos como o *ipInReceives*, que contabiliza os pacotes que ingressaram na camada de rede vindo da camada de enlace e o *ipInDelivers*, que contabiliza os pacotes que foram entregues da camada de rede para a camada de transporte. O objetivo é observar a propagação da anomalia ao longo das camadas no elemento analisado. A única exceção é o objeto *ipForwDatagrams*, que mesmo não atendendo a este requisito, foi escolhido por ser a única forma de observar o encaminhamento de pacotes. A Figura 5.4 mostra o grafo de dependências que contém os objetos escolhidos.

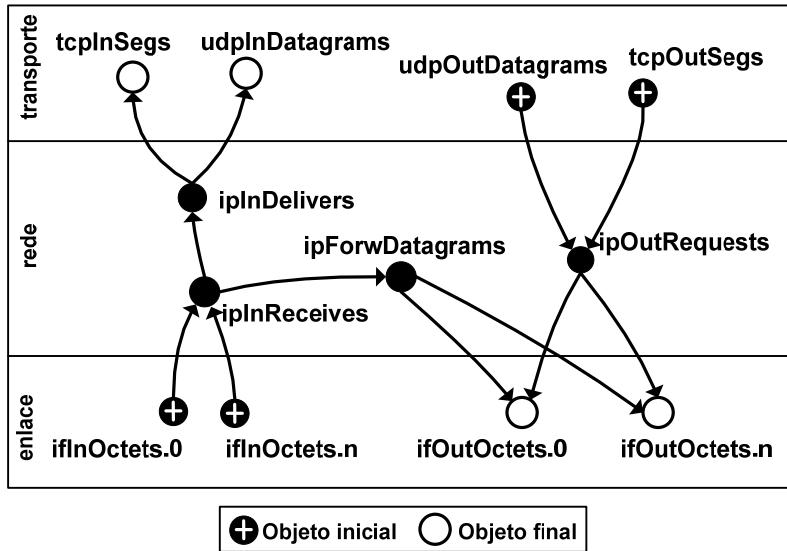


Figura 5.4 - Grafo de dependências utilizado nos testes.

A Figura 5.5 mostra os objetos monitorados em cada elemento de rede e como eles interagem dentro do contexto do ambiente de rede da UEL. Os objetos marcados em vermelho apresentaram problemas durante o monitoramento. Nos servidores da UEL, os objetos do grupo *interface* apresentam muitos erros de leitura quando há coletas em intervalos menores que 30 segundos. Foi necessário eliminar estes objetos do monitoramento, construindo o cenário apresentado na Figura 5.6.

A Figura 5.7 traz o grafo de dependências para a topologia da rede, que será usado no Módulo de Localização de Anomalias. É possível observar portas iguais a “0” em alguns atributos de arestas. Isto ocorre quando não há objetos do grupo *interface* sendo monitorados no equipamento em questão.

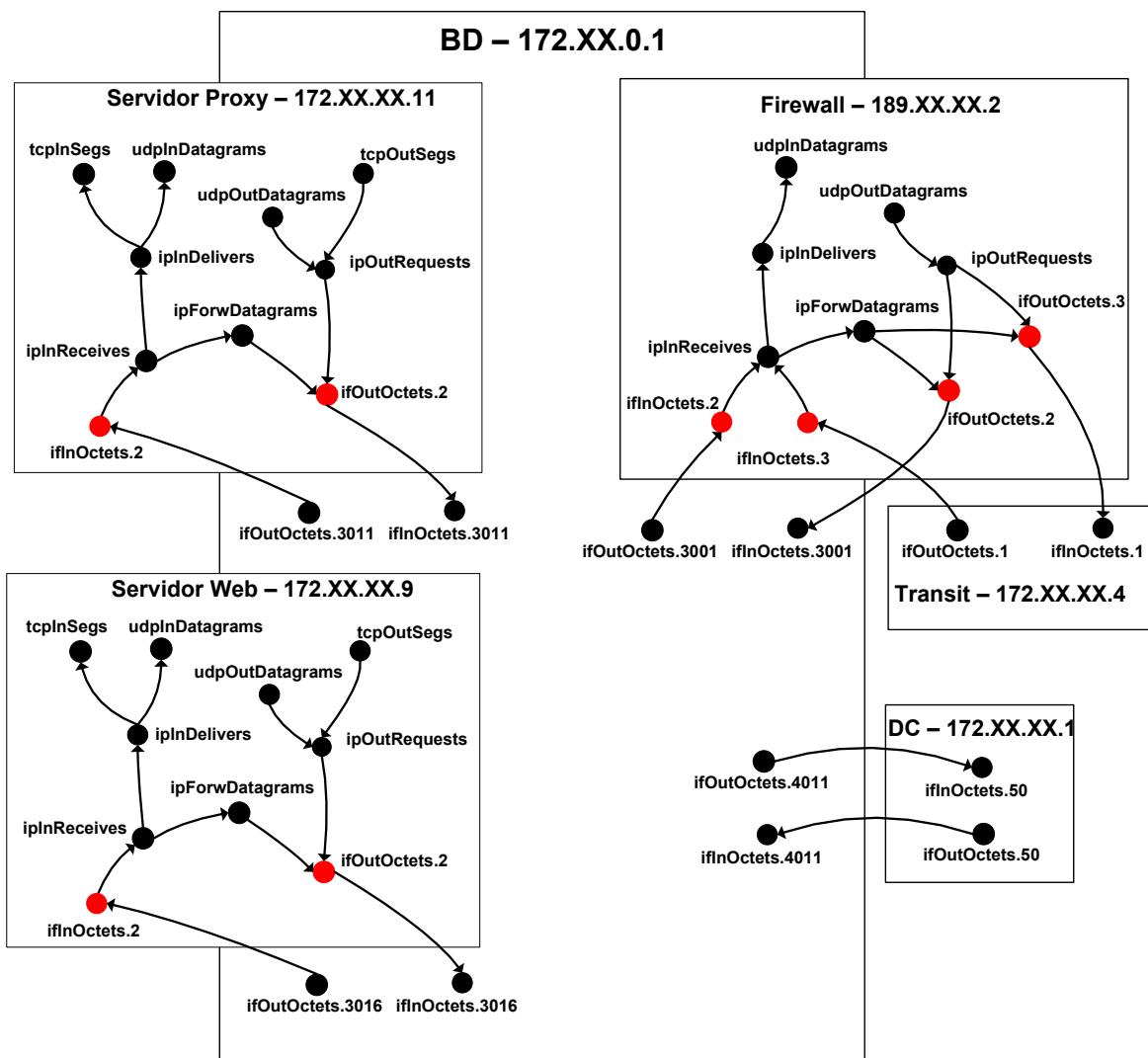


Figura 5.5 - Objetos monitorados dentro do contexto do ambiente de rede da UEL.

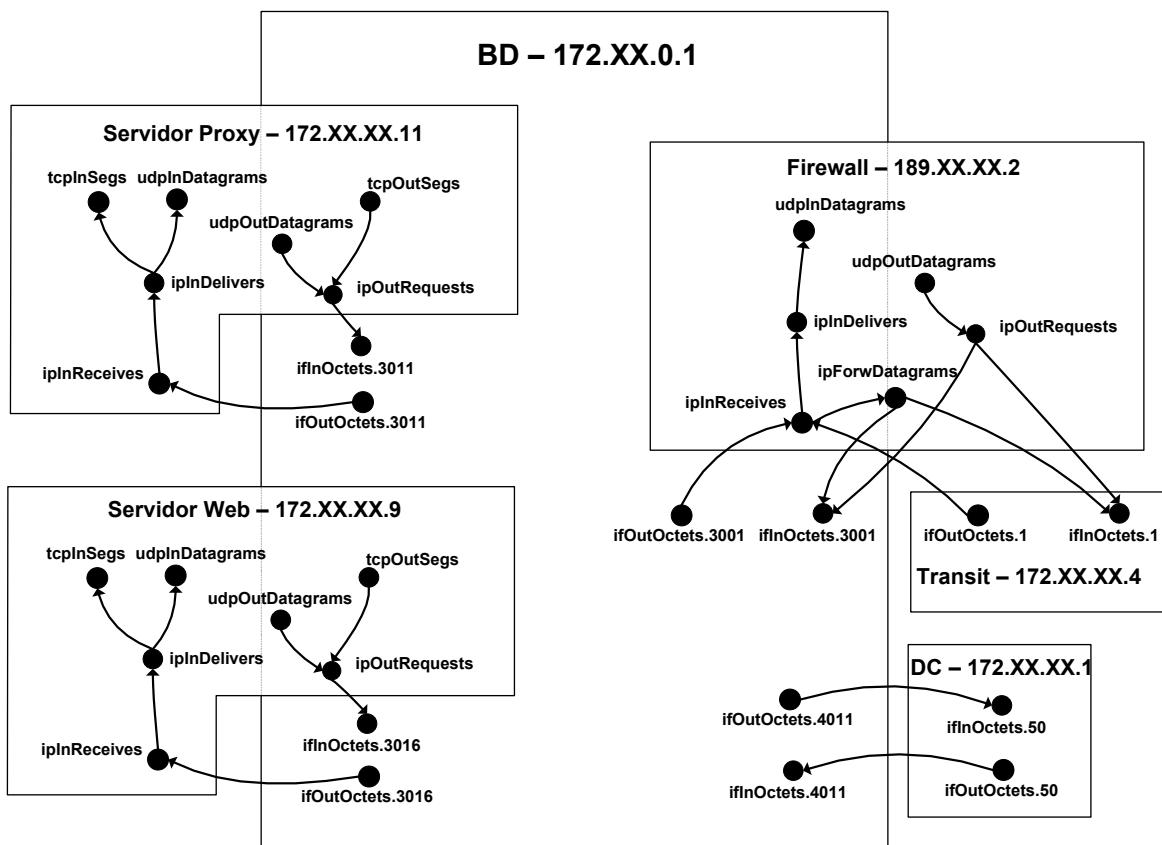


Figura 5.6 - Objetos monitorados dentro do contexto do ambiente de rede da UEL – remoção de objetos com problemas.

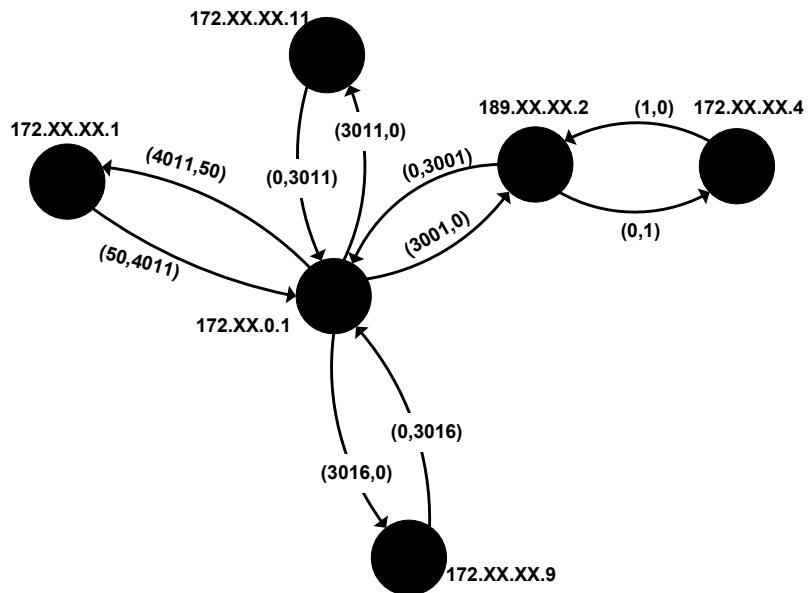


Figura 5.7 - Grafo da topologia de rede aplicado nos testes.

## 5.4 Resultados do Módulo de Detecção de Anomalias

Estes resultados têm por objetivo mostrar o potencial de detecção do Módulo de Detecção de Anomalias. Para tanto, foram usados gráficos conhecidos como Curvas ROC (*Receiver Operating Characteristic*) (SOULE et al., 2005). Para construir este gráfico, o sistema deve ser testado com diferentes níveis de sensibilidade. Neste trabalho, isto é alcançado com a variação do valor do  $\delta$  para um determinado valor de intervalo de histerese. As taxas de detecção e de falsos positivos verificadas para cada um dos níveis de sensibilidade testados são inseridas no gráfico. O eixo y traz as taxas de detecção e o eixo x traz as taxas de falsos positivos. Quanto mais próxima do ponto (0,1) é a curva, mais eficiente é o sistema de detecção. Este gráfico também é útil para verificarmos qual configuração de sensibilidade apresenta o melhor desempenho.

Para realização destes testes, foram definidos dois grupos de anomalias, denominados grupo A e grupo B, utilizando o algoritmo parametrizado de definição de anomalias apresentado na Tabela 4.1. Para o grupo A, o algoritmo foi executado com os seguintes valores de parâmetros:  $\alpha = 0,6$  e  $\gamma = 1$ . O grupo A simula a política de gerência que procura detectar a grande maioria dos desvios de comportamento. O grupo B, por outro lado, representa uma política que pretende detectar desvios mais persistentes. No caso do grupo B, foram utilizados os seguintes valores para os parâmetros:  $\alpha = 0,6$  e  $\gamma = 3$ .

Os testes foram realizados no Servidor Proxy, no Servidor Web e no Firewall, entre os dias 29/03/2009 e 02/05/2009. Para cada um destes elementos de rede, foram executados 5 conjuntos de testes. O primeiro conjunto com intervalo de histerese igual a 60 segundos, o segundo com intervalo de histerese igual a 120 segundos e assim por diante, acrescentando sempre 60 segundos até chegar ao quinto conjunto com 300 segundos. Em cada um dos conjuntos foram aplicados testes com diferentes valores de  $\delta$ , variando de 1 ao primeiro valor que faça a taxa de detecção e a taxa de falsos positivos serem iguais a zero.

A seguir são apresentadas as curvas ROC construídas durante os testes. Tabelas com os valores dos resultados para cada configuração de sensibilidade testada podem ser encontradas no Anexo A.

A Figura 5.8 traz os resultados para o grupo A de anomalias no servidor Proxy. É possível observar que os resultados para os diferentes intervalos de histerese seguem

comportamentos semelhantes. Isto mostra que, mesmo com intervalos de histerese diferentes, é possível encontrar valores para  $\delta$  que tragam resultados semelhantes. Por exemplo, com histerese igual a 60 segundos e  $\delta$  igual a 2, é obtida uma taxa de detecção de 80% e uma taxa de falsos positivos de 10%. Com histerese igual a 120 e  $\delta$  igual a 3, são obtidos resultados semelhantes, com a taxa de detecção em 76% e a taxa de falsos positivos em 11%. Mesmo com intervalo de histerese de 300 segundos, que é bem maior que o inicial de 60 segundos, são obtidas para o  $\delta$  igual a 4 a taxa de detecção igual a 75% e a de falsos positivos igual a 13%. Os resultados para o Proxy são satisfatórios, já que em determinadas configurações foram alcançadas simultaneamente uma taxa de detecção próxima a 80% e a de falsos positivos próxima a 10%.

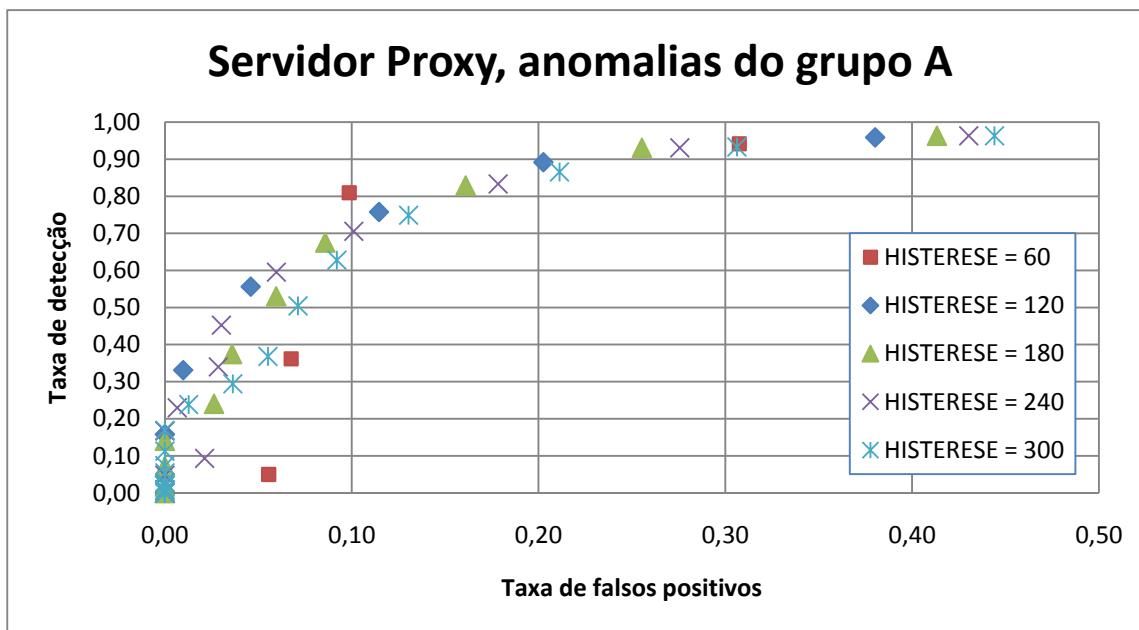


Figura 5.8 - Curva ROC para o servidor Proxy, anomalias do grupo A.

A Figura 5.9 mostra os resultados para o grupo A de anomalias no servidor Web. É possível observar que os pontos do gráfico se posicionam em uma linha um pouco abaixo do que foi observado no servidor Proxy. Desta forma, conclui-se que a performance do Módulo de Detecção de Anomalias ao monitorar o servidor Web foi levemente inferior à performance no monitoramento do servidor Proxy. De maneira similar ao servidor Proxy, foi observado, no servidor Web, que os resultados para os diferentes intervalos de histerese se aproximam. O Módulo de Detecção de Anomalias também obteve bons resultados para o

servidor Web. Algumas configurações resultaram em taxas de detecção e de falsos positivos por volta de 90% e 15%, respectivamente.

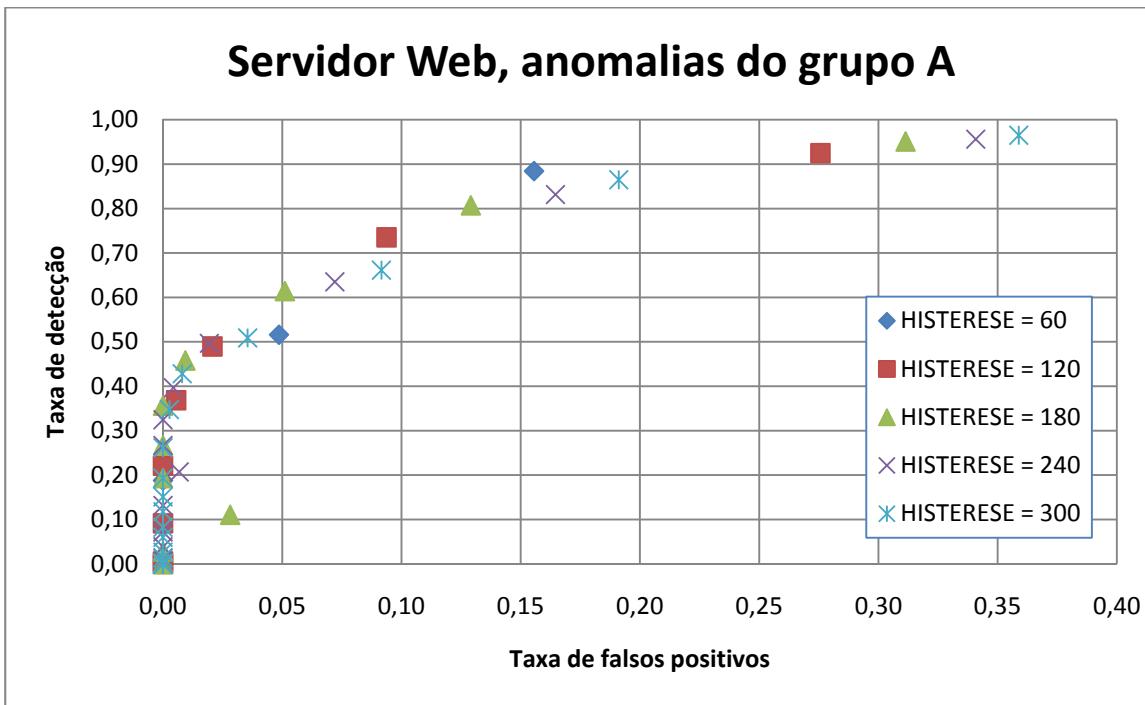


Figura 5.9 - Curva ROC para o servidor Web, anomalias do grupo A.

A Figura 5.10 traz as taxas de falsos positivos e de detecção obtidas no monitoramento do Firewall, levando em conta o grupo A de anomalias. A performance do Módulo de Detecção de Anomalias no monitoramento do Firewall foi inferior aos outros dois casos apresentados anteriormente. É possível perceber que os pontos do gráfico da Figura 5.10 formam uma curva mais próxima do centro do gráfico que na Figura 5.8 e na Figura 5.9, que mostram os resultados para o Servidor Proxy e o Servidor Web, respectivamente. Após observar o comportamento do tráfego no Firewall, foi constatado que este elemento apresenta menos anomalias que os outros, e elas são normalmente bastante discretas, dificultando o trabalho do Módulo de Detecção de Anomalias. Este fenômeno ocorre devido ao fato do Firewall agregar todo o tráfego que atravessa o perímetro entre a rede da universidade e a Internet. Desta forma, alguns desvios de comportamento que aparecem com grande intensidade em equipamentos internos à rede, os quais possuem níveis de tráfego bem mais baixos que o Firewall, podem não apresentar a mesma magnitude quando analisados no contexto do Firewall, o qual apresenta altos níveis de tráfego.

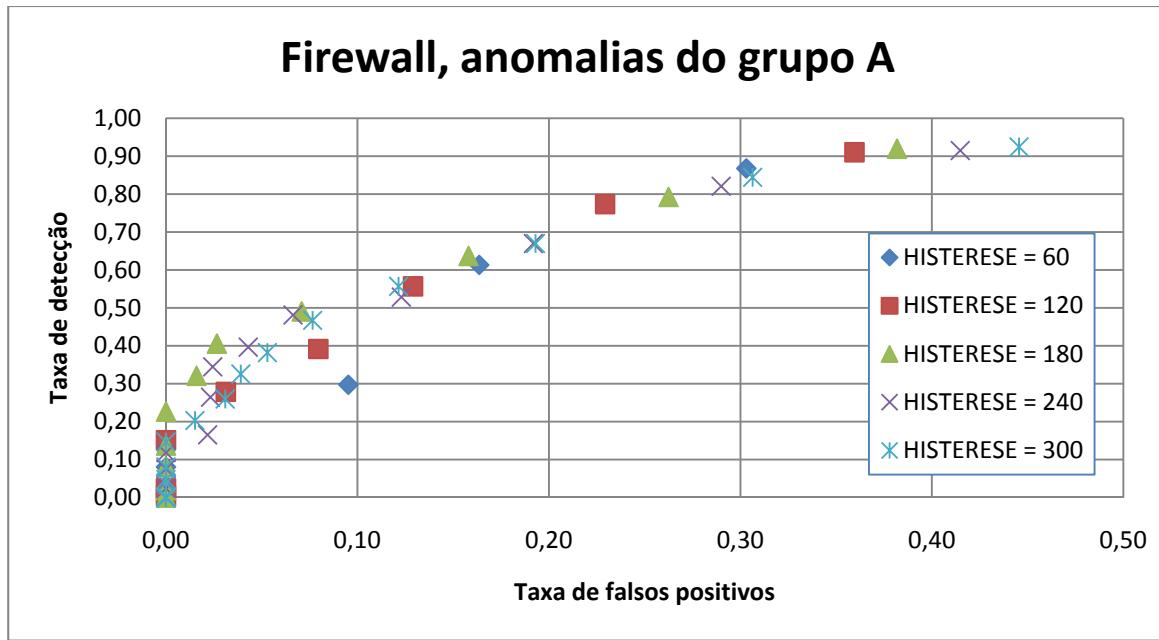


Figura 5.10 - Curva ROC para o Firewall, anomalias do grupo A.

A Figura 5.11 mostra os resultados para o Servidor Proxy, agora levando em conta as anomalias do grupo B. Os resultados são bastante satisfatórios, já que foram alcançadas boas taxas, como é o caso da configuração com intervalo de histerese igual a 240 segundos e  $\delta$  igual a 6, onde foi atingida uma taxa de detecção igual a 92% e uma taxa de falsos positivos igual a 15%. Como no grupo B de anomalias os desvios mais curtos não são considerados anomalias, não foi encontrado um valor de  $\delta$  para o intervalo de histerese igual a 60 segundos que ofereça bons resultados para as taxas de detecção e de falsos positivos simultaneamente. As melhores taxas de detecção são acompanhadas por taxas de falsos positivos bastante elevadas.

A Figura 5.12 traz os resultados para o Servidor Web, utilizando como referência o grupo B de anomalias. É possível observar que os resultados são levemente superiores aos obtidos para o servidor Proxy. Para as configurações com intervalo de histerese igual a 240 segundos e  $\delta$  igual a 6 e 7, o sistema alcançou taxas de detecção iguais a 92% e 87% e taxas de falsos positivos de 12% e 6%, respectivamente. Estes resultados são bastante satisfatórios.

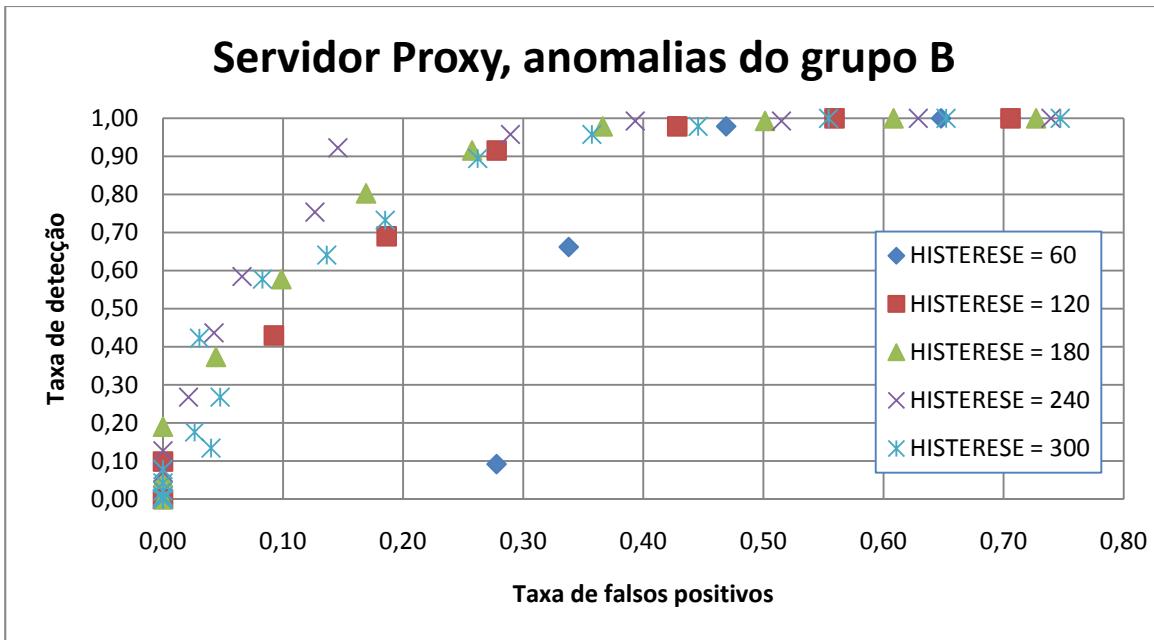


Figura 5.11 - Curva ROC para o servidor Proxy, anomalias do grupo B.

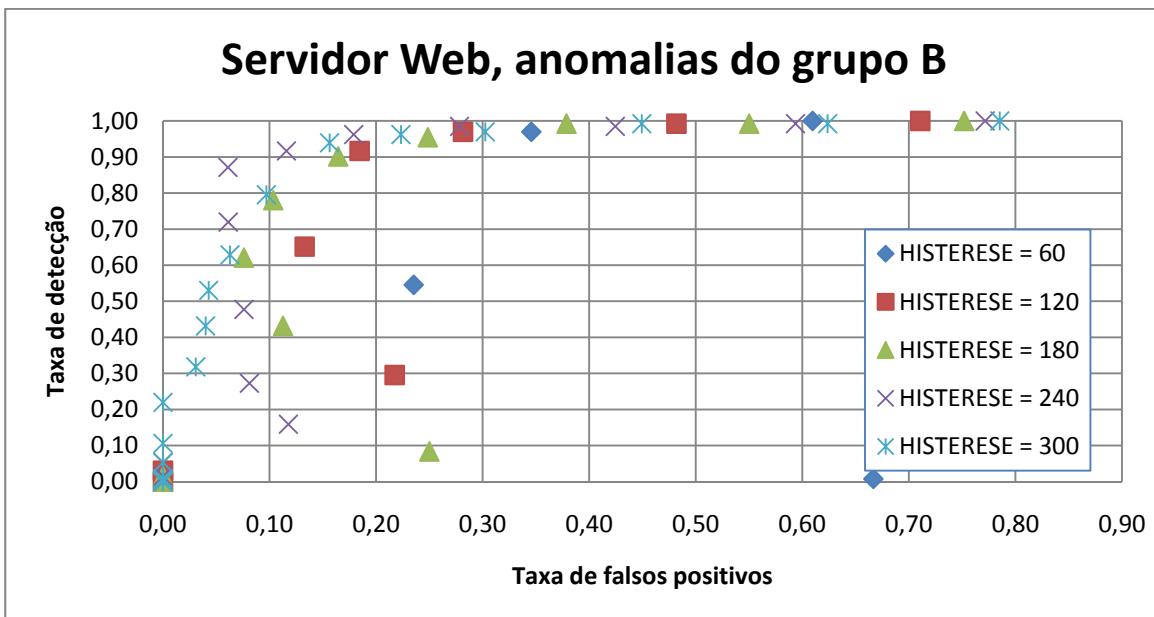


Figura 5.12 - Curva ROC para o servidor Web, anomalias do grupo B.

Os resultados para a detecção das anomalias do grupo B no Firewall são apresentados na Figura 5.13. Como já havia ocorrido no grupo A de anomalias, os resultados para o Firewall são inferiores aos resultados obtidos para o Servidor Proxy e o Servidor Web. A razão é a mesma: o Firewall apresenta menos desvios de comportamento, e em geral eles são mais discretos, apresentando valores de tráfego acima, porém próximos

dos respectivos valores do DSNS. Estas características de desvios dificultam as operações do Módulo de Detecção de Anomalias, fazendo, neste caso, com que as taxas de falsos positivos sejam altas.

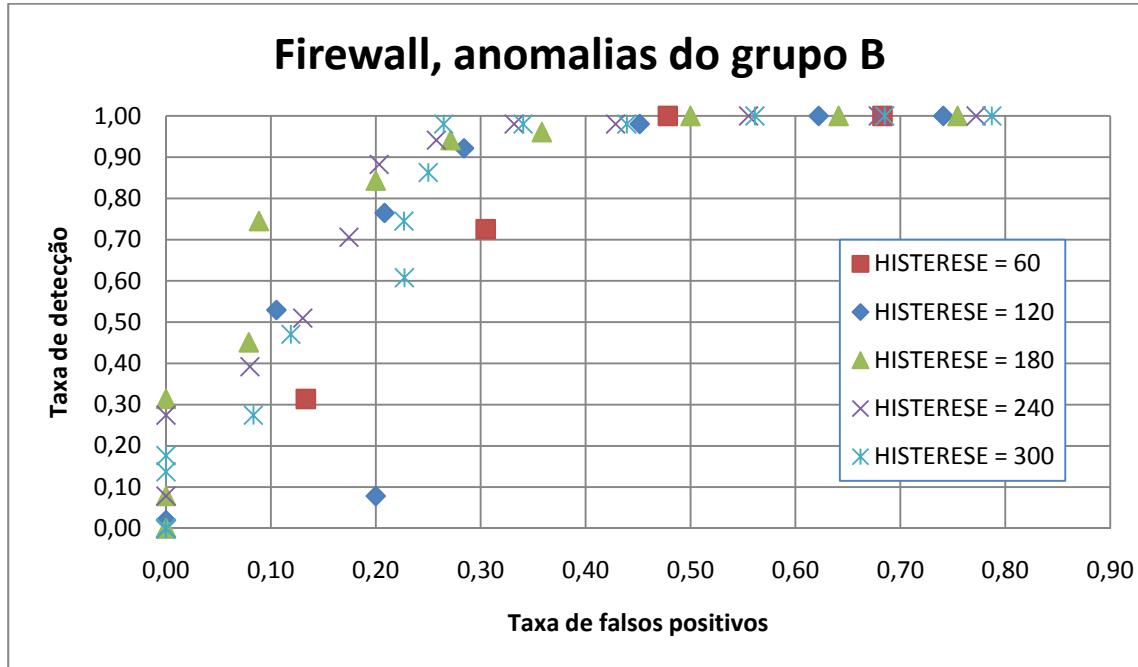


Figura 5.13 - Curva ROC para o Firewall, anomalias do grupo B.

A Figura 5.14 traz um comparativo entre os resultados para o grupo A e para o grupo B no servidor Proxy. É possível observar que os resultados são bastante semelhantes e satisfatórios, conforme já havia sido observado nas análises individuais de desempenho para cada um dos grupos de anomalias. Estes resultados mostram que, no caso do Servidor Proxy, as diferentes sensibilidades encontradas com a configuração dos parâmetros do Módulo de Detecção de Anomalias levaram a resultados semelhantes e bons em grupos de anomalias com características diferentes.

A Figura 5.15 traz o comparativo para o Servidor Web. Podemos observar que os resultados para o grupo de anomalias B são levemente superiores aos resultados para o grupo de anomalias A. O grupo A possui anomalias mais sutis, mais fáceis de serem confundidas com uma variação natural em relação ao grupo B, e por isso suas taxas de falsos positivos são um pouco mais elevadas para taxas de detecção semelhantes.

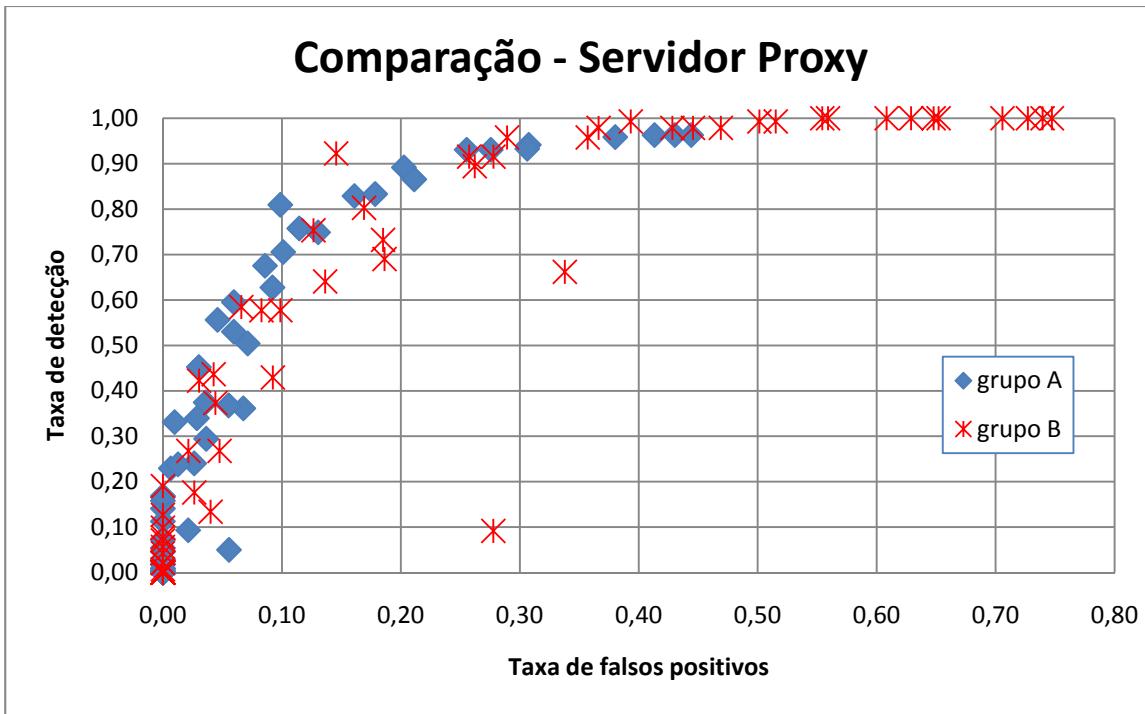


Figura 5.14 - Comparação entre desempenho do Módulo de Detecção de Anomalias para os grupos A e B de anomalias no Servidor Proxy.

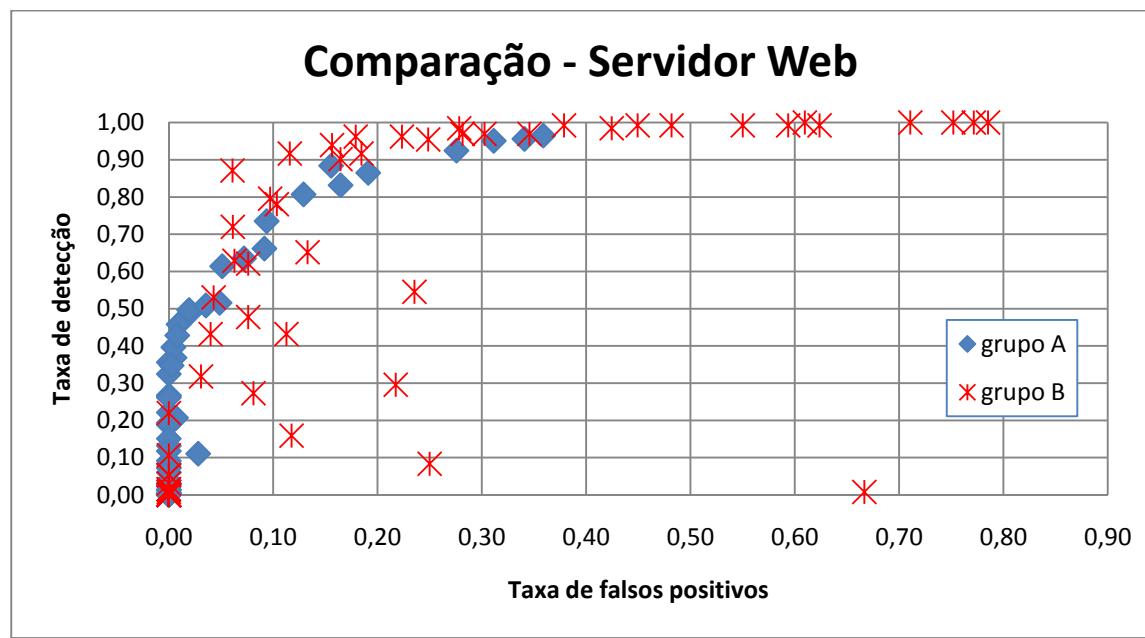


Figura 5.15 - Comparação entre desempenho do Módulo de Detecção de Anomalias para os grupos A e B de anomalias no Servidor Web.

A Figura 5.16 traz a comparação realizada para o Firewall. Os resultados para o grupo B de anomalias são levemente superiores aos encontrados para o grupo A. No Firewall, este fato também ocorre porque as anomalias do grupo A são mais sutis e difíceis de serem detectadas, levando a um desempenho inferior em relação ao encontrado para as anomalias do grupo B.

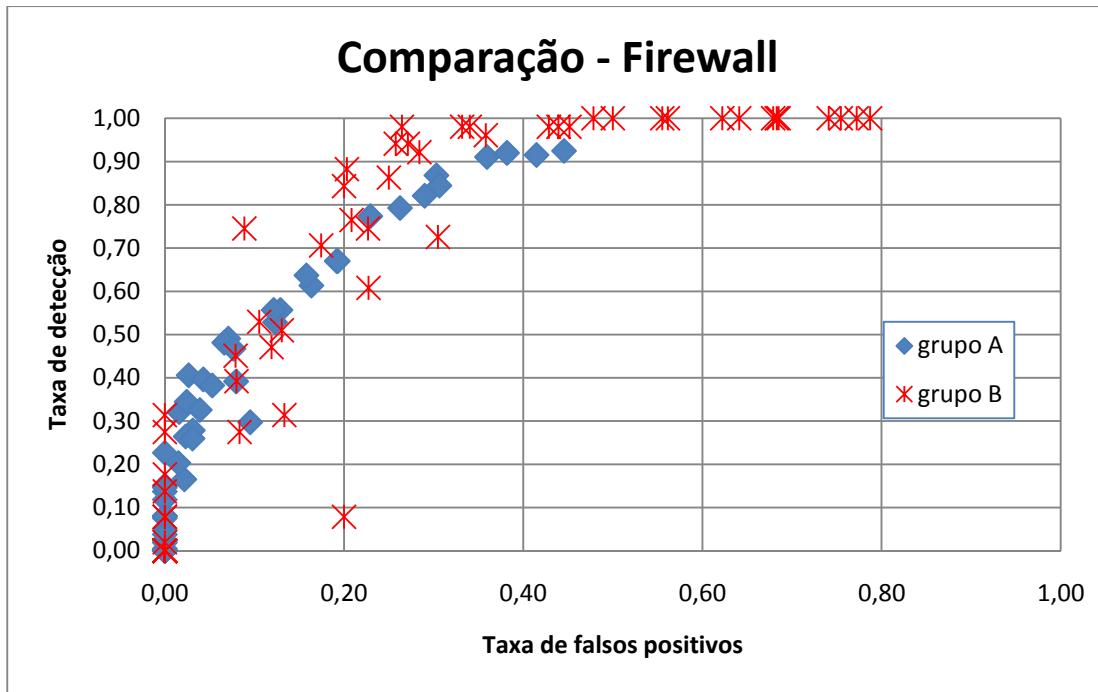


Figura 5.16 - Comparação entre desempenho do Módulo de Detecção de Anomalias para os grupos A e B de anomalias no Firewall.

Após apresentar estes resultados, há algumas conclusões que podem ser levantadas. O Módulo de Detecção de Anomalias apresentou bom desempenho, mesmo levando em conta anomalias com diferentes características e equipamentos diversos. Em vários casos, foi possível obter, simultaneamente, taxas de detecção próximas a 90% e taxas de falsos positivos próximas a 15%. O ponto negativo ficou nos resultados do Firewall, que foram inferiores aos obtidos para os outros dois equipamentos testados. O Módulo de Detecção de Anomalias encontrou dificuldades para lidar com os desvios de comportamento deste equipamento, que são mais discretos. Na comparação entre os resultados para os grupos A e B de anomalias, foi observada uma leve superioridade nos resultados do grupo B. Isto ocorre porque as anomalias do grupo B são mais fáceis de serem distinguidas do tráfego normal, facilitando o trabalho do Módulo de Detecção de Anomalias.

## 5.5 ***Resultados do Módulo de Configuração Automática***

Os experimentos conduzidos nesta seção têm como objetivo avaliar o desempenho do Módulo de Configuração Automática. Para tanto, as configurações definidas por este módulo ao longo de quatro semanas para o Servidor Proxy, o Servidor Web e o Firewall foram aplicadas e então analisadas segundo as taxas de detecção e de falsos positivos obtidas. Foram utilizados os grupos de anomalias A e B, assim como ocorreu nos testes que avaliaram o desempenho do Módulo de Detecção de Anomalias.

Os dados utilizados para estes testes foram coletados entre 29/03/2009 e 02/05/2009. A Tabela 5.1 mostra quais semanas foram usadas como período de treinamento, juntamente com os respectivos períodos de aplicação das configurações escolhidas. É importante lembrar que o Módulo de Configuração Automática testa diferentes combinações de parâmetros durante um período de treinamento e escolhe a combinação que resulta no melhor índice de eficiência, conforme mencionado na seção 4.4. Esta combinação de parâmetros é então aplicada em semanas posteriores.

Tabela 5.1 - Períodos de treinamento e respectivas semanas analisadas.

	<i>Semana de aplicação da configuração escolhida</i>	<i>Período de treinamento</i>
1	De 5 a 11 de abril	De 29 de março a 4 de abril
2	De 12 a 18 de abril	De 29 de março a 11 de abril
3	De 19 a 25 de abril	De 29 de março a 18 de abril
4	De 26 de abril a 2 de maio	De 29 de março a 25 de abril

A Figura 5.17 mostra as taxas de detecção obtidas para o grupo A de anomalias. Os resultados são bons, já que as taxas estão todas próximas ou acima de 80%. Isto demonstra que o Módulo de Configuração Automática escolheu boas configurações ao longo de quatro semanas para os equipamentos analisados.

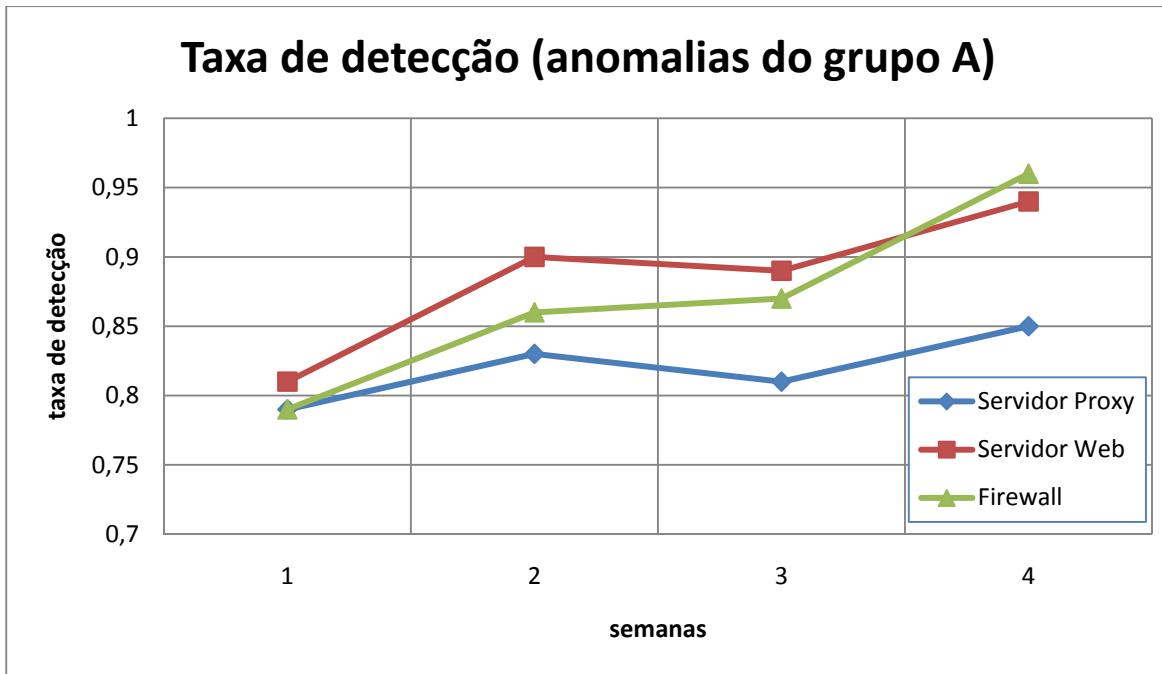


Figura 5.17 - Taxas de detecção para anomalias do grupo A.

As taxas de falsos positivos para o grupo A de anomalias são apresentadas na Figura 5.18. A maioria dos resultados ficou na faixa entre 12 e 20%. O Firewall apresentou resultados ruins, próximos a 40%. Esta situação já havia sido verificada nos testes do Módulo de Detecção de Anomalias, o qual utilizou o mesmo período de coleta e os mesmos equipamentos. Portanto, o fato de haver resultados ruins para o Firewall não é consequência de escolhas erradas do Módulo de Configuração Automática e sim de dificuldades enfrentadas pelo Módulo de Detecção de Anomalias. Para os outros equipamentos, podemos observar que o Módulo de Configuração Automática foi capaz de escolher boas configurações, as quais resultaram em boas taxas.

A Figura 5.19 traz a taxa de detecção para o grupo B de anomalias, onde novamente temos bons resultados para as configurações selecionadas pelo Módulo de Configuração Automática. Para o Firewall, 3 das 4 taxas de detecção coletadas foram iguais a 100%, enquanto para o servidor Proxy elas se situaram na faixa entre 90 e 100% e para o servidor Web na faixa entre 80 e 90%.

### Taxa de falsos positivos (anomalias do grupo A)

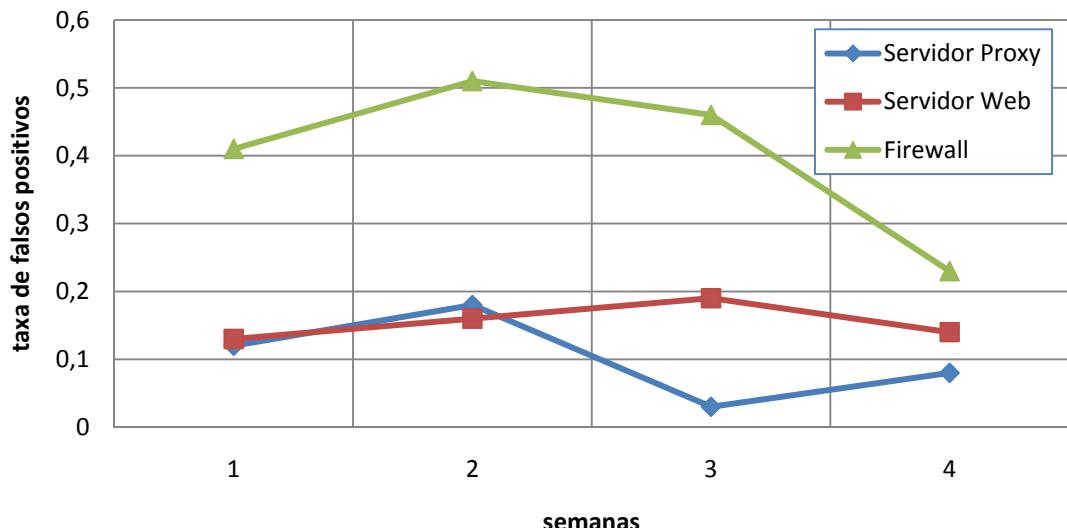


Figura 5.18 - Taxas de falsos positivos para anomalias do grupo A.

### Taxa de detecção (anomalias do grupo B)

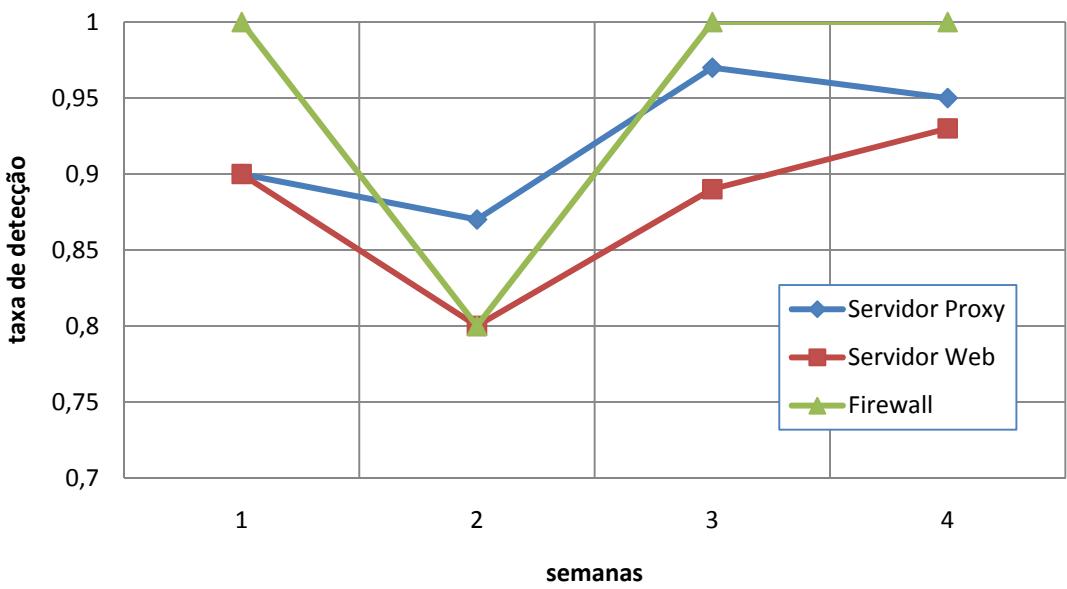


Figura 5.19 - Taxas de detecção para anomalias do grupo B.

As taxas de falsos positivos para o grupo B de anomalias são apresentadas na Figura 5.20. Os resultados para o Proxy variam entre 10 e 15%. As taxas de falsos positivos para o

servidor Web ficaram todas próximas dos 5%. O Firewall apresentou um resultado ruim, próximo a 60%. Como no caso do grupo A de anomalias, este resultado ruim está ligado ao funcionamento do Módulo de Detecção de Anomalias e não ao Módulo de Configuração Automática.

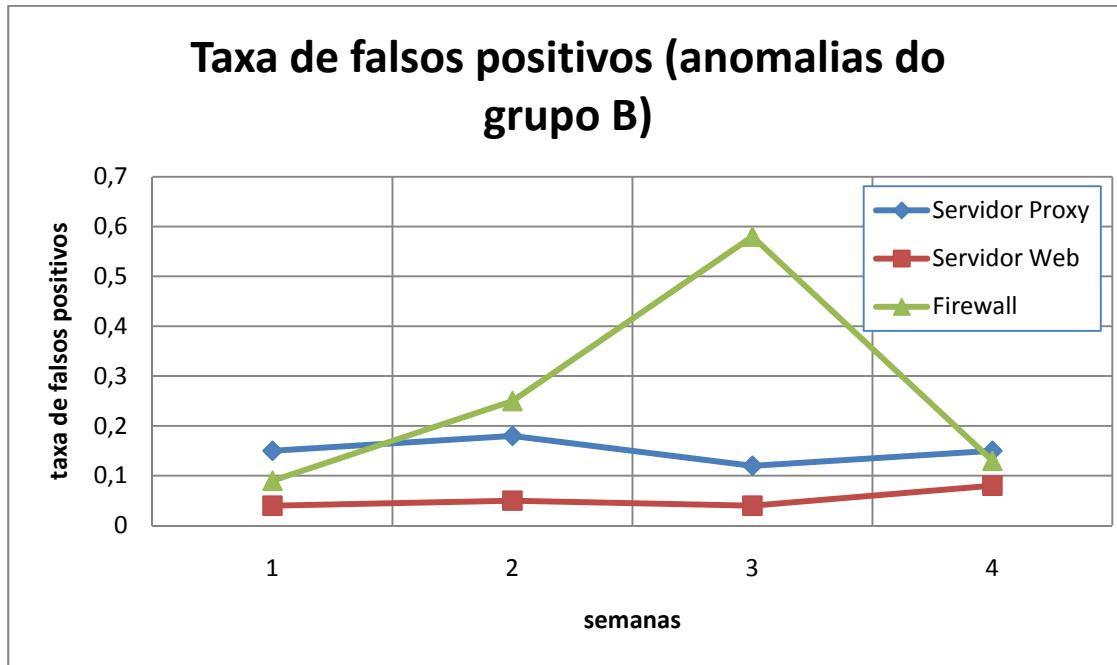


Figura 5.20 - Taxas de falsos positivos para anomalias do grupo B.

A Tabela 5.2 e a Tabela 5.3 apresentam os valores selecionados para os parâmetros pelo algoritmo de configuração durante os testes. Eles mostram que o sistema tem a capacidade de alterar a sua configuração conforme as características das anomalias. Os intervalos de histerese escolhidos para o grupo A foram mais curtos do que aqueles selecionados para o grupo B. Os valores de  $\delta$  foram trabalhados de forma a melhorar o equilíbrio entre as taxas de falsos positivos e as taxas de detecção dentro de cada intervalo de histerese escolhido.

Tabela 5.2 - Parâmetros escolhidos para o grupo A de anomalias.

<i>Semanas</i>	<i>Firewall</i>	<i>Proxy</i>	<i>Servidor web</i>
1	Histerese = 120 segundos	Histerese = 60 segundos	Histerese = 60 segundos
	$\delta = 1$	$\delta = 2$	$\delta = 1$
2	Histerese = 120	Histerese = 60	Histerese = 60

	segundos $\delta = 1$ Histerese = 120 segundos $\delta = 1$ Histerese = 60 segundos $\delta = 1$	segundos $\delta = 2$ Histerese = 60 segundos $\delta = 2$ Histerese = 60 segundos $\delta = 2$	segundos $\delta = 1$ Histerese = 60 segundos $\delta = 1$ Histerese = 60 segundos $\delta = 1$
3			
4			

Tabela 5.3 - Parâmetros escolhidos para o grupo B de anomalias.

<i>Semanas</i>	<i>Firewall</i>	<i>Proxy</i>	<i>Servidor web</i>
1	Histerese = 300 segundos $\delta = 6$	Histerese = 240 segundos $\delta = 6$	Histerese = 240 segundos $\delta = 7$
2	Histerese = 300 segundos $\delta = 6$	Histerese = 240 segundos $\delta = 6$	Histerese = 240 segundos $\delta = 7$
3	Histerese = 300 segundos $\delta = 6$	Histerese = 240 segundos $\delta = 6$	Histerese = 240 segundos $\delta = 7$
4	Histerese = 300 segundos $\delta = 6$	Histerese = 240 segundos $\delta = 6$	Histerese = 240 segundos $\delta = 7$

## 5.6 Casos reais de anomalias

O objetivo desta seção é mostrar dois exemplos reais de anomalias, detalhando o comportamento do sistema de detecção de anomalias frente ao problema. Estes dois exemplos somados envolvem todos os elementos de rede monitorados em nossos testes e trazem duas anomalias bastante comuns no ambiente estudado.

A primeira anomalia que será apresentada ocorreu no dia 25/03/2010, entre as 18h55 e 19h25. Nesta ocorrência, comum na rede da UEL, o servidor Web foi o ponto de origem do tráfego anômalo, que tinha como destino a Internet. Desta forma, foram afetados

também o switch *Black Diamond*, o Firewall e o switch Transit, que estão no caminho de acesso entre o servidor Web e a Internet, conforme é ilustrado na Figura 5.21.

A Figura 5.22 mostra os gráficos relativos aos objetos SNMP *tcpOutSegs* e *ipOutRequests*, que monitoram o fluxo de saída do servidor Web. Nestes gráficos, é possível observar uma significativa diferença entre os dados coletados no objeto SNMP e o DSNS e os alarmes de primeiro nível, os quais são indicados por triângulos pretos. Foram gerados cinco alarmes de segundo nível, que agruparam os dez alarmes de primeiro nível, gerados para os dois objetos já mencionados. Os alarmes de segundo nível indicaram uma anomalia no fluxo de saída do servidor Web.

O tráfego anômalo partiu do servidor Web para o switch do núcleo da rede da UEL, o switch BD. Neste switch, o tráfego anômalo ingressou através da porta 3016, que recebe a conexão do servidor Web. A Figura 5.23 mostra o gráfico para o objeto *ifInOctets* na porta 3016 do switch BD. Foram gerados cinco alarmes de primeiro nível e, consequentemente, foram gerados também cinco alarmes de segundo nível, indicando uma anomalia no fluxo de entrada da porta 3016.

A Figura 5.24 mostra que o tráfego anômalo partiu em direção ao Firewall através da porta 3001 do switch BD, após ingressar neste switch através da porta 3016. O gráfico mostra a significativa diferença entre os dados coletados no objeto *ifOutOctets* e o seu respectivo DSNS. Foram gerados 4 alarmes de primeiro nível, que, consequentemente, causaram a geração de 4 alarmes de segundo nível.

A Figura 5.25 mostra o tráfego anômalo chegando ao Firewall e sendo encaminhado para outro ponto da rede. Temos os gráficos da movimentação dos objetos *ipInReceives* e *ipForwDatagrams*, mostrando uma forte diferença entre os dados reais coletados e os respectivos DSNS. Foram gerados 7 alarmes de primeiro nível para os dois objetos. Estes 7 alarmes de primeiro nível resultaram em 3 alarmes de segundo nível, os quais apontaram uma anomalia no fluxo de encaminhamento do Firewall.

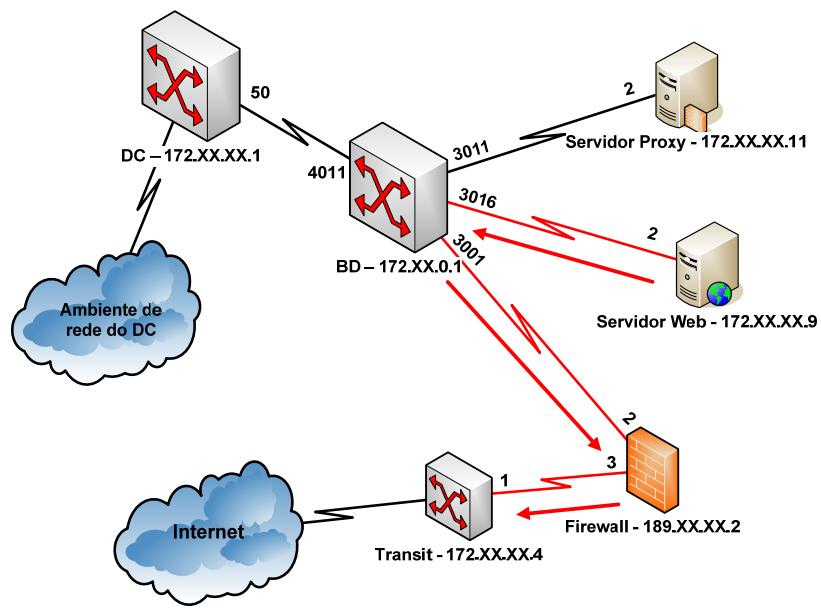


Figura 5.21 - Visão da rede que mostra a propagação da primeira anomalia apresentada.

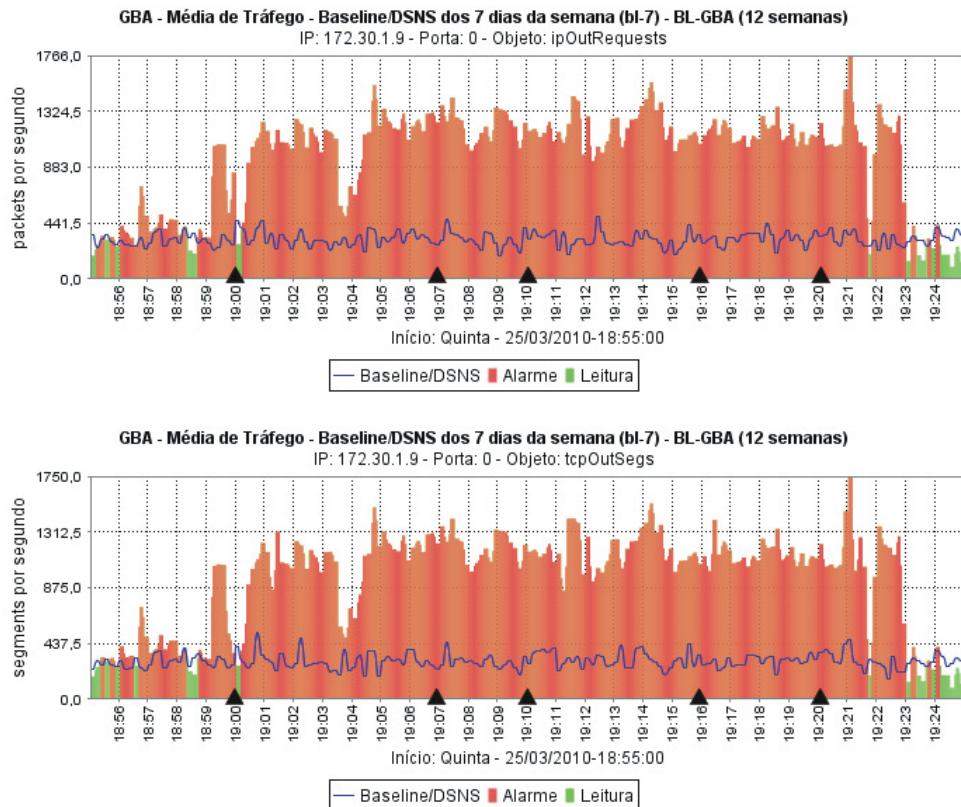


Figura 5.22 - Alarmes de primeiro nível nos objetos *ipOutRequests* e *tcpOutSegs* do servidor Web.

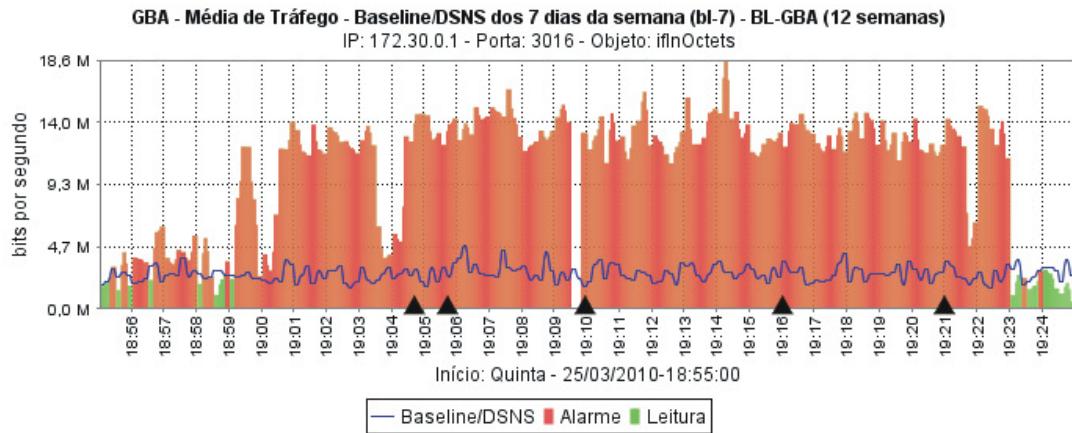


Figura 5.23 - Alarmes de primeiro nível no objeto *ifInOctets*, porta 3016, no switch BD.

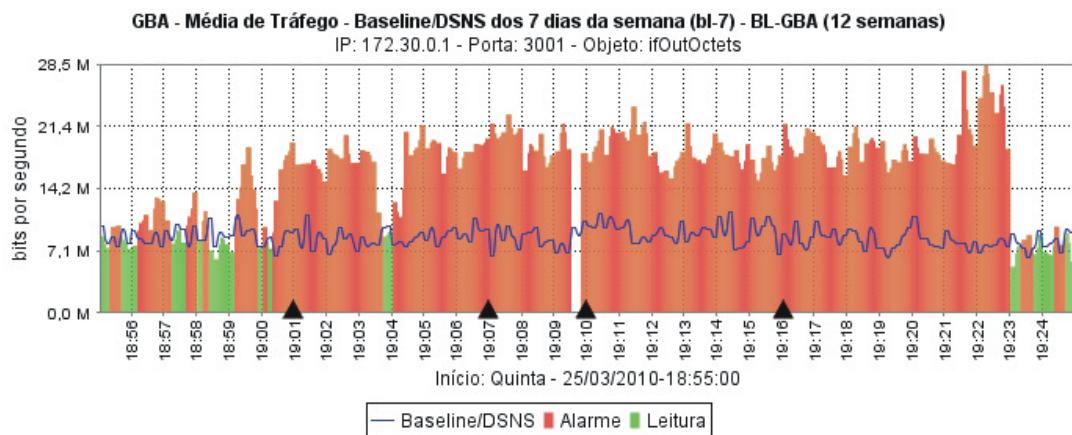


Figura 5.24 - Alarmes de primeiro nível no objeto *ifOutOctets*, porta 3001, no switch BD.

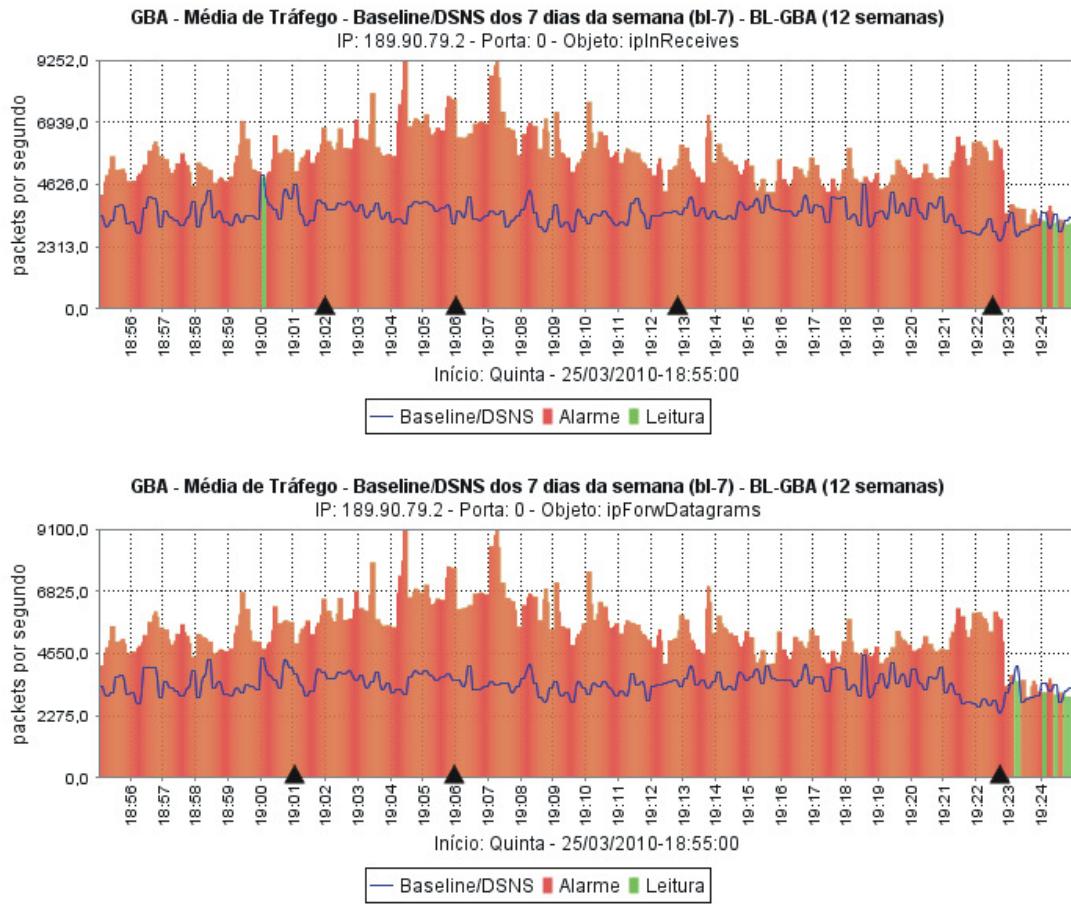


Figura 5.25 - Alarmes de primeiro nível nos objetos *ipInReceives* e *ipForwDatagrams* no Firewall.

A Figura 5.26 mostra o tráfego anômalo ingressando no switch Transit, antes de ser encaminhado para a Internet. No gráfico do objeto *ifInOctets*, na porta 1, é possível observar também a grande diferença entre os dados coletados e o DSNS. Foram gerados 5 alarmes de primeiro nível, os quais causaram a geração de 5 alarmes de segundo nível, indicando uma anomalia no fluxo de entrada da porta 1 do switch Transit.

Ao todo, para este caso de anomalia, foram gerados 31 alarmes de primeiro nível, que se transformaram em 22 alarmes de segundo nível. Estes alarmes resultaram em apenas dois alarmes de terceiro nível, que apresentaram toda a propagação da anomalia pela rede. A Figura 5.27 mostra os grafos de dependências e como a anomalia se comportou. O rótulo (1), na figura, destaca a propagação da anomalia no fluxo de saída do servidor Web. O rótulo (2) destaca a propagação do tráfego anômalo do servidor Web para o switch BD. O

rótulo (3) mostra o tráfego anômalo saindo do switch BD e ingressando no Firewall, enquanto o rótulo (4) destaca a propagação da anomalia pelo fluxo de encaminhamento do Firewall. Finalmente, temos o rótulo (5) destacando a entrada do tráfego anômalo no Switch Transit, que está ligado à Internet.

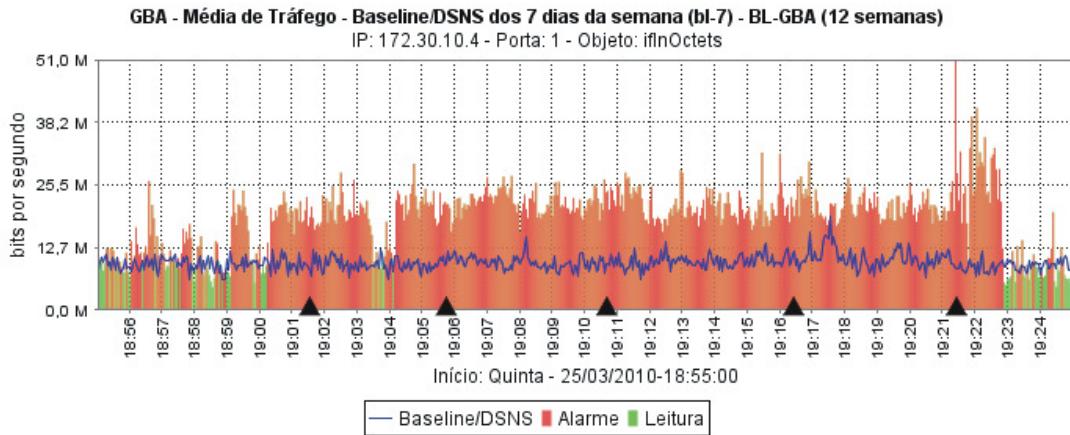


Figura 5.26 - Alarmes de primeiro nível no objeto *ifInOctets*, porta 1, switch Transit.

A segunda anomalia a ser estudada ocorreu no dia 16/03/2010, entre as 19h00 e 20h15. Esta anomalia envolveu um tráfego acima do normal ingressando na rede da UEL a partir da Internet e tendo como destino a rede local do Departamento de Computação (DC). Primeiramente foram afetados o switch Transit e o Firewall. O tráfego anômalo ingressou, então, no switch BD para ser encaminhado ao servidor Proxy. Após ser processado no servidor Proxy, o tráfego retornou ao switch BD para finalmente ser encaminhado ao seu destino, a rede local do DC da UEL. Este cenário é ilustrado na Figura 5.28.

A Figura 5.29 mostra o tráfego anômalo saindo do switch Transit em direção ao Firewall. O gráfico para o objeto *ifOutOctets* na porta 1 mostra que os dados coletados na MIB são muitos superiores ao valores encontrados no DSNS. Foram gerados, ao todo, 14 alarmes de primeiro nível, que causaram a geração de 14 alarmes de segundo nível, indicando uma anomalia no fluxo de saída da porta 1 do switch Transit.

A Figura 5.30 apresenta os alarmes de primeiro nível gerado no Firewall para esta ocorrência. Ao todo foram gerados 19 alarmes de primeiro nível, sendo 9 alarmes para o objeto *ipInReceives* e 10 alarmes para o objeto *ipForwDatagrams*. Como consequência,

foram gerados 9 alarmes de segundo nível, reportando uma anomalia no fluxo de encaminhamento do Firewall.

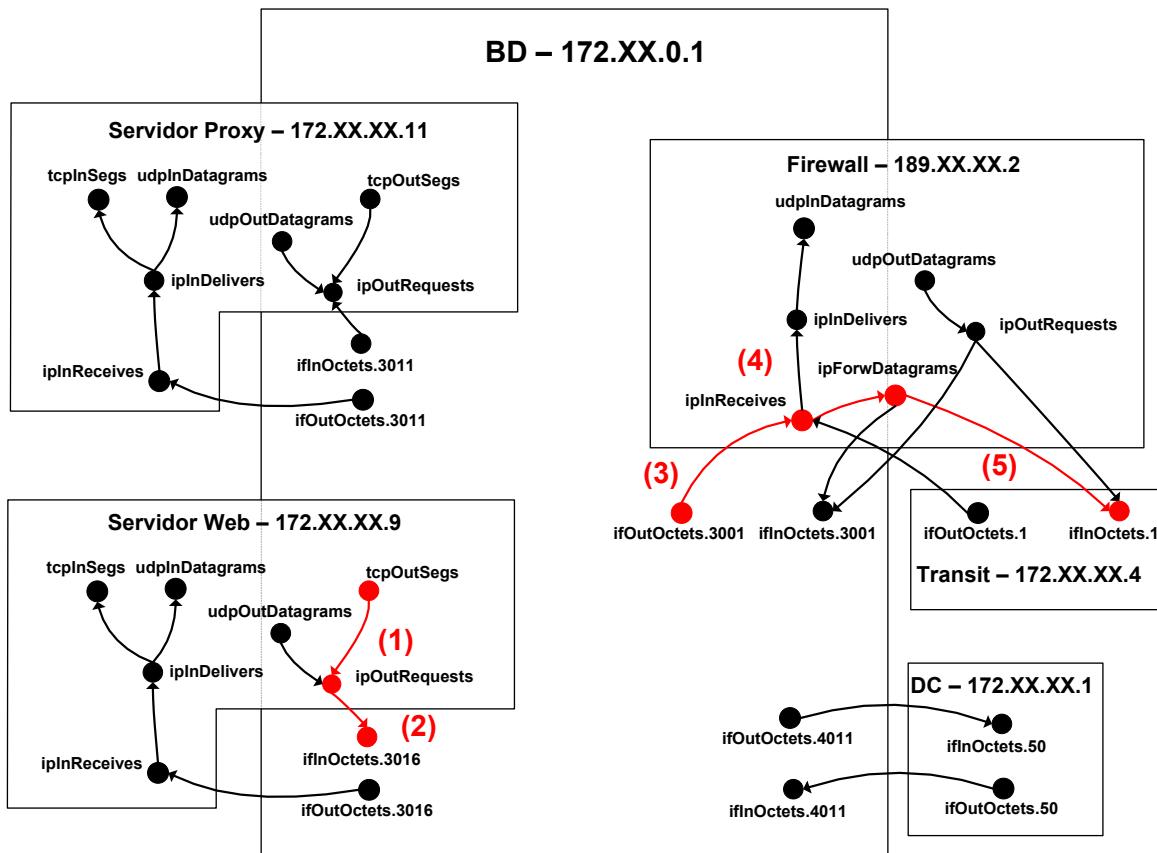


Figura 5.27 - Cenário de propagação do primeiro caso de anomalia.

A Figura 5.31 mostra que o tráfego anômalo está ingressando no switch BD, após atravessar o Firewall. São apresentados os 14 alarmes de primeiro nível gerados para o objeto *ifInOctets*, na porta 3001, que está conectada ao Firewall. Consequentemente, foram gerados 14 alarmes de segundo nível reportando uma anomalia de entrada da porta 3001 do switch BD.

A Figura 5.32 mostra o tráfego anômalo saindo do switch BD, em direção ao servidor Proxy. Foram gerados 15 alarmes de primeiro nível para o objeto *ifOutOctets*, na porta 3011, que está conectada ao servidor Proxy. Consequentemente, foram gerados 15 alarmes de segundo nível, destacando uma anomalia de saída na porta 3011 do switch BD.

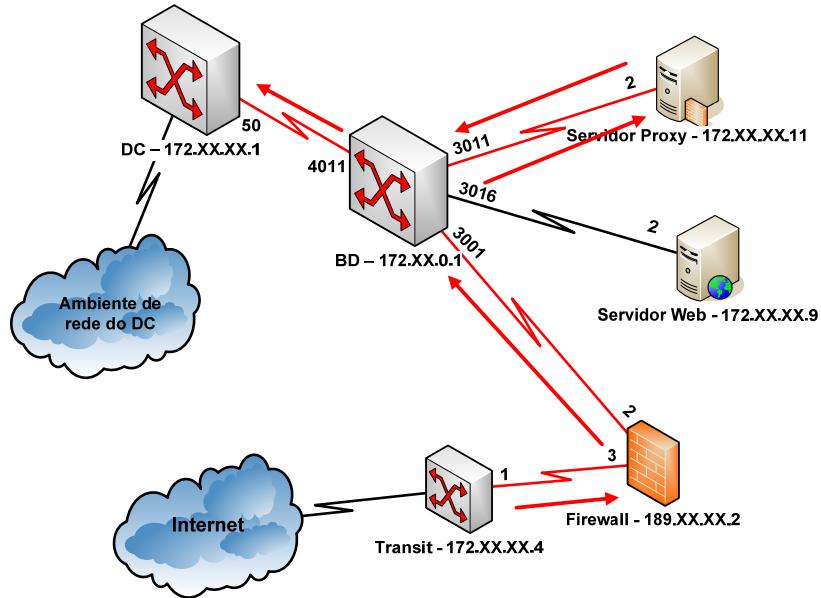


Figura 5.28 - Visão da rede que mostra a propagação da segunda anomalia apresentada.

A Figura 5.33 mostra a detecção da anomalia no fluxo de entrada do servidor Proxy. Nesta figura, temos os alarmes de primeiro nível gerados para os objetos *ipInReceives*, *ipInDelivers* e *tcpInSegs*. No total, foram gerados 41 alarmes de primeiro nível. A correlação destes alarmes resultou em apenas 13 alarmes de segundo nível.

A Figura 5.34 mostra novamente a detecção da anomalia no servidor Proxy, porém agora localizada em seu fluxo de saída. Todo o tráfego que ingressa na rede da UEL e tem como origem a Internet atravessa o Proxy, o qual realiza uma análise para bloquear conteúdos impróprios e implementa um mecanismo de *cache* das páginas mais acessadas. Por isso, o tráfego anômalo ingressou no Proxy, como foi apresentado na Figura 5.33, foi processado e agora está deixando o Proxy para voltar à rede da UEL. Os gráficos mostram 29 alarmes de primeiro nível gerados para os objetos *ipOutRequests* e *tcpOutSegs*. Após a correlação destes alarmes de primeiro nível, foram gerados 13 alarmes de segundo nível.

A Figura 5.35 complementa o cenário de propagação da anomalia pelo Servidor Proxy. O gráfico mostra os 14 alarmes de primeiro nível gerados para o objeto *ifInOctets* na porta 3011 do switch BD, que é responsável por conectar este switch ao servidor Proxy. Consequentemente, foram gerados 14 alarmes de segundo nível, que apontam uma anomalia no fluxo de entrada da porta 3011 do switch BD.

A Figura 5.36 e a Figura 5.37 mostram a propagação da anomalia do switch BD para o switch do Departamento de Computação. O gráfico da Figura 5.36 mostra os 15 alarmes de primeiro nível gerados para o objeto *ifOutOctets* na porta 4011 do switch BD, que é responsável por conectar este equipamento ao switch do Departamento de Computação. Foram gerados 15 alarmes de segundo nível apontando uma anomalia no fluxo de saída da porta 4011 do switch BD. A Figura 5.37 traz os 15 alarmes de primeiro nível gerados para o objeto *ifInOctets* na porta 50 do switch do DC. Eles resultaram em 15 alarmes de segundo nível que indicam uma anomalia no fluxo de entrada da porta 50 do switch do DC.

Neste estudo de caso, foi gerado um total de 176 alarmes de primeiro nível, 122 alarmes de segundo nível e apenas 15 alarmes de terceiro nível. A correlação realizada no segundo nível de análise promove uma redução de cerca de 30% no total de alarmes, enquanto a correlação de terceiro nível promove uma redução de 87% do total de alarmes em relação à quantidade de alarmes de segundo nível e de 91% em relação à quantidade de alarmes de primeiro nível. Além da forte redução na quantidade de alarmes que são enviados ao administrador de rede, é importante observar que os alarmes de terceiro nível mostram um cenário completo da anomalia, facilitando a visualização e a solução do problema.

A Figura 5.38 mostra os grafos de dependências de todos os equipamentos monitorados e como a anomalia se propagou através deles, resultado da análise em três níveis realizada em nossa proposta. A indicação (1) mostra a propagação da anomalia do switch Transit, ligado à Internet, para o Firewall da UEL. A indicação (2) mostra a propagação da anomalia dentro do Firewall, sinalizando uma anomalia no fluxo de encaminhamento deste equipamento. A indicação (3) mostra a propagação da anomalia do Firewall para o switch BD, enquanto a indicação (4) traz a propagação da anomalia do switch BD para o servidor Proxy. Em (5), podemos ver o comportamento da anomalia no fluxo de entrada do servidor Proxy, enquanto em (6) é destacada a anomalia no fluxo de saída do servidor Proxy, retornando à rede. A indicação (7) traz a volta da anomalia para o switch BD. Finalmente, temos em (8) e (9) a propagação da anomalia do switch BD para seu destino, o switch do Departamento de Computação.

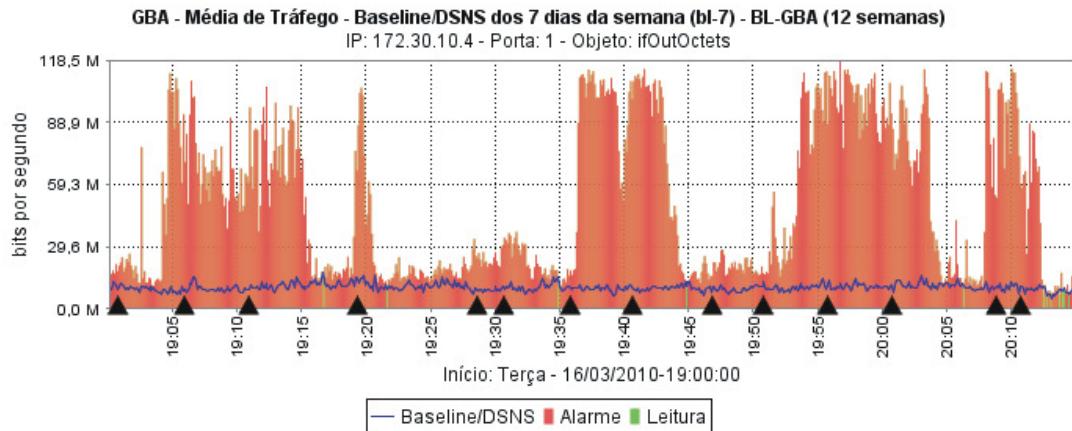


Figura 5.29 - Alarmes de primeiro nível no objeto *ifOutOctets*, porta 1, switch Transit.

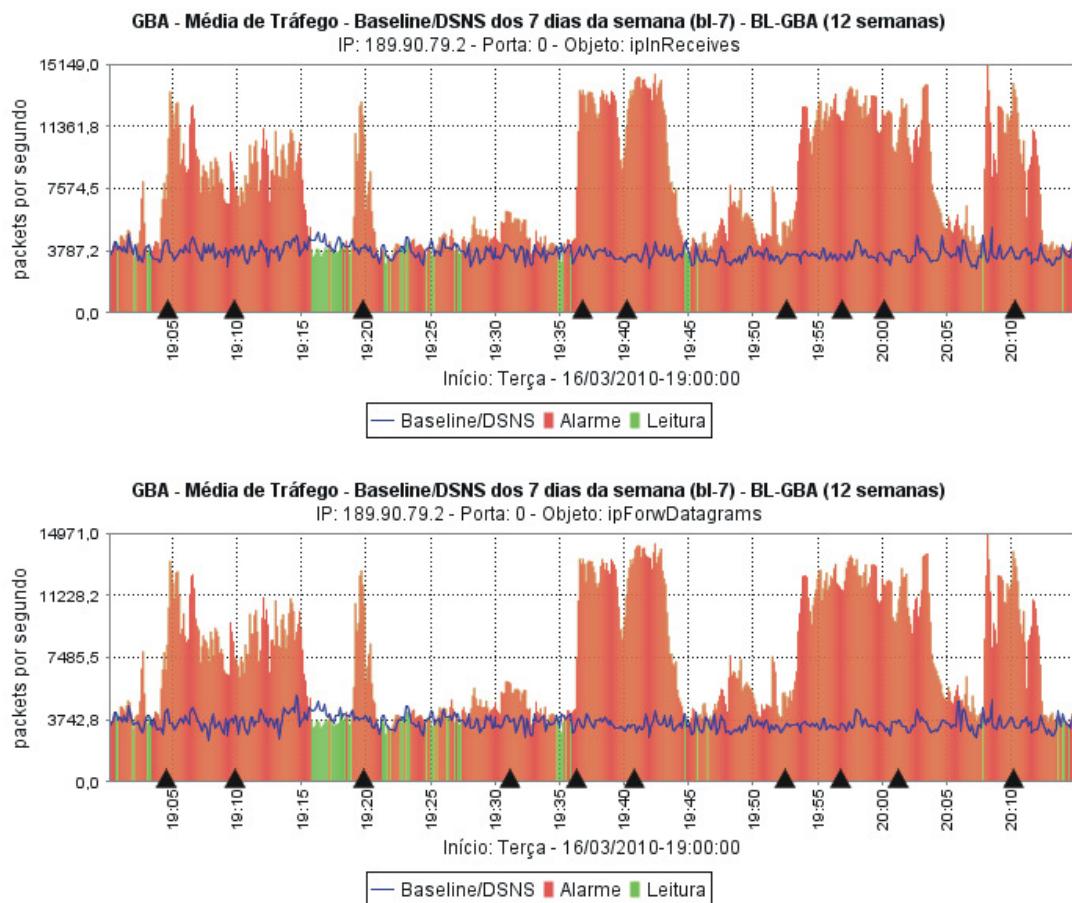


Figura 5.30 - Alarmes de primeiro nível nos objetos *ipInReceives* e *ipForwDatagrams* do Firewall.

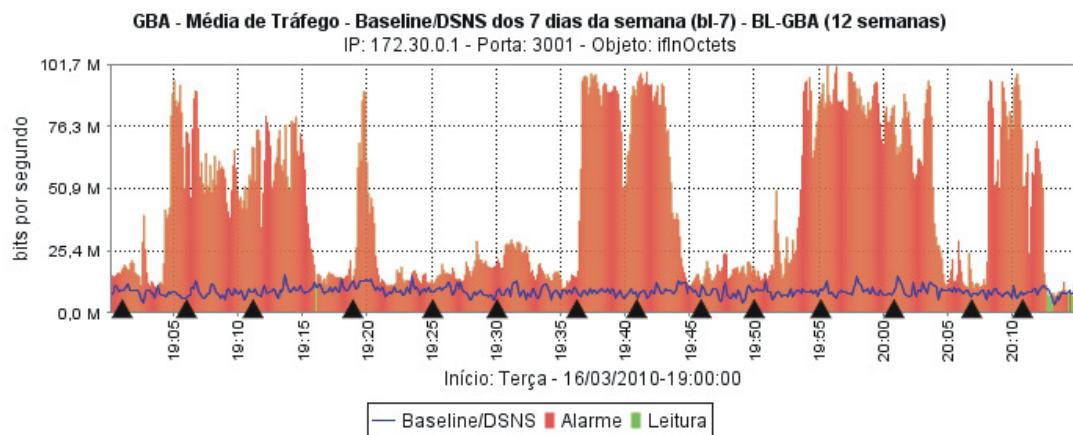


Figura 5.31 - Alarmes de primeiro nível no objeto *ifInOctets*, porta 3001, switch BD.

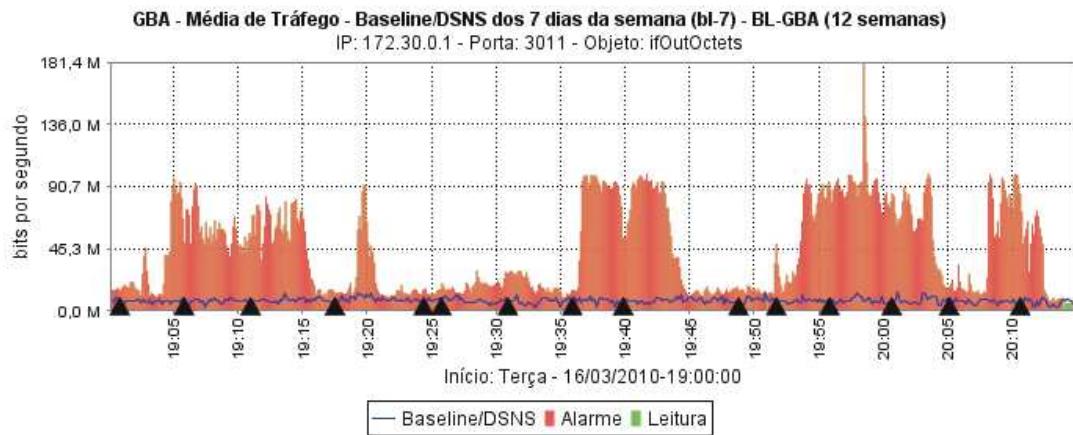


Figura 5.32 - Alarmes de primeiro nível no objeto *ifOutOctets*, porta 3011, switch BD.

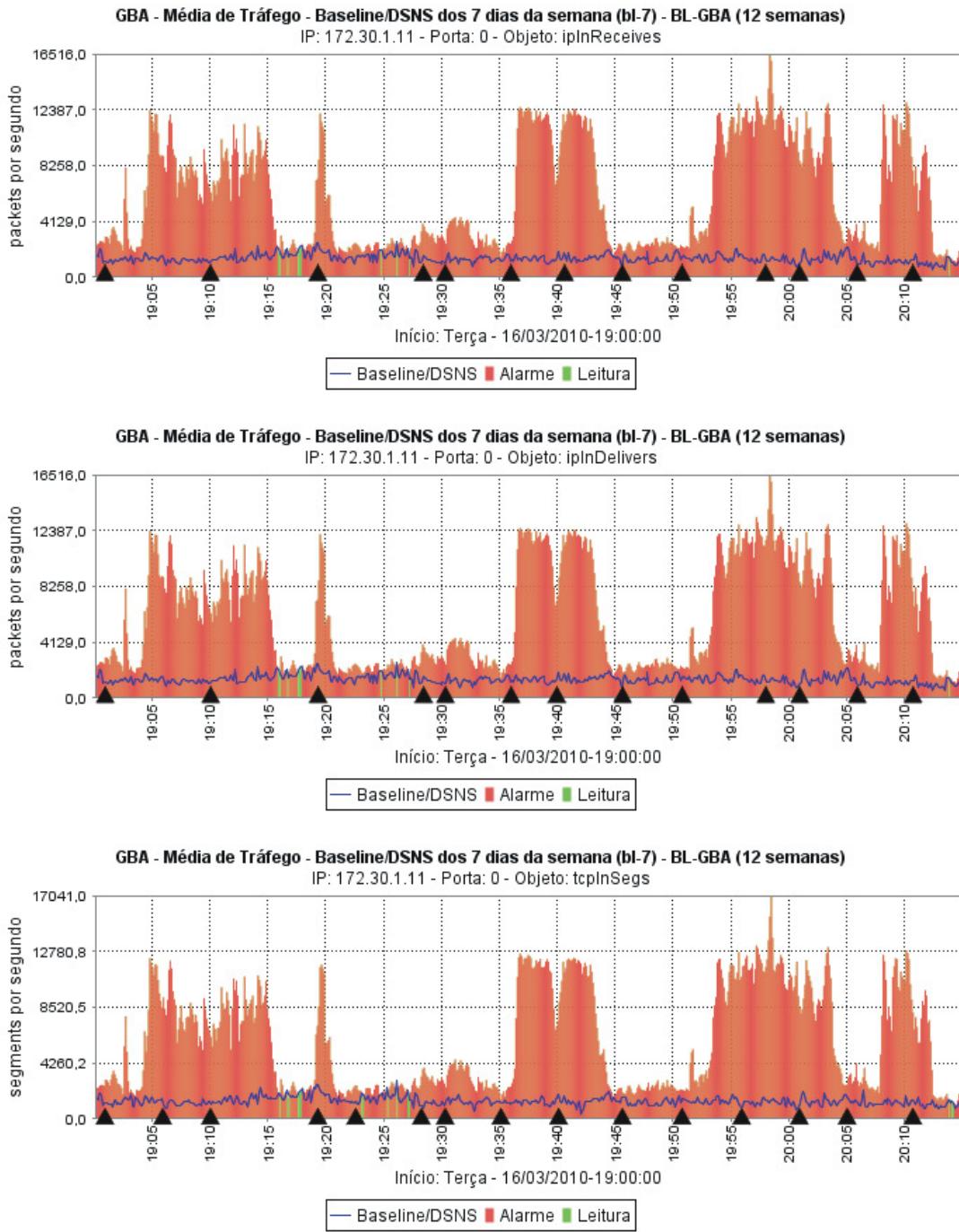


Figura 5.33 - Alarmes de primeiro nível nos objetos *ipInReceives*, *ipInDelivers* e *tcpInSegs*, servidor Proxy.

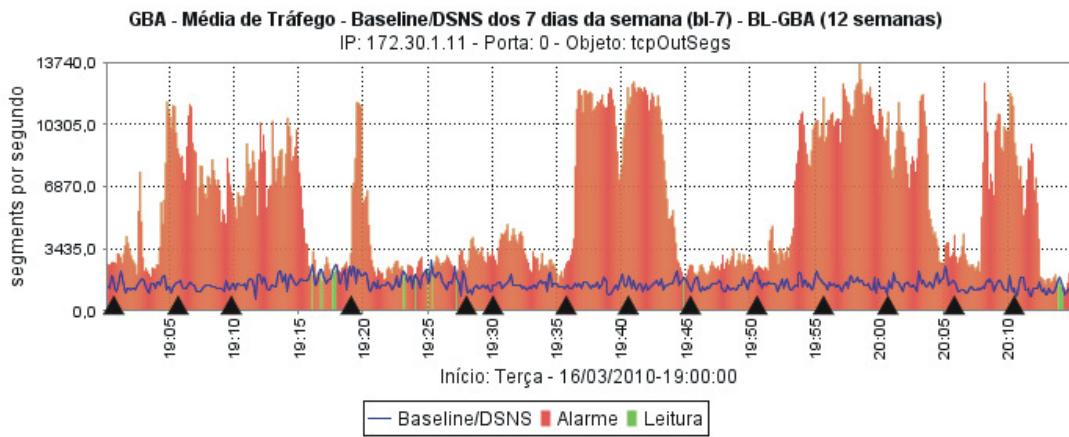
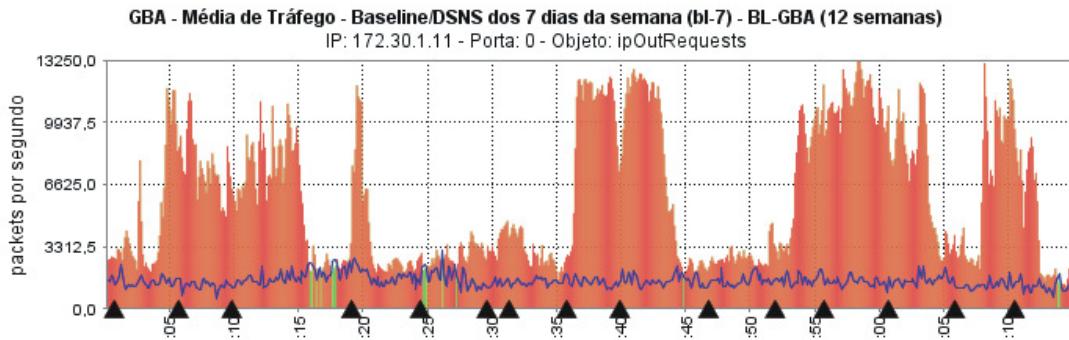


Figura 5.34 - Alarmes de primeiro nível nos objetos *ipOutRequests* e *tcpOutSegs*, servidor Proxy.

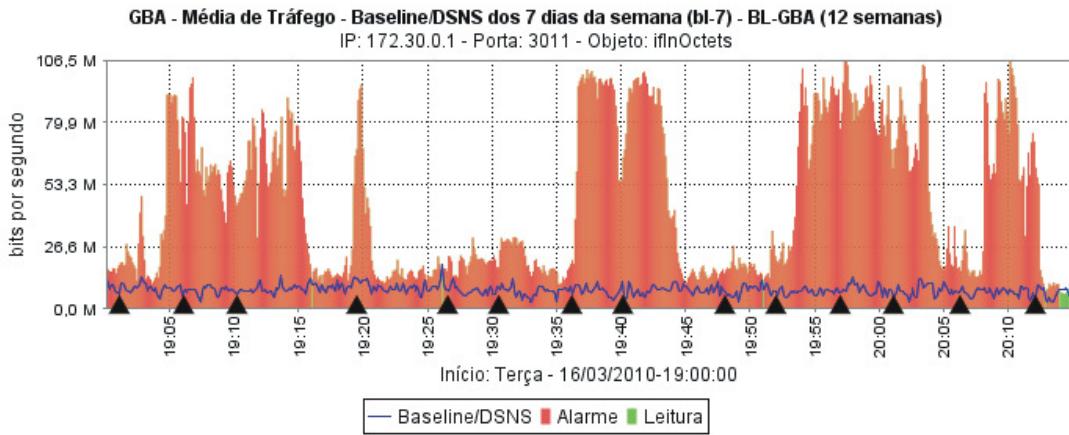


Figura 5.35 - Alarmes de primeiro nível no objeto *ifInOctets*, porta 3011, switch BD.

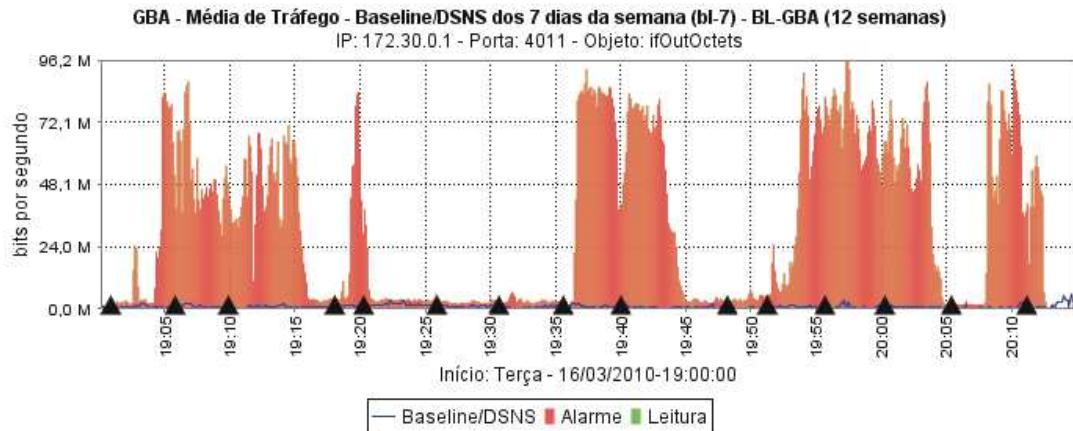


Figura 5.36 - Alarmes de primeiro nível no objeto *ifOutOctets*, porta 4011, switch BD.

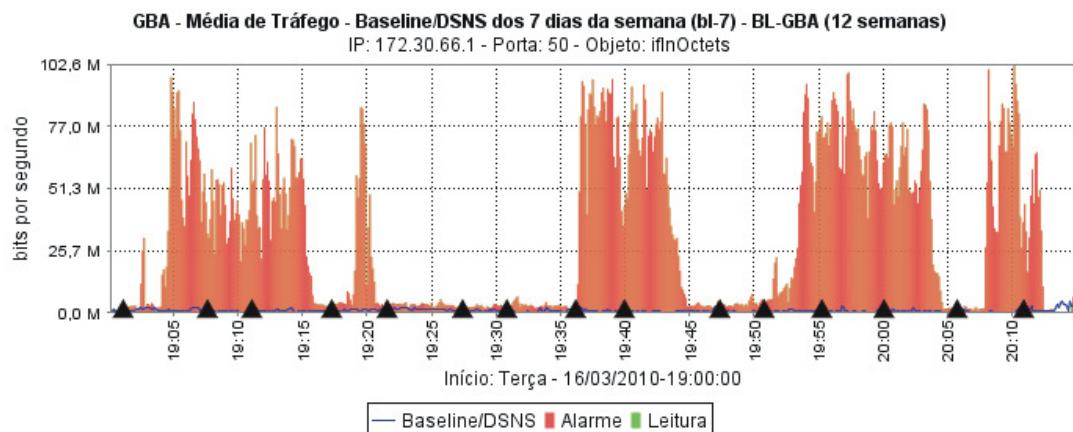


Figura 5.37 - Alarmes de primeiro nível no objeto *ifInOctets*, porta 50, switch do Departamento de Computação.

Após apresentar estes exemplos, a principal conclusão obtida é que, de fato, a ocorrência de uma anomalia acaba envolvendo diferentes elementos de rede e causando a geração de vários alarmes redundantes. Se fosse proposta uma abordagem que apenas monitorasse objetos SNMP individualmente, seriam gerados 207 alarmes, sem maiores conclusões do que estaria ocorrendo. Ao correlacioná-los, foram obtidos 144 alarmes de segundo nível, que já mostravam os equipamentos que participavam da anomalia e o comportamento de cada um deles. Ao realizar a análise de terceiro nível, o número de alarmes foi diminuído para 17 e foi apresentado ao administrador de rede o comportamento

do problema em toda a rede, com a geração de relatório mais completo, que facilita a solução do problema.

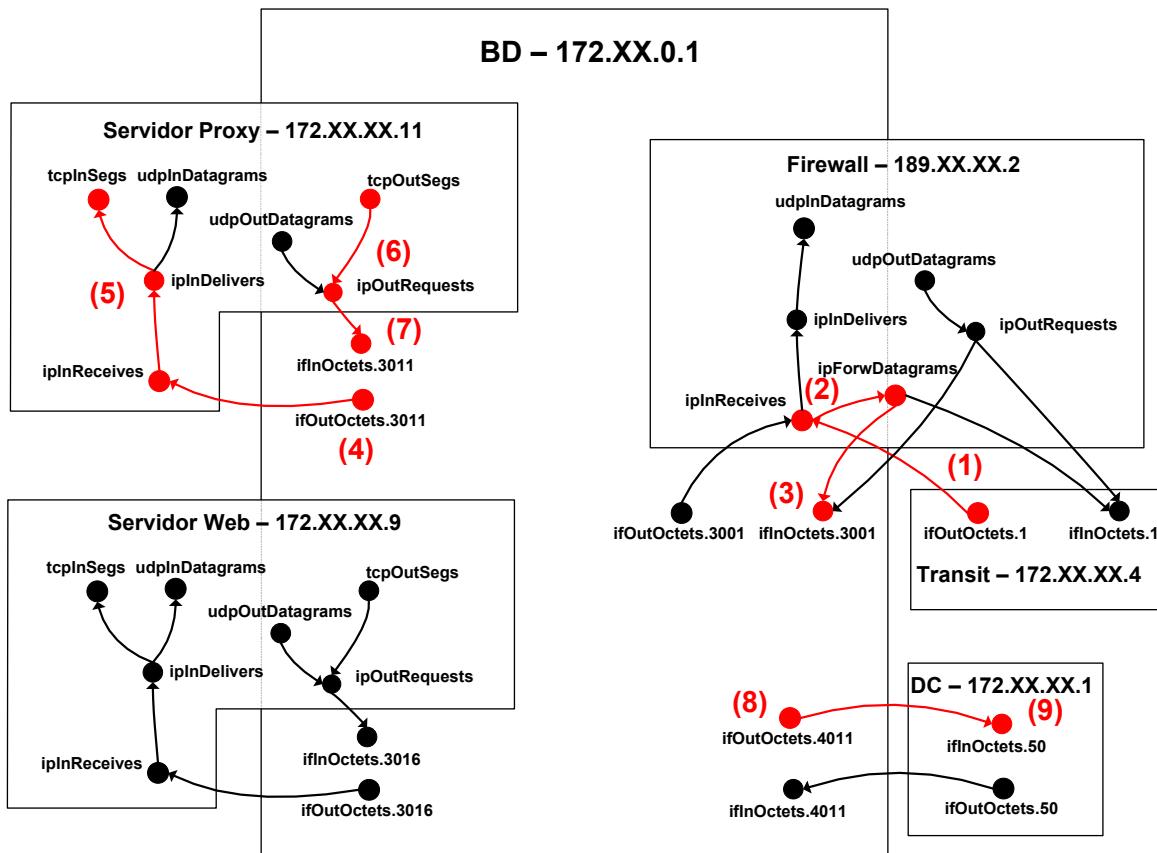


Figura 5.38 - Cenário de propagação do segundo caso de anomalia.

## 6 Conclusão

O principal objetivo deste trabalho foi a construção de uma proposta para detecção de anomalias baseada na caracterização do comportamento normal da rede. Ela utiliza dados coletados em objetos SNMP para detectar as anomalias e proporcionar ao administrador de rede uma visão panorâmica do problema, evitando que ele seja sobrecarregado por alarmes redundantes que são gerados em diferentes pontos da rede e reportam o mesmo problema. O sistema proposto também teve como objetivo disponibilizar facilidades para a configuração de seus níveis de sensibilidade a fim de tornar possível a adaptação a diferentes políticas de gerência. As principais contribuições obtidas com esta proposta foram:

- **Grafo de dependências para objetos SNMP da MIB-II:** este grafo permite que seja mapeado o comportamento da anomalia dentro do elemento de rede. O cruzamento dos mapas de comportamento gerados para diferentes elementos interconectados permite que tenhamos a visão panorâmica da propagação da anomalia, ajudando o administrador de rede a encontrar a origem do problema. A análise individual de objetos SNMP leva à geração de vários alarmes redundantes e com pouco significado. O grafo de dependências, por sua vez, permite que seja realizada a correlação das causas destes alarmes, diminuindo a quantidade final de notificações que chegam ao administrador de redes. É importante ressaltar que estas notificações finais agregam as informações de todos os alarmes que foram correlacionados, apresentando ao administrador a propagação da anomalia pela rede de forma a apoiar a solução do problema;
- **Configuração da sensibilidade do sistema de detecção:** o sistema proposto nesta tese permite a alteração do seu nível de sensibilidade perante os desvios de comportamento. Esta característica é importante por permitir que o sistema de detecção se adapte à política de gerência que o administrador deseja adotar na rede analisada. Além disso, o sistema inclui um algoritmo

- para a realização da configuração automática do nível de sensibilidade, diminuindo a necessidade de intervenção humana;
- **Desenvolvimento e testes em um ambiente real de rede:** a proposta foi desenvolvida e implementada na ferramenta GBA, utilizando dados reais coletados da rede da Universidade Estadual de Londrina (UEL) para a realização de testes que validaram a proposta;
  - **Levantamento de métodos e trabalhos relacionados:** foi apresentado um extenso levantamento das técnicas normalmente empregadas na detecção de anomalias e dos diversos trabalhos científicos que fizeram uso destas técnicas para propor os sistemas de detecção;

O sistema proposto nesta tese foi implementado dentro da ferramenta GBA, utilizando a linguagem Java e a especificação EJB 3 (EJB, 2010). Alguns pontos da proposta tiveram que ser adaptados a fim de realizar a sua implantação no ambiente real da rede da UEL. Não foi possível monitorar os 35 objetos do grafo de dependências simultaneamente em um elemento de rede, já que o agente SNMP não seria capaz de oferecer respostas a tantas requisições com intervalos de poucos segundos. Por isso, foi escolhido um conjunto reduzido de objetos, focando naqueles que monitoram a transição entre as camadas do protocolo TCP/IP. Além disso, não foi possível monitorar os objetos *ifInOctets* e *ifOutOctets* nos servidores, pois os dados não podiam ser obtidos em um intervalo de coleta menor que 30 segundos, o que dificulta a detecção de anomalias. Todas estas questões foram equacionadas e estas experiências foram devidamente registradas na seção 5.3 deste trabalho, para que futuros trabalhos tirem proveito destas conclusões.

Com relação aos testes realizados, o primeiro ponto abordado foi o potencial de detecção, avaliado por meio de gráficos conhecidos como curvas ROC (SOULE et al., 2005). A fim de construir as curvas ROC, o sistema de detecção foi aplicado com diferentes níveis de sensibilidade. Para cada um dos níveis de sensibilidade testados, foi coletada a taxa de falsos positivos e a taxa de detecção. A reunião destes resultados em um único gráfico formou as curvas ROC. Além disso, os testes foram realizados sobre dois grupos distintos de anomalias, a fim de simular políticas de gerência diferentes. O grupo A de anomalias representou a política de gerência na qual o administrador de rede deseja ser informado sobre a ocorrência da grande maioria dos desvios, enquanto o grupo B de

anomalias representava uma política de gerência mais permissiva, na qual apenas desvios de comportamento mais significativos deveriam ser considerados anomalias. Os testes foram realizados durante o mês de abril de 2009 em três equipamentos de rede da UEL: servidor Proxy, servidor Web e Firewall.

Nos testes realizados no servidor Proxy, as melhores taxas de detecção para os grupos A e B de anomalias ficaram em torno de 80%, acompanhadas de taxas de falsos positivos de aproximadamente 10%. Nos testes do servidor Web, os resultados para o grupo A de anomalias alcançaram taxas de detecção de 90% e taxas de falsos positivos iguais a 15%. Considerando o grupo B de anomalias, os resultados para o servidor Web foram muito bons, alcançando uma taxa de detecção de 87% combinada com uma taxa de falsos positivos de apenas 6%. Para o Firewall, os resultados não foram tão bons quanto nos outros dois servidores. As taxas de detecção encontradas para o grupo A de anomalias foram baixas, ficando em torno de 60%, acompanhadas por taxas de falsos positivos em torno de 15%. Para o grupo B, os resultados foram melhores, com taxas de detecção próximas a 70%, acompanhadas por taxas de falsos positivos próximas a 10%. Comparando as curvas ROC geradas para os diferentes grupos de anomalias, é possível observar que os resultados foram levemente superiores para o grupo B de anomalias. Isto ocorre porque as anomalias do grupo B são caracterizadas por anomalias de duração mais longa e que se distanciam de maneira mais evidente do comportamento normal, sendo mais fáceis de serem detectadas.

Outro ponto avaliado durante os testes foi a configuração automática dos parâmetros do sistema. Novamente, os testes foram realizados considerando os grupos de anomalias A e B. Os resultados demonstraram que o sistema foi capaz de se adaptar às necessidades do administrador de redes ao selecionar configurações de parâmetros que ofereceram boas taxas de falsos positivos e de detecção. As taxas de detecção resultantes das configurações automaticamente escolhidas pelo sistema ficaram na maioria dos casos entre 80 e 100%, enquanto as taxas de falsos positivos ficaram entre 5 e 20%. Estas taxas estão próximas aos melhores resultados encontrados nos testes de potencial de detecção, mostrando que a proposta de configuração automática foi eficaz.

Nos dois casos de anomalias reais apresentados, foi observado que a propagação do tráfego anômalo afeta objetos de diferentes equipamentos da rede de maneira semelhante.

Utilizando os grafos de dependência dos objetos SNMP e os dados sobre a topologia da rede, o sistema explorou as dependências entre os objetos e equipamentos monitorados e foi capaz de indicar ao administrador de rede, nos dois casos, como a anomalia estava se propagando.

O processamento em três níveis dos dados coletados nos diversos objetos SNMP resultou em uma pequena quantidade de alarmes, que apresentaram, por outro lado, informações bastante significativas. Se fossem analisados apenas os objetos SNMP individualmente, o administrador de redes teria recebido 207 alarmes nos dois casos estudados. Realizando a análise em três níveis, foram gerados 17 alarmes, que resumem as informações dos 207 alarmes gerados para os objetos. A utilização dos grafos de dependências melhorou a qualidade e diminuiu a quantidade dos alarmes gerados, evitando a sobrecarga do administrador de rede.

Como sugestão de trabalho futuro, pode-se explorar objetos relacionados a erros e descarte de pacotes, que trazem informações importantes e necessitam de tratamento especial por terem um comportamento bem diferente dos outros objetos. O fato dos objetos relacionados a erros e descartes apresentarem um comportamento ainda mais instável que os outros objetos faz com que seja muito difícil definir perfis de comportamento normal. O modelo BLGBA, por exemplo, não é eficiente nestes casos.

No sistema de detecção de anomalias proposto neste trabalho, a correlação dos alarmes de primeiro nível leva à geração de alarmes de segundo nível somente se a anomalia percorrer um caminho completo dentro do grafo de dependências, que se inicia em um ponto inicial de monitoramento e se encerra em um ponto final de monitoramento. Sugere-se como trabalho futuro o estudo do impacto causado pela geração de alarmes para situações onde há apenas a propagação parcial da anomalia no grafo de dependências.

O trabalho de correlação de alarmes no terceiro nível também pode ser estendido. Estes trabalhos futuros apontam para a utilização de técnicas que possam melhorar este processo de correlação, montando matrizes de tráfego que levem em conta a função e as características dos equipamentos, de forma a trazer diagnósticos cada vez mais completos ao administrador de rede. A realização de verificações extras nos elementos afetados para auxiliar na tomada automática de decisões também forma um campo promissor de

pesquisa. A utilização do sistema de detecção de anomalias proposto nesta tese pode ser um primeiro passo na construção de sistemas autogerenciáveis, que sejam capazes, por exemplo, de restringir automaticamente os recursos disponíveis para um usuário quando há abusos.

Por fim, podemos dizer que o modelo de dependências de objetos SNMP proposto neste trabalho não se restringe aos objetos contidos na MIB-II. Outras MIBs podem ser modeladas da mesma forma para que os comportamentos das anomalias sejam mapeados em diversos ambientes e situações.

## **7 Bibliografia**

- ABUSINA, Z. U. M.; ZABIR, S. M. S.; ASHIR, A., CHAKRABORTY, D.; SUGANUMA, T.; SHIRATORI, N. **An Engineering Approach to Dynamic Prediction of Network Performance from Application Logs**, International Journal of Network Management, v. 15, p. 151-162, 2005.
- AGRAWAL, S.; NAIDU, K. V. M.; RASTOGI, R. **Diagnosing Link-level Anomalies using Passive Probes**, Proceedings of IEEE INFOCOM 2007, p. 1757-1765, 2007.
- AL-KASASSBEH, M.; ADDA, M. **Network fault detection with Wiener filter-based agent**, Journal of Network and Computer Applications, v. 32, n. 4, p. 824-833, 2009.
- ANDERSON, J. P. **Computer security threat monitoring and surveillance**, James P. Anderson Co., Fort, Washington, PA, USA, Technical Report 98-17, 1980.
- ANDROULIDAKIS, G.; CHATZIQIANNAKIS, V.; PAPAVASSILIOU, S. **Network anomaly detection and classification via opportunistic sampling**, IEEE Network, V. 23, n. 1, pp. 6-12, 2009.
- BARFORD, P.; KLINE, J.; PLONKA, D.; RON, A. **A signal analysis of network traffic anomalies**, Internet Measurement Workshop; Proceedings of the second ACM SIGCOMM Workshop on Internet measurement, Marseille, France, Pages: 71 – 82, 2002, ISBN:1-58113-603-X.
- CASE, J. D.; PARTRIDGE, C. **Case Diagrams: A First Step to Diagrammed Management Information Bases**, ACM SIGCOMM Computer Communication Review, v. 19, issue 1, ACM Press New York, NY, USA, 1989.
- CHAO, C. S.; YANG, D. L.; LIU, A. C. **A Time-aware Fault Diagnosis System in LAN**, Proceedings of IEEE/IFIP International Symposium on Integrated Network Management, 2001, p. 499-512, maio 2001.
- COATES, M.; HERO III, A. O.; NOWAK, R.; YU, B. **Internet Tomography**, IEEE Signal Processing Magazine, v. 19, n. 3, p. 47-65, 2002.

- COLE, E.; KRUTZ, R.; CONLEY, J. W. **Network Security Bible**, Wiley Publishing, 2005.
- DENNING, D. E. **An Intrusion Detection Model**, IEEE Transactions on Software Engineering, v. 13, n. 2, p. 222-232, 1987.
- EJB - Enterprise JavaBeans Technology**, disponível via WEB no endereço: <http://java.sun.com/products/ejb/> em 27/03/2010.
- ELLIS, D. **Worm anatomy and model**, Proceedings of the 2003 ACM workshop on rapid malcode, p. 42-50, 2003.
- ESTAN, C.; KEYS, K.; MOORE, D.; VARGHESE, G. **Building a better NetFlow**, ACM SIGCOMM Computer Communication Review, v. 34, n. 4, pp. 245-256, 2004.
- ESTEVEZ-TAPIADOR, J. M.; GARCIA-TEODORO, P.; DIAZ-VERDEJO, J. E. **Anomaly detection methods in wired networks: a survey and taxonomy**, Computer Communications 27, Elsevier, pp. 1569-1584, 2004.
- FARRAPOSO, S.; OWEZARSKI, P.; MONTEIRO, E. **A Multi-Scale Tomographic Algorithm for Detecting and Classifying Anomalies**, Proceedings of IEEE International Conference on Communications 2007, 363-370, 2007.
- GERSTING, J. L. **Mathematical Structures for Computer Science**, 5. ed., W H Freeman, 2002.
- GIARRATANO, J. C.; RILEY, G. D. **Expert Systems: Principles and Programming**, 4. ed., Course Technology, 2004.
- HAJJI, H. **Statistical Analysis of Network Traffic for Adaptive Faults Detection**, IEEE Transaction on Neural Networks, v. 16, n. 5, pp. 1503-1063, 2005.
- JAKOBSON, G.; WEISSMAN, M. D. **Alarm correlation**, IEEE Network, p. 52-59, Novembro, 1993.
- JIANG, J.; PAPAVASSILIOU, S. **Detecting Network Attacks in the Internet via Statistical Network Traffic Normally Prediction**, Journal of Network and Systems Management, v. 12, p. 51-72, mar. 2004.
- JUNG, J.; KRISHNAMURTHY, B.; RABINOVICH, M. **Flash Crowds and Denial of Service Attacks: Characterization and Implications for CDN's and Web Sites**, Proceedings of the 11<sup>th</sup> International Conference on World Wide Web, p.

293-304, 2002.

KIM, S. S.; REDDY, A. L. N. **Statistical techniques for detecting traffic anomalies through packet header data**, IEEE/ACM Transactions on Networking, V. 16, n. 3, 2008.

KIND, A.; STOECKLIN, M. P.; DIMITROPOULOS, X. **Histogram-Based Traffic Anomaly Detection**, IEEE Transactions on Network Service Management, V. 6, n. 2, 2009.

KLINE, J.; NAM, S.; BARFORD, P.; PLONKA, D.; RON, A. **Traffic Anomaly Detection at Fine Time Scales with Bayes Net**, The Third International Conference on Internet Monitoring and Protection, p. 37-46, 2008.

KRÜGEL, C.; TOTH, T.; KIRDA, E. **Service Specific Anomaly Detection for Network Intrusion Detection**, Proceedings of Symposium on Applied Computing 2002, SAC 2002. pp. 201-108, 2002.

KUANG, L.; ZULKERNINE, M. **An Anomaly Intrusion Detection Method using the CSI-KNN algorithm**, Proceedings of the 2008 ACM Symposium on Applied Computing, p. 921-926, 2008.

LAKHINA, A.; CROVELLA, M.; DIOT, C. **Characterization of Network-Wide Traffic Anomalies in Traffic Flows**, Proceedings of the 4th ACM SIGCOMM Internet Measurement Conference (IMC'04), pp. 201-206, 2004.

LI, J.; MANIKOPOULOS, C. **Early Statistical Anomaly Intrusion Detection of DOS Attacks Using MIB Traffic Parameters**, Proceedings of the 2003 IEEE Workshop on Information Assurance, United States Military Academy, p. 53-59, jun. 2003.

LI, M.; YU, S.; HE, L.; **Detecting Network-wide Traffic Anomalies based on Spatial HMM**, Proceedings of 2008 IFIP International Conference on Network Parallel Computing, p. 198-203, 2008.

LIM, S. Y.; JONES, A. **Network Anomaly Detection System: The State of Art of Network Behaviour Analysis**, Proceedings of International Conference on Convergence and Hybrid Information Technology 2008, p. 459-465, 2008.

MAHONEY, M. V.; CHAN, P. K. **Learning Nonstationary Models of Normal Network Traffic for Detecting Novel Attacks**, The Eighth ACM SIGKDD

- International Conference on Knowledge Discovery and Data Mining, pp. 376-385, 2002.
- MAURO, D. R.; SCHMIDT, K. J. **Essential SNMP**, O'Reilly, 2001.
- NET-SNMP** – Net-SNMP disponível via WEB no endereço <http://www.net-snmp.org> (24/07/2010).
- NGUYEN, H. X.; THIRAN, P. **Network Loss Inference with Second Order Statistics of End-to-End Flows**, Proceedings of the 7th ACM SIGCOMM conference on Internet measurement, p. 227-240, 2007.
- PANDA, D.; RAHMAN, R.; LANE, D. **EJB 3 in Action**, Manning, 2007.
- PATCHA, A.; PARK, J-M. **An overview of anomaly detection techniques: existing solutions and latest technological trends**, v. 51, n. 12, p. 3448-3470, 2007.
- PATHCHAR**, disponível via Web no endereço <http://www.caida.org/tools/utilities/others/pathchar/> (08/02/2010).
- PROENÇA JUNIOR, M. L. “**Baseline Aplicado a Gerência de Redes**”, tese de doutorado, Faculdade de Engenharia Elétrica e de Computação, Universidade Estadual de Campinas, Campinas, 2005.
- REALI, G.; MONACELLI, L. **Fault Localization in Data Networks**, IEEE Communications Letters, v. 13, n. 3, p. 161-163, 2009.
- RINGBERG, H.; SOULE, A.; REXFORD, J.; DIOT, C. **Sensitivity of PCA for traffic anomaly detection**, Proceedings of the 2007 ACM SIGMETRICS international conference on Measurement and modeling of computer systems, pp. 109-120, 2007.
- RFC 1157 – INTERNET ENGINEERING TASK FORCE (IETF). **A Simple Network Management Protocol (SNMP)**, RFC 1157, 1990.
- RFC 1213 – INTERNET ENGINEERING TASK FORCE (IETF). **Management Information Base for Network Management of TCP/IP-based internet: MIB-II**, RFC 1213, 1991.
- RFC 1757 – INTERNET ENGINEERING TASK FORCE (IETF). **Remote Network Monitoring Management Information Base**, RFC 1757, 1995.
- RFC 3954 – INTERNET ENGINEERING TASK FORCE (IETF). **Cisco Systems NetFlow Services Export Version 9**, RFC 3954, 2004.

- ROUGHAN, M.; GRIFFIN, T.; MAO, Z. M.; GREENBERG, A.; FREEMAN, B. **IP Forwarding Anomalies and Improving their Detection Using Multiple Data Sources**, Proceedings of the ACM SIGCOMM workshop on Network troubleshooting: research, theory and operations practice meet malfunctioning reality, p. 307-312, 2004.
- SAMAAN, N.; KARMOUCH, A. **Network Anomaly Diagnosis via Statistical Analysis and Evidential Reasoning**, IEEE Transactions on Network and Service Management, v. 5, n. 2, p. 65-77, 2008.
- SEKAR, R.; GUPTA, A.; FRULLO, J.; SHANBHAG, T.; TIWARI, A.; YANG, H.; ZHOU, S. **Specification-based Anomaly Detection: A New Approach for Detecting Network Intrusions**, Proceedings of the 9th ACM Conference on Computer and Communications Security (CCS 2002), p. 265-274, 2002.
- SIRIS, V. A.; PAPAGALOU, F. **Application of anomaly detection algorithms for detecting SYN flooding attacks**, Computer Communications 29, Elsevier, pp. 1433-1442, 2006.
- SOULE, A.; SALAMATIAN, K.; TAFT, N. **Combining Filtering and Statistical Methods for Anomaly Detection**, Proceedings of ACM SIGCOMM Internet Measurement Conference 2005 (IMC'05), p. 331-344, October 19-21, 2005, Berkeley, CA, USA.
- SHON, T.; MOON, J. **A hybrid machine learning approach to network anomaly detection**, Information Sciences, v. 177, no. 18, Sep. 2007, p. 3799-3821.
- STALLINGS, W., **SNMP, SNMPv2, SNMPv3, and RMON 1 and 2, 3**. Addison-Wesley, 1998.
- STEINDER, M.; SETHI, A. S. **Probabilistic Fault Localization in Communication Systems Using Belief Networks**, IEEE/ACM Transactions on Networking, v. 12, n. 5, 2004.
- STEINDER, M.; SETHI, A. S. **A survey of fault localization techniques in computer networks**, Science of Computer Programming, n. 53, 165-194, 2004.
- TANG, Y.; AL-SHAER, E.; BOUTABA, R. **Efficient Fault Diagnosis Using Incremental Alarm Correlation and Active Investigation for Internet and Overlay Networks**, IEEE Transactions on Network and Service Management,

v. 5, n. 1, p. 36-49, 2009.

THOTTAN, M.; JI, C. **Anomaly detection in IP networks**, Signal Processing, IEEE Transactions on Volume: 51, Issue: 8, Aug. 2003, Pages: 2191 – 2204.

VALDES, A.; SKINNER, K. **Probabilistic Alert Correlation**, Recent Advances in Intrusion Detection : 4th International Symposium, RAID 2001 Davis, CA, USA, October 10-12, 2001, Proceedings, p. 54-68, 2001.

WU, N.; ZHANG, J. **Factor analysis based anomaly detection** Information Assurance Workshop, 2003. IEEE Systems, Man and Cybernetics Society, June 2003, p. 108-115.

WU, Q.; SHAO, Z. **Network Anomaly Detection Using Time Series Analysis**, Proceedings of the Joint International Conference on Autonomic and Autonomous Systems and International Conference on Networking and Services (ICAS/ICNS 2005), 2005.

XIAO, L.; SHAO, Z.; LIU, G. **K-means Algorithm Based on Particle Swarm Optimization Algorithm for Anomaly Intrusion Detection**, Proceedings of the 6<sup>th</sup> World Congress on Intelligent Control and Automation, p. 5854-5858, 2006.

## Anexo A

### A.1 Anomalias do grupo A

Servidor Proxy				
Histerese	$\delta$	Taxa de detecção	Taxa de falsos positivos	Eficiência
60 segundos	1	0,94	0,31	1,63
	2	0,80	0,10	1,70
	3	0,36	0,07	1,29
	4	0,05	0,06	0,99

Servidor Proxy				
Histerese	$\delta$	Taxa de detecção	Taxa de falsos positivos	Eficiência
120 segundos	1	0,96	0,38	1,58
	2	0,89	0,20	1,69
	3	0,76	0,11	1,64
	4	0,56	0,05	1,51
	5	0,33	0,01	1,32
	6	0,16	0,00	1,16
	7	0,05	0,00	1,05
	8	0,00	0,00	1,00

Servidor Proxy				
Histerese	$\delta$	Taxa de detecção	Taxa de falsos positivos	Eficiência
180 segundos	1	0,96	0,41	1,55
	2	0,93	0,26	1,68
	3	0,83	0,16	1,67
	4	0,68	0,09	1,59
	5	0,53	0,06	1,47
	6	0,37	0,04	1,34
	7	0,24	0,03	1,21
	8	0,14	0,00	1,14
	9	0,07	0,00	1,07
	10	0,02	0,00	1,02
	11	0,00	0,00	1,00
	12	0,00	0,00	1,00

Servidor Proxy				
Histerese	$\delta$	Taxa de detecção	Taxa de falsos positivos	Eficiência
	1	0,96	0,43	1,53

240 segundos	2	0,93	0,28	1,66
	3	0,83	0,18	1,65
	4	0,71	0,10	1,60
	5	0,60	0,06	1,54
	6	0,45	0,03	1,42
	7	0,34	0,03	1,31
	8	0,23	0,01	1,22
	9	0,17	0,00	1,17
	10	0,09	0,02	1,07
	11	0,05	0,00	1,05
	12	0,03	0,00	1,03
	13	0,01	0,00	1,01
	14	0,00	0,00	1,00
	15	0,00	0,00	1,00
	16	0,00	0,00	1,00

Servidor Proxy				
Histerese	$\delta$	Taxa de detecção	Taxa de falsos positivos	Eficiência
300 segundos	1	0,96	0,44	1,52
	2	0,93	0,31	1,63
	3	0,87	0,21	1,65
	4	0,75	0,13	1,62
	5	0,63	0,09	1,54
	6	0,50	0,07	1,43
	7	0,37	0,06	1,31
	8	0,29	0,04	1,26
	9	0,24	0,01	1,23
	10	0,17	0,00	1,17
	11	0,11	0,00	1,11
	12	0,07	0,00	1,07
	13	0,05	0,00	1,05
	14	0,03	0,00	1,03
	15	0,02	0,00	1,02
	16	0,01	0,00	1,01
	17	0,00	0,00	1,00
	18	0,00	0,00	1,00
	19	0,00	0,00	1,00
	20	0,00	0,00	1,00

Servidor Web				
Histerese	$\delta$	Taxa de detecção	Taxa de falsos positivos	Eficiência

60 segundos	1	0,88	0,16	1,73
	2	0,52	0,05	1,47
	3	0,19	0,00	1,19
	4	0,01	0,00	1,01

Servidor Web				
Histerese	$\delta$	Taxa de detecção	Taxa de falsos positivos	Eficiência
120 segundos	1	0,92	0,28	1,65
	2	0,74	0,09	1,64
	3	0,49	0,02	1,47
	4	0,37	0,01	1,36
	5	0,22	0,00	1,22
	6	0,09	0,00	1,09
	7	0,01	0,00	1,01
	8	0,00	0,00	1,00

Servidor Web				
Histerese	$\delta$	Taxa de detecção	Taxa de falsos positivos	Eficiência
180 segundos	1	0,95	0,31	1,64
	2	0,81	0,13	1,68
	3	0,61	0,05	1,56
	4	0,46	0,01	1,45
	5	0,36	0,00	1,36
	6	0,27	0,00	1,27
	7	0,19	0,00	1,19
	8	0,11	0,03	1,08
	9	0,03	0,00	1,03
	10	0,00	0,00	1,00
	11	0,00	0,00	1,00
	12	0,00	0,00	1,00

Servidor Web				
Histerese	$\delta$	Taxa de detecção	Taxa de falsos positivos	Eficiência
240 segundos	1	0,96	0,34	1,62
	2	0,83	0,16	1,67
	3	0,64	0,07	1,56
	4	0,50	0,02	1,48
	5	0,40	0,00	1,39
	6	0,32	0,00	1,32
	7	0,27	0,00	1,27
	8	0,21	0,01	1,20

	9	0,13	0,00	1,13
	10	0,07	0,00	1,07
	11	0,04	0,00	1,04
	12	0,01	0,00	1,01
	13	0,00	0,00	1,00
	14	0,00	0,00	1,00
	15	0,00	0,00	1,00
	16	0,00	0,00	1,00

Servidor Web				
Histerese	$\delta$	Taxa de detecção	Taxa de falsos positivos	Eficiência
300 segundos	1	0,96	0,36	1,61
	2	0,86	0,19	1,67
	3	0,66	0,09	1,57
	4	0,51	0,04	1,47
	5	0,43	0,01	1,42
	6	0,35	0,00	1,34
	7	0,26	0,00	1,26
	8	0,19	0,00	1,19
	9	0,15	0,00	1,15
	10	0,12	0,00	1,12
	11	0,08	0,00	1,08
	12	0,06	0,00	1,06
	13	0,03	0,00	1,03
	14	0,01	0,00	1,01
	15	0,00	0,00	1,00
	16	0,00	0,00	1,00
	17	0,00	0,00	1,00
	18	0,00	0,00	1,00
	19	0,00	0,00	1,00
	20	0,00	0,00	1,00

Firewall				
Histerese	$\delta$	Taxa de detecção	Taxa de falsos positivos	Eficiência
60 segundos	1	0,87	0,30	1,56
	2	0,61	0,16	1,45
	3	0,30	0,10	1,20
	4	0,08	0,00	1,08

Firewall				
Histerese	$\delta$	Taxa de detecção	Taxa de falsos positivos	Eficiência

120 segundos	1	0,91	0,36	1,55
	2	0,77	0,23	1,54
	3	0,56	0,13	1,43
	4	0,39	0,08	1,31
	5	0,28	0,03	1,25
	6	0,15	0,00	1,15
	7	0,02	0,00	1,02
	8	0,00	0,00	1,00

Firewall				
Histerese	$\delta$	Taxa de detecção	Taxa de falsos positivos	Eficiência
180 segundos	1	0,92	0,38	1,54
	2	0,79	0,26	1,53
	3	0,64	0,16	1,48
	4	0,49	0,07	1,42
	5	0,41	0,03	1,38
	6	0,32	0,02	1,30
	7	0,23	0,00	1,23
	8	0,14	0,00	1,14
	9	0,08	0,00	1,08
	10	0,02	0,00	1,02
	11	0,00	0,00	1,00
	12	0,00	0,00	1,00

Firewall				
Histerese	$\delta$	Taxa de detecção	Taxa de falsos positivos	Eficiência
240 segundos	1	0,92	0,41	1,50
	2	0,82	0,29	1,53
	3	0,67	0,19	1,48
	4	0,53	0,12	1,41
	5	0,48	0,07	1,41
	6	0,40	0,04	1,41
	7	0,34	0,02	1,35
	8	0,26	0,02	1,32
	9	0,17	0,02	1,24
	10	0,12	0,00	1,12
	11	0,08	0,00	1,08
	12	0,02	0,00	1,02
	13	0,00	0,00	1,00
	14	0,00	0,00	1,00
	15	0,00	0,00	1,00

	16	0,00	0,00	1,00
--	----	------	------	------

Firewall				
Histerese	$\delta$	Taxa de detecção	Taxa de falsos positivos	Eficiência
300 segundos	1	0,92	0,45	1,48
	2	0,84	0,31	1,54
	3	0,67	0,19	1,48
	4	0,56	0,12	1,44
	5	0,47	0,08	1,39
	6	0,38	0,05	1,33
	7	0,33	0,04	1,29
	8	0,26	0,03	1,23
	9	0,20	0,02	1,19
	10	0,15	0,00	1,15
	11	0,08	0,00	1,08
	12	0,05	0,00	1,05
	13	0,04	0,00	1,04
	14	0,00	0,00	1,00
	15	0,00	0,00	1,00
	16	0,00	0,00	1,00
	17	0,00	0,00	1,00
	18	0,00	0,00	1,00
	19	0,00	0,00	1,00
	20	0,00	0,00	1,00

## A.2 Anomalias do grupo B

Servidor Proxy				
Histerese	$\delta$	Taxa de detecção	Taxa de falsos positivos	Eficiência
60 segundos	1	1,00	0,65	1,35
	2	0,98	0,47	1,51
	3	0,66	0,34	1,32
	4	0,09	0,28	0,81

Servidor Proxy				
Histerese	$\delta$	Taxa de detecção	Taxa de falsos positivos	Eficiência
120 segundos	1	1,00	0,71	1,29
	2	1,00	0,56	1,44
	3	0,98	0,43	1,55
	4	0,92	0,28	1,64

	5	0,69	0,19	1,50
	6	0,43	0,09	1,34
	7	0,10	0,00	1,10
	8	0,00	0,00	1,00

Servidor Proxy				
Histerese	$\delta$	Taxa de detecção	Taxa de falsos positivos	Eficiência
180 segundos	1	1,00	0,73	1,27
	2	1,00	0,61	1,39
	3	0,99	0,50	1,49
	4	0,98	0,37	1,61
	5	0,92	0,26	1,66
	6	0,80	0,17	1,63
	7	0,58	0,10	1,48
	8	0,37	0,04	1,33
	9	0,19	0,00	1,19
	10	0,06	0,00	1,06
	11	0,00	0,00	1,00
	12	0,00	0,00	1,00

Servidor Proxy				
Histerese	$\delta$	Taxa de detecção	Taxa de falsos positivos	Eficiência
240 segundos	1	1,00	0,74	1,26
	2	1,00	0,63	1,37
	3	0,99	0,52	1,48
	4	0,99	0,39	1,60
	5	0,96	0,29	1,67
	6	0,92	0,15	1,78
	7	0,75	0,13	1,63
	8	0,58	0,07	1,52
	9	0,44	0,04	1,39
	10	0,27	0,02	1,25
	11	0,13	0,00	1,13
	12	0,07	0,00	1,07
	13	0,02	0,00	1,02
	14	0,00	0,00	1,00
	15	0,00	0,00	1,00
	16	0,00	0,00	1,00

Servidor Proxy				
Histerese	$\delta$	Taxa de detecção	Taxa de falsos positivos	Eficiência

300 segundos	1	1,00	0,75	1,25
	2	1,00	0,65	1,35
	3	1,00	0,55	1,45
	4	0,98	0,45	1,53
	5	0,96	0,36	1,60
	6	0,89	0,26	1,63
	7	0,73	0,19	1,55
	8	0,64	0,14	1,50
	9	0,58	0,08	1,49
	10	0,42	0,03	1,39
	11	0,27	0,05	1,22
	12	0,18	0,03	1,15
	13	0,13	0,04	1,09
	14	0,08	0,00	1,08
	15	0,04	0,00	1,04
	16	0,02	0,00	1,02
	17	0,01	0,00	1,01
	18	0,00	0,00	1,00
	19	0,00	0,00	1,00
	20	0,00	0,00	1,00

Servidor Web				
Histerese	$\delta$	Taxa de detecção	Taxa de falsos positivos	Eficiência
60 segundos	1	1,00	0,61	1,39
	2	0,97	0,35	1,62
	3	0,55	0,24	1,31
	4	0,01	0,67	0,34

Servidor Web				
Histerese	$\delta$	Taxa de detecção	Taxa de falsos positivos	Eficiência
120 segundos	1	1,00	0,71	1,29
	2	0,99	0,48	1,51
	3	0,97	0,28	1,69
	4	0,92	0,18	1,73
	5	0,65	0,13	1,52
	6	0,30	0,22	1,08
	7	0,03	0,00	1,03
	8	0,00	0,00	1,00

Servidor Web
--------------

Histerese	$\delta$	Taxa de detecção	Taxa de falsos positivos	Eficiência
180 segundos	1	1,00	0,75	1,25
	2	0,99	0,55	1,44
	3	0,99	0,38	1,61
	4	0,95	0,25	1,71
	5	0,90	0,16	1,74
	6	0,78	0,10	1,68
	7	0,62	0,08	1,55
	8	0,43	0,11	1,32
	9	0,08	0,25	0,83
	10	0,02	0,00	1,02
	11	0,00	0,00	1,00
	12	0,00	0,00	1,00

Servidor Web				
Histerese	$\delta$	Taxa de detecção	Taxa de falsos positivos	Eficiência
240 segundos	1	1,00	0,77	1,23
	2	0,99	0,59	1,40
	3	0,98	0,42	1,56
	4	0,98	0,28	1,71
	5	0,96	0,18	1,78
	6	0,92	0,12	1,80
	7	0,87	0,06	1,81
	8	0,72	0,06	1,66
	9	0,48	0,08	1,40
	10	0,27	0,08	1,19
	11	0,16	0,12	1,04
	12	0,05	0,00	1,05
	13	0,02	0,00	1,02
	14	0,00	0,00	1,00
	15	0,00	0,00	1,00
	16	0,00	0,00	1,00

Servidor Web				
Histerese	$\delta$	Taxa de detecção	Taxa de falsos positivos	Eficiência
300 segundos	1	1,00	0,79	1,21
	2	0,99	0,62	1,37
	3	0,99	0,45	1,54
	4	0,97	0,30	1,67
	5	0,96	0,22	1,74
	6	0,94	0,16	1,78

	7	0,80	0,10	1,70
	8	0,63	0,06	1,57
	9	0,53	0,04	1,49
	10	0,43	0,04	1,39
	11	0,32	0,03	1,29
	12	0,22	0,00	1,22
	13	0,11	0,00	1,11
	14	0,05	0,00	1,05
	15	0,01	0,00	1,01
	16	0,00	0,00	1,00
	17	0,00	0,00	1,00
	18	0,00	0,00	1,00
	19	0,00	0,00	1,00
	20	0,00	0,00	1,00

Firewall				
Histerese	$\delta$	Taxa de detecção	Taxa de falsos positivos	Eficiência
60 segundos	1	1,00	0,68	1,32
	2	1,00	0,48	1,52
	3	0,73	0,30	1,42
	4	0,31	0,13	1,18

Firewall				
Histerese	$\delta$	Taxa de detecção	Taxa de falsos positivos	Eficiência
120 segundos	1	1,00	0,74	1,26
	2	1,00	0,62	1,38
	3	0,98	0,45	1,53
	4	0,92	0,28	1,64
	5	0,76	0,21	1,56
	6	0,53	0,11	1,42
	7	0,08	0,20	0,88
	8	0,02	0,00	1,02

Firewall				
Histerese	$\delta$	Taxa de detecção	Taxa de falsos positivos	Eficiência
180 segundos	1	1,00	0,75	1,25
	2	1,00	0,64	1,36
	3	1,00	0,50	1,50
	4	0,96	0,36	1,60
	5	0,94	0,27	1,67

	6	0,84	0,20	1,64
	7	0,75	0,09	1,66
	8	0,45	0,08	1,37
	9	0,31	0,00	1,31
	10	0,08	0,00	1,08
	11	0,00	0,00	1,00
	12	0,00	0,00	1,00

Firewall				
Histerese	$\delta$	Taxa de detecção	Taxa de falsos positivos	Eficiência
240 segundos	1	1,00	0,77	1,23
	2	1,00	0,68	1,32
	3	1,00	0,56	1,44
	4	0,98	0,43	1,55
	5	0,98	0,33	1,65
	6	0,94	0,26	1,68
	7	0,88	0,20	1,68
	8	0,71	0,17	1,53
	9	0,51	0,13	1,38
	10	0,39	0,08	1,31
	11	0,27	0,00	1,27
	12	0,08	0,00	1,08
	13	0,00	0,00	1,00
	14	0,00	0,00	1,00
	15	0,00	0,00	1,00
	16	0,00	0,00	1,00

Firewall				
Histerese	$\delta$	Taxa de detecção	Taxa de falsos positivos	Eficiência
300 segundos	1	1,00	0,79	1,21
	2	1,00	0,69	1,31
	3	1,00	0,56	1,44
	4	0,98	0,44	1,54
	5	0,98	0,34	1,64
	6	0,98	0,26	1,72
	7	0,86	0,25	1,61
	8	0,75	0,23	1,52
	9	0,61	0,23	1,38
	10	0,47	0,12	1,35
	11	0,27	0,08	1,19
	12	0,18	0,00	1,18

	13	0,14	0,00	1,14
	14	0,00	0,00	0,00
	15	0,00	0,00	0,00
	16	0,00	0,00	0,00
	17	0,00	0,00	0,00
	18	0,00	0,00	0,00
	19	0,00	0,00	0,00
	20	0,00	0,00	0,00