UNIVERSIDADE PAULISTA

PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE PRODUÇÃO

UM ANALISADOR DE TRÁFEGO DE REDES DE COMPUTADORES BASEADO EM LÓGICA PARACONSISTENTE ANOTADA EVIDENCIAL Eτ

Tese apresentada ao Programa de Pós-Graduação em Engenharia de Produção da Universidade Paulista – UNIP, para obtenção do título de Doutor em Engenharia de Produção.

AVELINO PALMA PIMENTA JUNIOR

São Paulo

2018

UNIVERSIDADE PAULISTA

PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE PRODUÇÃO

UM ANALISADOR DE TRÁFEGO DE REDES DE COMPUTADORES BASEADO EM LÓGICA PARACONSISTENTE ANOTADA EVIDENCIAL Eτ

Tese apresentada ao Programa de Pós-Graduação em Engenharia de Produção da Universidade Paulista – UNIP, para obtenção do título de Doutor em Engenharia de Produção.

Orientador: Prof. Dr. Jair Minoro Abe

Área de concentração: Gestão de Sistemas de Operação.

Linha de pesquisa: Métodos Quantitativos em Engenharia da Produção

AVELINO PALMA PIMENTA JUNIOR

São Paulo

2018

FICHA CATALOGRÁFICA

Pimenta Junior, Avelino Palma.
Um analisador de tráfego de redes de computadores baseado em Lógica Paraconsistente Anotada Evidencial Eτ / Avelino Palma Pimenta Junior. - 2018.
90 f. : il. color. + CD-ROM.

Tese de Doutorado Apresentada ao Programa de Pós Graduação em Engenharia de Produção da Universidade Paulista, São Paulo, 2018.

Área de Concentração: Gestão de Sistemas de Operação.
Orientador: Prof. Dr. Jair Minoro Abe.

1. Lógica Eτ. 2. Redes de computadores. 3. Reconhecimento de padrões. I. Abe, Jair Minoro (orientador). II. Título.

Ficha elaborada pelo Bibliotecário Rodney Eloy CRB8-6450

AVELINO PALMA PIMENTA JUNIOR

UM ANALISADOR DE TRÁFEGO DE REDES DE COMPUTADORES BASEADO EM LÓGICA PARACONSISTENTE ANOTADA EVIDENCIAL Eτ

Tese apresentada ao Programa de Pós-Graduação em Engenharia de Produção da Universidade Paulista – UNIP, para obtenção do título de Doutor em Engenharia de Produção.

Aprovado em:

BANCA EXAMINADORA

_____/__/___
Dr. Jair Minoro Abe
Universidade Paulista – UNIP

_____/__/___
Dr. José Benedito Sacomano
Universidade Paulista – UNIP

_____/__/___
Dr.ª Irenilza de Alencar Nääs
Universidade Paulista – UNIP

_____/__/___
Dr. Márcio José da Cunha
Universidade Federal de Uberlândia

_____/__/___
Dr. Josué Silva de Morais
Universidade Federal de Uberlândia

DEDICATÓRIA

Dedico aos meus pais, Avelino e Nympha, origem e fundamento da minha vida, refúgio e apoio em todos os momentos, e ao meu filho, Guilherme Perez Pimenta, que me ensinou o significado do amor incondicional.

AGRADECIMENTOS

Agradeço ao Prof. Dr. Jair Minoro Abe que, ao longo de todo este período, foi fonte inesgotável de conhecimento, paciência, sabedoria e amizade.

Agradeço à vice-reitora da UNIP, Prof.ª Melânia Dalla Torre, pela confiança e amizade ao longo de 20 anos de trabalho conjunto.

Agradeço à Prof.ª Dr.ª Irenilza de Alencar Nääs pelo conhecimento compartilhado, objetividade e excelência.

"Se você é pobre, trabalhe.
Se você é rico, trabalhe.
Se está sob o peso de responsabilidades aparentemente injustas, trabalhe.
Se você é feliz, continue a trabalhar, a preguiça dá lugar a dúvidas e receios.
Se a tristeza o esmaga e as pessoas amadas não parecem sinceras, trabalhe.
Se tiver decepções, trabalhe.
Se a fé titubeia e a razão falha, trabalhe.
Quando os sonhos estão desfeitos, e as esperanças parecem mortas, trabalhe.
Trabalhe como se a sua vida estivesse em perigo. Ela está mesmo.
Seja qual for o seu problema, trabalhe.
Trabalhe fielmente e trabalhe com fé.
O trabalho é o maior remédio material que existe. O trabalho cura (todos) tanto os padecidos mentais como físicos."

(William Wallace Rose)

RESUMO

As redes de computadores, cada vez mais, têm se tornado um meio importante de difusão de informações para as corporações, instituições de ensino e usuários comuns. A facilidade, rapidez e baixo custo no acesso a conteúdos têm feito com que cada vez mais usuários se conectem na rede mundial de computadores. Porém, essa demanda por recursos pode gerar também um aumento significativo de problemas na operação da infraestrutura da rede que, por suas características de distribuição e descentralização, acaba envolvendo todos os que nela se encontram conectados. Ainda que as redes de computadores possam utilizar diferentes equipamentos e arquiteturas, os elementos para análise do desempenho praticamente são os mesmos. Porém, a natureza estocástica de cada um deles dificulta uma análise determinística de eventuais problemas que possam ocorrer. Dessa forma, não é possível utilizar métricas pré-definidas para a análise de seu comportamento, sendo necessário efetuar uma parametrização dos valores de operação da rede de forma personalizada de acordo com seu desenho atual. Esses valores podem ser obtidos a partir da opinião de especialistas na área, como também podem ser aprendidos a partir da operação da própria rede. A partir da determinação dos valores considerados normais na operação de uma rede de computadores, eventuais desvios podem ser detectados, entre eles, a ocorrência dos chamados *malwares*, que podem facilmente ser disseminados nesse ambiente. Esta tese tem como objetivo estabelecer os atributos de funcionamento para cada rede particular de computadores e, a partir da opinião de especialistas e da operação da própria rede, detectar e discutir situações potencialmente anômalas em seu funcionamento e buscar os chamados *malwares* de forma sistêmica e não intrusiva com o uso da Lógica Eτ.

Palavras-chave: Lógica Eτ, redes de computadores, reconhecimento de padrões, tomada de decisão.

ABSTRACT

The computer networks, each and every time, have more and more become an important medium to spreading information for corporations, teaching institutions, and regular users. The user-friendly interface, speed, and low cost to access contents have taken more users to connect in the world wide web. However, this demand for resources may also generate a significant increase in operation problems in the web infrastructure, which, for its characteristics of distribution and decentralization, ends up involving everyone who is connected to it. Even though the computer networks may use different equipment and architectures, the elements to analyze performance are basically the same. However, the stochastic nature of each one makes it difficult to perform a deterministic analysis of eventual problems that may occur. This way, it is not possible to use pre-defined metrics for the analysis of its behavior, making it necessary to perform a parametrization of web operation values in a personalized way, according to its design in the moment. These values may be obtained from the opinion of specialists in the area, as well as apprehended from operating the network itself. From the determination of values considered normal in the operation of a computer network, occasional detours may be detected, among which, the occurrence of the so-called *malwares*, which may easily be spread in this environment. This thesis has the objective of stablishing the working attributes for each particular computer network and, from the opinion of specialists and the operation of the network itself, detect and discuss potentially anomalous in its working and search for the so-called malwares in a systemic and non-intrusive way, with the use of the Logic $E\tau$.

Key words: Logic $E\tau$; computer networks; standards recognitions; decision making

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

APMS	Advances in Production Management Systems
CERT.br	Centro de Estudos, Resposta e Tratamento de Incidentes de Segurança no Brasil
CGI	Comitê Gestor da internet no Brasil
CSIRT	Computer Security Incident Response Team
CSMA/CD	Carrier Sense Media Acces with Collision Detection
G_{ce}	Certainty Degree
G_{un}	Uncertainty Degree
Gbps	Gigabit por segundo
LAN	Local Area Network
Lógica Eτ	Lógica Paraconsistente Anotada Evidencial Eτ
Mbps	Megabit por segundo
MySQL CE	Banco de dados MySQL Community Edition
NIC	Núcleo de informação e coordenação do Ponto BR
QoS	Quality of Service
TI	Tecnologia da Informação
μ	Letra grega mi, que simboliza evidência favorável
λ	Letra grega lambda, que simboliza evidência contrária

SUMÁRIO

1 CONSIDERAÇÕES INICIAIS

Uma rede de computadores consiste em vários hosts conectados, que podem ser representados por um *desktop*, um *laptop*, um *smartphone*, dispositivos portáteis, sensores biomédicos, entre outros (XU; HE; LI, 2014)(ZHUMING BI; LI DA XU; CHENGEN WANG, 2014). Desde sua criação, a implantação de redes apontou para uma variedade de dispositivos de diferentes fabricantes e arquiteturas que, muitas vezes, possuem características significativamente distintas. Além disso, o tráfego de dados pode apresentar um padrão aleatório de comportamento, que, por sua vez, pode levar a interpretações imprecisas sobre o funcionamento dos usuários e da rede.

Dada essa limitação, os analisadores de redes de computadores normalmente procuram apresentar apenas uma visão instantânea do funcionamento das operações, sem efetuar uma análise qualitativa do que de fato ocorre.

Neste trabalho, será utilizada uma abordagem que permita ao gestor de infraestrutura computacional identificar padrões de operação de uma rede em particular, detectar problemas e eventualmente auxiliar na tomada de decisão adequada de acordo com o cenário de operação apresentado.

1.1 Introdução

O aumento do número de usuários, serviços e aplicativos da rede, juntamente com os muitos avanços na tecnologia da informação, tornam as redes e sistemas informáticos essenciais para a sobrevivência de todas as empresas, organizações e instituições educacionais (OBAIDAT; NICOPOLITIDIS; ZARAI, 2015).

Os problemas relacionados à segurança da informação na internet existem desde o seu surgimento. Porém, à medida que a tecnologia avança e os sistemas de gerenciamento da informação tornam-se cada vez mais poderosos, a questão de incremento da segurança da informação torna-se cada vez mais crítica (WHITE; REA, 2006).

É possível capturar os pacotes de dados que passam por esses dispositivos, e a extração de informações a partir deles pode prover ao administrador uma importante ferramenta de auxílio na tomada de decisões.

Deve-se, portanto, procurar uma forma de analisar os atributos de rede em busca de informações, por exemplo, em um sistema de análise dos *logs* de rede, de modo a extrair padrões significativos nas requisições e respostas.

Alguns elementos podem ser interessantes para a análise do tráfego de pacotes, dentre os quais: endereço lógico associado à requisição do recurso, horário da requisição, tempo de espera da resposta, tipo de resultado obtido, quantidade de dados da resposta na transação e destino da requisição (ROUSSKOV; SOLOVIEV, 1999).

O aumento crescente da rede mundial de computadores também leva a um aumento da complexidade de sua infraestrutura. Dessa forma, os métodos clássicos de análise do funcionamento da rede podem não ser os mais adequados para esse cenário (FERNANDEZ-PRIETO et al., 2012). Portanto, a Lógica Paraconsistente Anotada Evidencial Eτ (Lógica Eτ) pode se constituir como uma técnica viável para a busca por indícios de problemas, sejam eles causados tanto pela operação-padrão da rede, quanto por elementos intencionais (ABE, 1992) (ABE, 2015). Nesse último caso, pode ser constituído pela figura do usuário ou mesmo de um aplicativo dessa natureza (MISRA; VERMA; SHARMA, 2014).

A utilização da Lógica Eτ surge, portanto, como uma alternativa viável para a tomada de decisões em situações de incerteza, inconsistência e contradição nas mais diversas áreas como robótica, eletrônica, controle de tráfego, entre outras (DA SILVA FILHO, J.I., 2010) (ABE; AKAMA; NAKAMATSU, 2015).

1.1.1 Justificativa

O desenvolvimento desse projeto teve como principal motivação o grande número de incidentes que ocorrem anualmente nas redes de computadores, cujo funcionamento está diretamente ligado à internet. Como agravante, sua característica de entrega de dados baseada em datagramas, um serviço não orientado à conexão (COMER, 2000), em que os dados não seguem sempre o mesmo caminho entre origem e destino, dificulta de forma significativa o rastreamento do ponto em que uma ação maliciosa pode ter iniciado. De um modo geral, o número de incidentes aumenta a cada ano, de acordo com entidades que monitoram esse tipo de evento. Muitas vezes, essas situações podem levar a perdas

financeiras e, frequentemente, os custos operacionais decorrentes da perda de dados não podem ser estimados (LEE; YEH; WANG, 2013). Além disso, um serviço responsivo desempenha um papel crítico na determinação da satisfação do usuário final. Um cliente que experimenta um grande atraso ou perda depois de efetuar uma solicitação ao servidor da web de uma empresa geralmente muda para um concorrente que fornece um serviço mais rápido (AUSTIN et al., 2001).

Para se ter uma ideia mais precisa da relevância dos incidentes de segurança no país, é necessário conhecer a sua dimensão. No Brasil, O CERT.br é o Grupo de Resposta a Incidentes de Segurança para a internet Brasileira, mantido pelo NIC.br do CGI. É responsável por tratar incidentes de segurança em computadores que envolvam redes conectadas à internet brasileira (CERT, 2017a). Atua como um ponto central para notificações de incidentes de segurança no Brasil, provendo a coordenação e o apoio no processo de resposta a incidentes e, quando necessário, colocando as partes envolvidas em contato.

Além do processo de tratamento a incidentes, o CERT.br também atua através do trabalho de conscientização sobre os problemas de segurança, da análise de tendências e correlação entre eventos na internet brasileira e do auxílio ao estabelecimento de novos CSIRTs no Brasil.

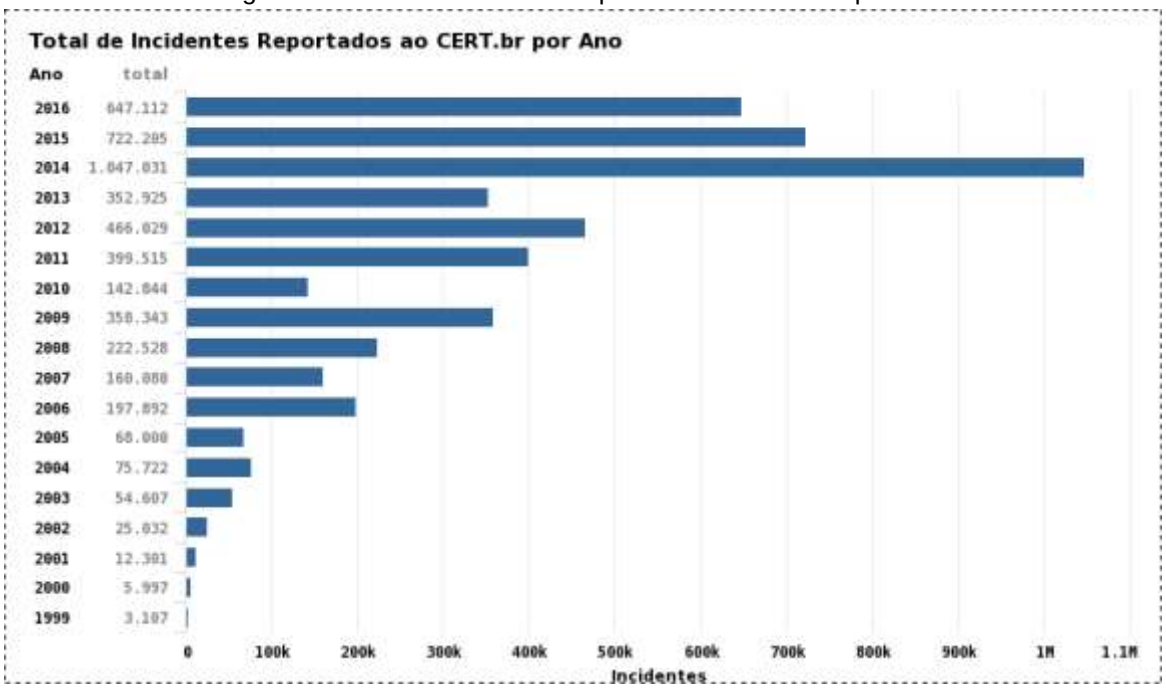
Essas atividades têm como objetivo estratégico aumentar os níveis de segurança e de capacidade de tratamento de incidentes das redes conectadas à internet no Brasil. As atividades conduzidas pelo CERT.br fazem parte das atribuições do CGI.br de (CERT, 2017a):

a) Estabelecer diretrizes estratégicas relacionadas ao uso e desenvolvimento da internet no Brasil;
b) Promover estudos e recomendar procedimentos, normas e padrões técnicos e operacionais para a segurança das redes e serviços de internet, bem como para a sua crescente e adequada utilização pela sociedade;
c) Ser representado nos fóruns técnicos nacionais e internacionais relativos à internet.

Dessa forma, o CERT pode ser considerado como a entidade referência responsável por compilar todos os eventos relacionados à segurança da internet no

Brasil. Desde o ano de 1999, as principais estatísticas relativas a problemas dessa natureza têm sido estudadas, tais como são apresentadas na Figura 1:

Figura 1 – Total de incidentes reportados ao CERT.br por ano



Fonte: Cert (2017b).

Como é possível verificar, desde 1999 até 2016, o total anual de incidentes passou de aproximadamente 3000 para quase 650.000 ocorrências, tendo um pico de mais de 1 milhão de ocorrências em 2014. Isso mostra claramente o quão fundamental é a gerência e o monitoramento das informações que trafegam em uma rede de computadores.

1.2 Objetivos

1.2.1 Objetivo Geral

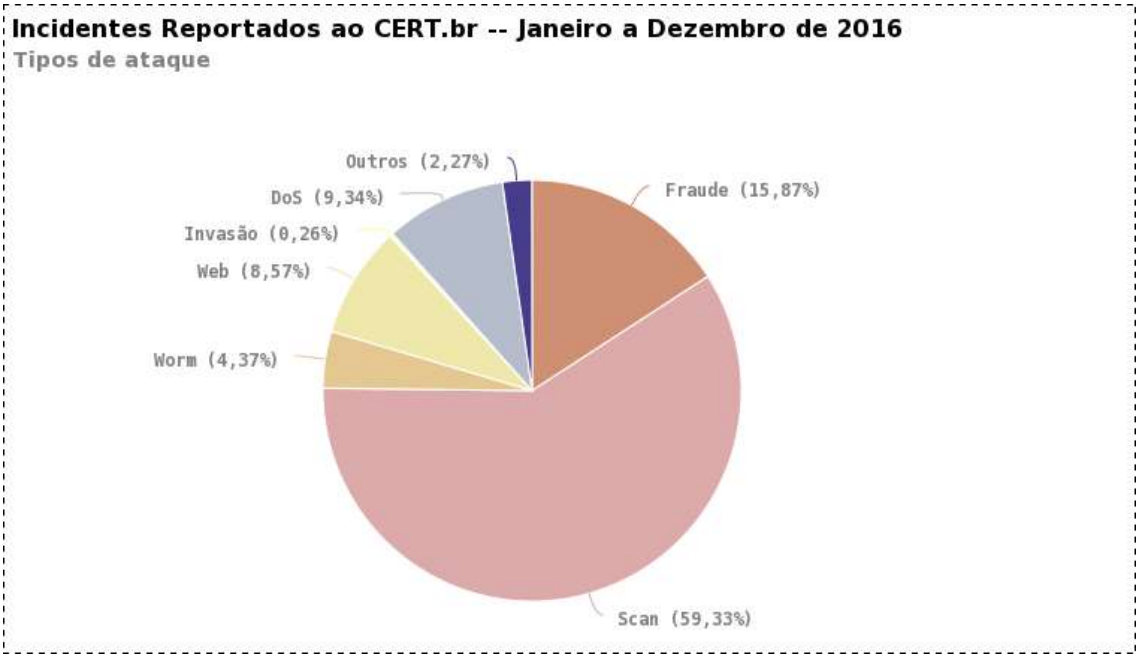
Ao se considerar a criação de um analisador de tráfego de rede com base em atributos obtidos a partir de *logs* de acesso da rede, tem-se como objetivo determinar como os dispositivos de uma rede se comportam, ou seja, quais tipos de atividades eles desempenham, com que frequência e, eventualmente, quais problemas podem estar ocorrendo. Essa informação pode ser utilizada de diversas maneiras. Além da questão da detecção de problemas de operação, pode ser

possível também mapear as preferências das atividades desenvolvidas nas redes ou ainda efetuar o balanceamento de carga nos recursos computacionais quando necessários, de modo a prover a disponibilidade máxima necessária nas operações de rede.

É importante ressaltar que eventuais problemas podem ser detectados ainda antes de serem percebidos pelo usuário. Muitas vezes, a degradação do desempenho de uma rede de computadores só pode ser notada quando pouco há a ser feito, frequentemente gerando perdas de diversas naturezas, tais como financeiras ou de informação. Essas perdas podem afetar as decisões estratégicas de uma empresa, dependendo da sua dimensão e do atraso com que são percebidas.

Dentre os diversos tipos de incidentes que ocorrem em redes de computadores, alguns podem ser causados por ações maliciosas. O analisador de tráfego de rede criado para este projeto se mostrou capaz de também detectar tais ações. De acordo com o CERT, foram estes os principais causadores de problemas nas redes, conforme apresentado pela Figura 2:

Figura 2 – Tipos de ataque reportados ao CERT em 2016



Fonte: Cert (2017c).

A partir do desenvolvimento deste projeto, foi possível utilizar um instrumento não invasivo e sistêmico, ou seja, capaz de abranger a rede como um todo, com rápida detecção e baixo custo de implantação para a maior parte das empresas no

país. Tal instrumento foi elaborado a partir de uma plataforma baseada em código fonte aberto, tanto na parte de desenvolvimento quanto na aquisição dos dados para análise.

1.2.2 Objetivos Específicos

Os objetivos específicos deste trabalho tiveram como ponto de partida a observação dos problemas existentes na maioria das empresas que utilizam as redes de computadorese foram divididos em três partes:

a) **Objetivo Específico 1:** determinação dos atributos que devem ser considerados ao avaliar o desempenho da rede. Esta etapa compreende verificar, junto aos especialistas na área de infraestrutura, quais elementos devem ser utilizados para a análise de desempenho de uma rede de computadores. Existem inúmeros parâmetros que podem ser quantificados no momento em que os dados fluem por uma rede de computadores. Tais parâmetros são armazenados nos chamados *logs* de operação, que podem ser obtidos a partir de servidores ou roteadores. Essa etapa gerou um artigo, que foi publicado na APMS do ano de 2015 (PIMENTA JR; ABE, 2015).
b) **Objetivo Específico 2:** verificação da operação da rede em função de dados de utilização. A partir de diversas medidas de desempenho detectadas ao longo de intervalos de tempo na rede de computadores, considerando os atributos significativos determinados no primeiro trabalho, busca-se determinar quais as possíveis interpretações que podem ser obtidas e que poderão auxiliar o gestor na tomada de decisões. De forma resumida, nessa etapa, é determinado o que é considerado normal ou não na operação de uma rede de computadores a partir da parametrização dos seus atributos de funcionamento. É fundamental frisar que não existem medidas absolutas do que é ou não um bom desempenho de uma rede de computadores, tendo em vista que as medidas são altamente dependentes do projeto que se elabora a partir dos requisitos. Essa etapa gerou um artigo que foi publicado na APMS do ano de 2016 (PIMENTA JR; ABE, 2016).

c) **Objetivo Específico 3:** busca por anomalias no funcionamento de dispositivos de rede. Diversos elementos podem gerar situações de anomalia ou inconsistência dentro da rede de computadores. Um deles é a entrada e disseminação dos denominados *malwares*. A partir da busca por situações de inconsistência, é possível localizar os dispositivos que potencialmente podem estar comprometidos. No modelo atual utilizado pela maioria das grandes empresas, essa busca normalmente envolve o rastreamento pontual e *in loco,* o que pode gerar problemas de retardo, alto custo financeiro e violação da privacidade. O atual desenvolvimento desse trabalho teve como um dos objetivos superar essas três características não desejáveis.

Os três objetivos específicos descritos foram motivadores da produção de três artigos científicos, além da criação de um analisador do tráfego de dados para esse fim. Este se mostrou capaz de extrair padrões de funcionamento da rede à luz da Lógica $E\tau$.

Foi possível detectar erros e problemas de diversas naturezas, tanto específicos quanto sistêmicos, a partir de eventuais desvios do que se considera a faixa de normalidade. Erros específicos podem ser considerados como aqueles que ocorrem em pontos isolados da rede como, por exemplo, um determinado equipamento com problemas físicos ou lógicos. Já o erro sistêmico ocorre quando o funcionamento da rede é afetado como um todo, devido a uma operação acima dos seus limites, que pode ser traduzida como problemas de congestionamento (quando o meio não é capaz de suportar um tráfego de dados intenso) ou fluxo (quando um ou mais receptores não são capazes de processar um volume de dados suficientemente grande). De maneira simplificada, a diferença do primeiro para o segundo é o número de equipamentos envolvidos.

2 REFERENCIAL TEÓRICO

2.1 Elementos de redes de computadores

Redes de computadores foram concebidas inicialmente como um meio para compartilhar dispositivos e serviços e se tornaram parte do cotidiano de muitas empresas como um instrumento capaz de oferecer um conjunto de recursos imprescindíveis para os usuários, provendo serviços e informações estratégicas. Atualmente, são utilizadas em todos os tipos de negócios, mais notadamente em grandes empresas, mas também em pequenas e médias, tendo em vista que os dispositivos de redes possuem um espectro muito amplo de preços, o que acaba facilitando a aquisição. O investimento em redes de computadores, nesse caso, não é mais considerado um gasto, ainda que, muitas vezes, os custos envolvidos na implantação e manutenção da infraestrutura possam alcançar valores significativamente altos, dependendo da natureza do projeto implementado.

Dessa forma, a falta de monitoramento de recursos e serviços de rede pode ocasionar uma grande variedade de consequências, desde aquelas consideradas simples até as mais graves, gerando, em alguns casos, danos irreparáveis. Para evitar esse tipo de problema e manter o ambiente de rede operacional, é necessário que medidas preventivas e eventualmente corretivas sejam tomadas; porém, antes, é fundamental que a origem do problema seja detectada.

Uma rede de computadores consiste de vários dispositivos conectados, que podem ser representados por um *desktop*, um *laptop*, um *smartphone*, entre outros. Nesse ambiente altamente heterogêneo, a necessidade de serviços eficientes de entrega de conteúdo está se tornando um requisito importante para a nova infraestrutura de serviços de internet (CANALI; CARDELLINI; LANCELLOTTI, 2006)**.** Muitos desses equipamentos podem ter diferentes arquiteturas e também utilizar diferentes sistemas operacionais e aplicativos. Redes de computadores são fundamentalmente heterogêneas e sua distribuição responde mais às demandas correntes do que ao projeto original. Dentro da mesma rede, diferentes enlaces podem operar em velocidades distintas e serem executados em diferentes mídias, tais como 1 Gbps ou 100 Mbps, cobre ou fibra ótica (KUROSE; ROSS, 2013). A internet, responsável por aglutinar essas redes inicialmente isoladas, representa um importante meio de troca de dados e serviços. De acordo com Santos, Moreira e

Rocha (2010), trata-se de um projeto que surgiu nos Estados Unidos da América em 1966, com o objetivo de interligar centros militares e de pesquisa no país, tendo recebido inicialmente a designação de ARPANET, cuja primeira fase é representada na Figura 3.

Figura 3 – Distribuição física da ARPANET em 1969

Fonte: Chiappa (2014).

Um dos problemas gerados por essa diversidade, constituída por dispositivos físicos, arquiteturas e sistemas operacionais, relaciona-se com as dificuldades em medir o desempenho de uma rede. Métodos de avaliação típicos, com referências pré-estabelecidas de desempenho, no entanto, são limitados na sua aplicabilidade. Muitas vezes, eles não são representativos das características de tráfego de qualquer instalação (DAVISON; WU, 2004).

Como mencionado por Masuda, Ishida e Nishi (2013), houve um drástico aumento na quantidade de informações que transita pela internet por conta do grande aumento no número de usuários. "Dessa forma, com o aumento crescente de usuários da *World Wide Web*, a necessidade de um desempenho satisfatório torna-se cada vez mais relevante" (BENADITP; FRANCISF, 2015). Portanto, o monitoramento das informações acaba por se tornar um fator fundamental nos departamentos de tecnologia (LIN; HUANG, 2013).

Alguns elementos importantes devem ser considerados na gerência de tráfego de dados em redes de computadores, tais como: confiabilidade, integridade e disponibilidade (KUROSE; ROSS, 2013; ROSEN, 2008).

A confiabilidade pode ser entendida como a capacidade da rede de computadores em transmitir dados, com sucesso, de uma determinada origem até um destino (LIN; YEH, 2011). A manutenção dessa característica é um desafio constante, tendo em vista que falhas são inevitáveis nas redes de computadores. Dessa forma, sua detecção e isolamento imediatos são necessários (GARSHASBI, 2016).

A integridade é a capacidade de manter intacta a informação, de modo a garantir a sua qualidade (KUROSE; ROSS, 2013). A perda da integridade da informação acaba por criar inconsistências no sistema e prejudicar, de forma significativa, o seu aspecto funcional. Um ambiente de rede que opera além de sua capacidade operacional - ou seja, que gera altas taxas de erros de comunicação - pode levar a situações potencialmente perigosas de perda da integridade dos dados, o que pode impactar de maneira significativa nos sistemas que deles dependem.

A disponibilidade é a capacidade de prover acesso aos sistemas de informação tão logo estes sejam solicitados (KUROSE; ROSS, 2013). Um sistema com baixa disponibilidade acaba por gerar insatisfação e baixa produtividade do usuário. Portanto, preservar a estabilidade da rede é importante para garantir que os serviços não sejam interrompidos (YEH; FIONDELLA, 2016).

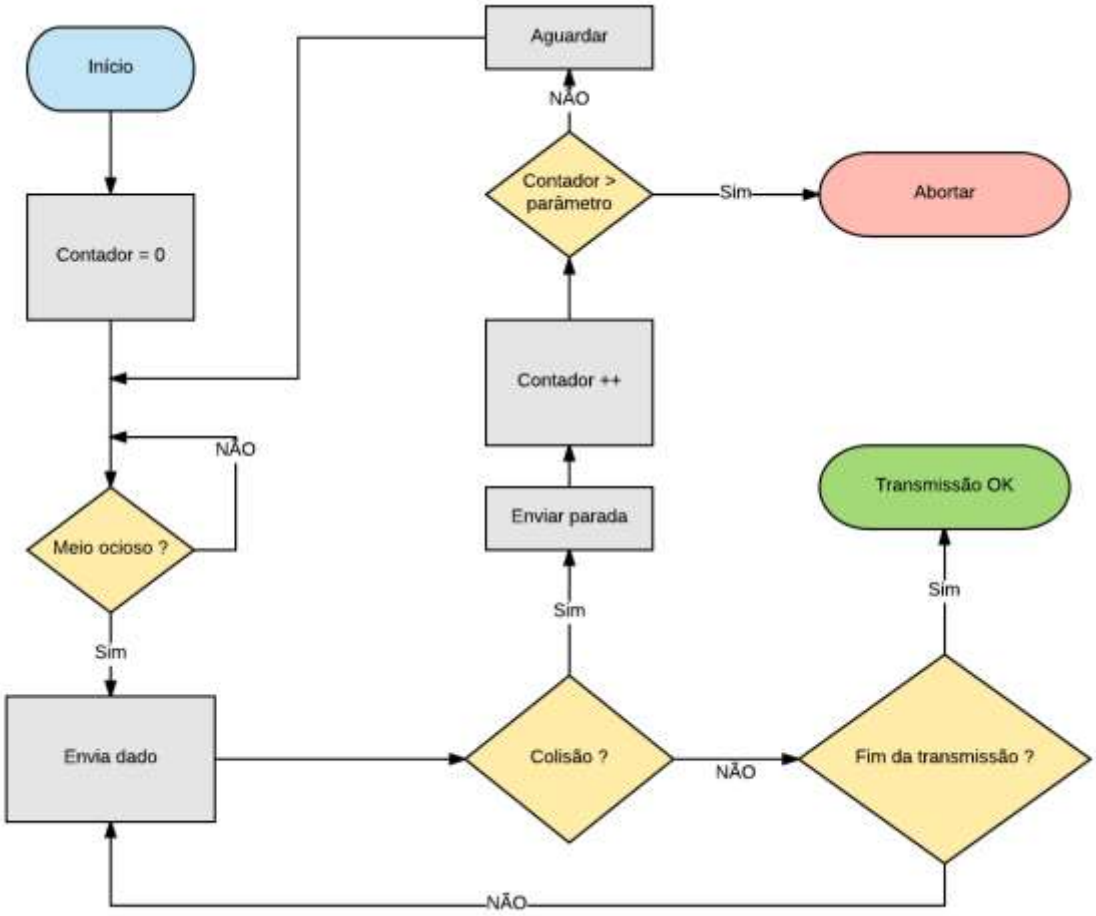
Nas redes de computadores, erros são muitas vezes causados por anomalias, que podem ser entendidas como comportamentos inesperados em uma rede (FIDALGO; LOPES, 2005; BRIGHENTI; SANZ-BOBI, 2011). Essas falhas são tratadas por métodos de acesso como o conhecido por CSMA – acesso múltiplo com detecção de portadora – que é um protocolo de controle de acesso ao meio (TANENBAUM, 2003). Sob a variante com detecção de colisão (CSMA/CD), esse protocolo é amplamente utilizado em ambientes de rede local (LAN), como o 802.3 Ethernet, que é o modelo de LAN mais utilizado no mundo (GÓMEZ-CORRAL, 2010).

De forma resumida, quando um dispositivo deseja enviar dados em uma rede, inicialmente é verificada a existência de tráfego no canal de comunicação e, caso, naquele instante, o meio estiver livre, envia-se um pacote de dados na rede (caso contrário, deve-se aguardar para efetuar nova tentativa). Porém, se no momento do

envio eventualmente ocorrer a perda do dado, é gerado um número aleatório de espera e uma retransmissão é efetuada, seguindo a lógica inicial (GORRY FAIRHURST, 2004). Porém, existe um limite finito e parametrizável de retransmissões, tendo em vista que um número indefinido poderia agravar ainda mais a situação da rede potencialmente com problemas. Dessa forma, com o uso desse algoritmo, espera-se evitar que um dispositivo que não consiga enviar seus dados em uma rede efetue retransmissões indefinidamente, gerando taxas de erros cada vez maiores.

Ainda que nos dias atuais a perda de dados em função de colisão no meio de comunicação tenha diminuído significativamente em função da substituição dos concentradores denominados *Hubs*, que deram lugar aos atuais *Switches*, a utilização do método CSMA/CD permanece devido à questão de compatibilidade com sistemas legados. Seu funcionamento pode ser representado conforme a Figura 4:

Figura 4 – Representação do método de acesso CSMA/CD



Fonte: Tanenbaum (2003).

O que é observado na prática em redes de computadores é que muitas anomalias relatadas acabam sendo falsas, refletindo um comportamento incomum, porém benigno (GRANA et al., 2016). Isso faz com que a incerteza se torne um elemento ainda mais relevante nesse tipo de avaliação. Considerando sua própria natureza, o funcionamento das redes de computadores é significativamente afetado por eventos estocásticos, o que acaba, por sua vez, refletindo na análise de performance efetuada. O argumento para essa afirmação baseia-se no princípio de que as ações dos usuários se apresentam como elementos aleatórios (BEN-PORAT; BREMLER-BARR; LEVY, 2014). De fato, a variabilidade de serviços disponíveis é considerável, portanto, os tipos de comportamento dos usuários acabam por acompanhar essa tendência.

Além disso, o estabelecimento de um conjunto de critérios deve ser feito de forma a evitar falsos positivos (FOSSACECA; MAZZUCHI; SARKANI, 2015), cujos desdobramentos poderão gerar diversos problemas, inclusive os de ordem legal.

2.2 Conceitos e tipos de malwares

Segundo o CGI.BR (2012), *malwares* são programas especificamente desenvolvidos para executar ações danosas e atividades maliciosas em um computador. Podem se espalhar facilmente em uma rede de computadores devido à interconectividade das estações de trabalho (J.B.SHUKLA et al., 2014). Algumas das diversas formas como os códigos maliciosos podem infectar ou comprometer um computador são:

a) Pela exploração de vulnerabilidades existentes nos programas instalados;
b) Pela execução automática de mídias removíveis infectadas, como pen-drives;
c) Pelo acesso a páginas maliciosas da web, utilizando navegadores vulneráveis;
d) Pela ação direta de atacantes que, após invadirem o computador, incluam arquivos contendo códigos maliciosos;
e) Pela execução de arquivos previamente infectados, obtidos em anexos de mensagens eletrônicas, via mídias removíveis, em páginas da web ou diretamente de outros computadores (através do compartilhamento de recursos).

De acordo com Zuben (2012), alguns dos principais tipos de *malware* que podem ser encontrados nos sistemas computacionais, e suas respectivas características, são:

a) **Vírus:** programa ou parte de um programa de computador, normalmente malicioso, que se propaga inserindo cópias de si mesmo e se tornando parte de outros programas e arquivos. Depende da execução do programa ou arquivo infectado para se tornar ativo e continuar o processo de infecção;

b) ***Worm*:** programa capaz de se propagar automaticamente pelas redes, enviando cópias de si mesmo de computador para computador. Utilizam, como meio de propagação, a execução direta de suas cópias e a exploração automática de vulnerabilidades existentes em programas instalados em computadores. Consomem grandes quantidades de recursos e afetam a utilização de computadores e redes;

c) ***Bot*:** programa que dispõe de mecanismos de comunicação com o invasor, os quais permitem que ele seja controlado remotamente. O processo de infecção e propagação é similar ao do *worm.* Faz uso dos chamados “computadores zumbis”, controlados remotamente sem o conhecimento dos proprietários. Podem executar ações maliciosas, tais como: ataques na internet, furto de dados e envio de spam;

d) ***Spyware*:** programa projetado para monitorar as atividades de um sistema e enviar as informações coletadas para terceiros. Pode ser usado de forma legítima ou maliciosa, dependendo de como é instalado, das ações realizadas, do tipo de informação monitorada e do uso que é feito por quem recebe a informação;

e) ***Backdoor*:** programa que permite o retorno de um invasor a um computador comprometido por meio da inclusão de serviços criados ou modificados para esse fim. Pode ser incluído pela ação de outros códigos maliciosos ou por atacantes;

f) ***Trojan*:** programa que, além de executar as funções para as quais foi aparentemente projetado, também executa outras funções, normalmente maliciosas, sem o conhecimento do usuário. Pode ser instalado pela ação do usuário via arquivos recebidos ou por atacantes ou via alteração de programas já existentes.

Atualmente, a forma mais utilizada de proteção em grandes empresas é a utilização dos chamados antivírus coorporativos, cujo uso baseia-se no princípio de instalação de um módulo em cada cliente da rede, o qual permite a atualização periódica da base de dados de vírus existentes por um programa gestor normalmente instalado no servidor. Este mesmo programa recebe o *feedback* dos módulos clientes, em caso de infecção do equipamento, permitindo a geração de relatórios ao administrador.

Existem diversos problemas nesse tipo de abordagem. O primeiro diz respeito aos custos de aquisição, tendo em vista que os denominados antivírus "*Corporate Edition*" têm seus valores calculados em função do número de clientes conectados à rede. Considerando a fácil disseminação de *malwares* em redes de computadores pela sua característica própria de difusão de informações, não há alternativa a não ser adquirir licenças para todos os equipamentos da rede.

O segundo problema diz respeito aos custos de manutenção. Os antivírus coorporativos disponíveis no mercado têm prazo determinado de atualização de suas bases de dados e *engines* de programas. Dessa forma, há a necessidade de gastos para a utilização contínua do aplicativo ao longo do tempo.

O terceiro problema diz respeito à defasagem tecnológica. Um *malware* sempre estará adiante dos sistemas capazes de detectá-lo, considerando-se que, para que seja criado um antídoto, é necessário primeiro isolar a contaminação durante sua fase aguda de infecção da rede. Não há como determinar quanto tempo decorre entre a infecção inicial e a completa imunização dos grandes sistemas, tendo em vista que esse período depende de diversos fatores, dentre os quais o potencial de letalidade. Em geral, *malwares* com baixo potencial de comprometimento demoram mais tempo para serem notados, pois o principal objetivo de sua criação pode ser apenas capturar dados dos usuários, dificultando que sua presença seja percebida.

2.3 A Lógica Eτ

Altos níveis de incerteza e imprevisibilidade são componentes importantes no monitoramento de redes de computadores. O argumento para essa assertiva é baseado no princípio de que as ações dos usuários são apresentadas como elementos aleatórios (BEN-PORAT; BREMLER-BARR; LEVY, 2014). Portanto, o uso

de uma lógica não clássica se torna uma opção. A Lógica Eτ pode ser uma técnica viável para buscar indícios de problemas durante o funcionamento normal da rede ou por elementos intencionais (PIMENTA; ABE; DE OLIVEIRA, 2015;PIMENTA JR; ABE, 2016). Nesse último caso, pode ser causado por mau uso ou software malicioso (MISRA; VERMA; SHARMA, 2014).

Segundo Abe, Akama e Nakamatsu (2015), as fórmulas atômicas da Lógica Eτ são do tipo p (μ, λ), onde (μ, λ) ∈ $[0, 1]^2$ ([0, 1] é o intervalo unitário real) e p denota uma variável proposicional. Portanto, entre várias leituras, p (μ, λ) pode ser lido intuitivamente: supõe-se que a evidência favorável de p é μ e a evidência contrária de p é λ. Assim, temos, por exemplo, as seguintes leituras particulares:

a) $p_{(1.0, 0.0)}$ pode ser lido como uma proposição verdadeira;
b) $p_{(0.0, 1.0)}$ como falso;
c) $p_{(1.0, 1.0)}$ como inconsistente;
d) $p_{(0.0, 0.0)}$ como paracompleto e
e) $p_{(0,5, 0,5)}$ como uma proposição indefinida.

Os graus de incerteza e certeza associados a (μ, λ) são definidos (ABE, 2015; AKAMA, 2016):

a) Grau de incerteza: $G_{un}(\mu, \lambda) = \mu + \lambda - 1$ $(0 \leq \mu, \lambda \leq 1)$ e;

b) Grau de certeza: $G_{ce}(\mu, \lambda) = \mu - \lambda$ $(0 \leq \mu, \lambda \leq 1)$.

Uma relação de ordem é definida em $[0, 1]^2$: $(\mu_1, \lambda_1) \leq (\mu_2\ \lambda_2) \Leftrightarrow \mu_1 \leq \mu_2$ e $\lambda_2 \leq \lambda_{1,}$ formando um Reticulado, simbolizado por τ. A Tabela 1 representa todos os estados e transições possíveis pela Lógica Eτ

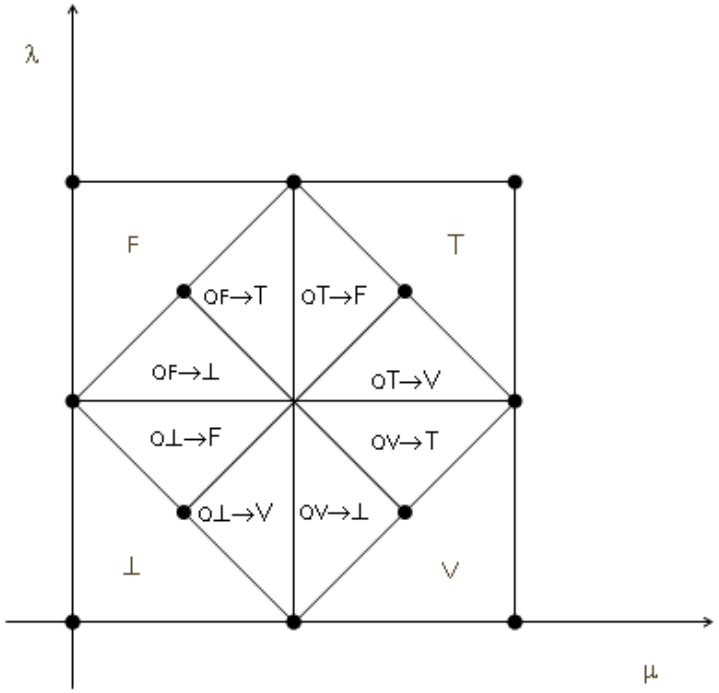
Tabela 1 – Estados extremos e não extremos

Estados extremos	Símbolo	Estados nãoextremos	Símbolo
Verdadeiro	V	Quasi verdadeiro tendendo a Inconsistente	QV→ T
Falso	F	Quasi verdadeiro tendendo a Paracompleto	QV→ ⊥
Inconsistente	T	Quasi falso tendendo a Inconsistente	QF→ T
Paracompleto	⊥	Quasi falso tendendo a Paracompleto	QF→ ⊥
		Quasi inconsistente tendendo a Verdadeiro	QT→ V
		Quasi inconsistente tendendo a Falso	QT→ F
		Quasi paracompleto tendendo a Verdadeiro	Q⊥→ V
		Quasi paracompleto tendendo a Falso	Q⊥→ F

Fonte: Abe (2015).

Com o grau de certeza e incerteza, é possível determinar os seguintes 12 estados de saída, mostrados na Figura 5:

Figura 5 – Estados extremos e não extremos do Reticulado τ



Fonte: Abe (2015).

3 DESENVOLVIMENTO

O desenvolvimento do projeto do analisador de redes de computadores foi composto por diversos elementos, tais como levantamento dos atributos de interesse, estudo de como os mesmos poderiam ser quantificados e determinação dos estados da rede a partir da Lógica $E\tau$. Com base nos resultados obtidos, a observação *in loco* do funcionamento da rede e a confrontação com os resultados teóricos obtidos foram realizadas.

A fonte de informação de onde poderiam ser obtidos os dados para análise também foi importante, tendo em vista que as redes de computadores podem gerar quantidades muito grandes de informação de tráfego, as quais podem demandar um custo de processamento significativo. Os passos detalhados de como foi realizado esse trabalho são descritos nos itens a seguir.

3.1 Procedimentos metodológicos

Ao se considerar uma hierarquia de complexidade nos protocolos de comunicação de uma rede de computadores, uma abordagem possível passa pela seleção de níveis mais baixos de abstração das operações de rede. Uma comparação possível, nesse caso, poderia ser feita com a comunicação humana. Em vez de uma análise do conteúdo que é dito pelo emissor, o foco se volta para a forma como ele fala. De fato, ao se observar a maneira como um indivíduo se comunica, é possível extrair diversas informações a respeito de seu estado, tais como euforia, raiva, indiferença, etc.

O mesmo pode ser aplicado a uma rede de computadores. A partir de atributos como tempo de resposta, quantidade de dados trafegados, número de transações e taxa de erros, pode ser possível extrair padrões de acesso dos usuários e, consequentemente, obter informações importantes sobre o funcionamento da rede. Eventuais tomadas de decisão sobre como atuar na rede tornar-se-ão mais fáceis e viáveis.

Os artigos desenvolvidos para esse trabalho possuem o mesmo escopo, porém utilizam abordagens distintas. No primeiro, submetido e aprovado pela APMS do ano de 2015, foram utilizadas as avaliações de especialistas na área (analistas

de infraestrutura e gerentes de rede) com o objetivo de avaliar as crenças favoráveis e contrárias a cada um dos atributos analisados. Já no segundo artigo, submetido e aprovado pela APMS em 2016, as crenças foram estabelecidas a partir de cálculos matemáticos, considerando os atributos e as especificidades da rede analisada. Algumas situações reais do funcionamento da rede foram discutidas. No terceiro artigo, cuja elaboração está sendo concluída para submissão, os resultados dos dois primeiros artigos são utilizados na busca não intrusiva e sistêmica dos chamados *malwares* em uma rede de computadores.

Em todos os artigos produzidos, alguns procedimentos são comuns e foram executados conforme a sequência a seguir:

a) A partir de uma rede de computadores em operação, foram obtidos os atributos de todos os equipamentos nela conectados, durante cinco dias úteis da semana;
b) Os atributos escolhidos foram tempo de resposta, quantidade de dados trafegados, número de transações e total de erros. Esses atributos foram determinados no primeiro artigo, com base na opinião de especialistas nas áreas de gerência de redes de computadores e infraestrutura;
c) Os dados foram, então, tabulados em planilhas e importados para tabelas de um banco de dados relacional, para que pudessem ser acessados por uma aplicação. Embora cada uma das tabelas utilizadas varie entre 60000 e 150000 linhas de requisições, não foi necessária a utilização de algo além de um gerenciador de banco de dados do porte do MySQL*CE*. Esse requisito mínimo garante que, mesmo em redes de pequeno porte e com baixo orçamento, esse sistema também possa ser utilizado;
d) Foram definidos os diversos graus de crença e incerteza para cada uma das faixas, tendo como objeto da proposição a operação normal da rede. As crenças foram determinadas através do processo de normalização dos valores dos atributos acumulados de cada um dos hosts em determinado intervalo de tempo. Intervalos de tempo maiores permitem um aprendizado mais consistente do comportamento dos hosts da rede, tendo em vista que conseguem abranger um maior número de situações da operação da rede, o que permite minimizar eventuais desvios;

e) O analisador do tráfego de dados foi utilizado para a extração dos padrões de funcionamento da rede, considerando os intervalos e os equipamentos em atividade. O comportamento de cada um dos hosts em operação foi confrontado com o que se espera de um comportamento normal da rede, cujos valores são altamente variáveis e dependentes de seus requisitos e projeto específico.

Com a informação obtida, foi possível mapear o funcionamento dos computadores da rede e, em caso de detecção de erros, executar as ações cabíveis. Atualmente, a abordagem das empresas considera que todo dispositivo de rede é um gerador potencial de problemas, portanto, um agente local deve ser instalado em cada equipamento para monitorar constantemente as atividades. Essa abordagem gera custos muitas vezes proibitivos, assim como tempo de processamento.

3.2 Comparação com modelos existentes

Para comparar este trabalho com os modelos existentes, é necessário detalhar os materiais e ferramentas utilizados no projeto desenvolvido. A fonte de informação pode ser adquirida de diferentes dispositivos de rede, como roteadores, proxies ou switches. Nesse caso, o processo de aprendizado reuniu valores de atributos dos logs de proxy de um roteador. Conforme mencionado anteriormente, cada *log* de dia de trabalho varia em tamanho, entre 60000 a 150000 registros, cada um representando uma solicitação de recurso externo por um determinado *host* de rede.

Os atributos foram normalizados para cada *host* que compõe a rede, considerando o intervalo analisado. Em seguida, foram aplicados conceitos da Lógica $E\tau$ para cada equipamento e foram determinadas a evidência favorável e contrária dos atributos. Com o auxílio de um analisador de tráfego de dados, tornou-se possível determinar o comportamento dos *hosts* da rede dentro de um intervalo de tempo específico. Finalmente, realizou-se uma análise global considerando os vários estados lógicos possíveis contemplados pela Lógica $E\tau$.

Ao contrário dos modelos de detecção de anomalias propostos por Zhu e Sastry (2011) e Yaacobet al. (2010), que contam com dados simulados para emular um ambiente de rede real e a geração sintética de anomalias, este projeto emprega

dados continuamente coletados a partir da operação de uma rede operacional real para o processo de aprendizagem. Outra diferença de Pena et al. (2014) é que este usa Assinatura Digital de Segmento de Rede com Análise de Fluxo (DSNSF), que estabelece um perfil para o desempenho normal de um segmento de rede considerando a história de seu movimento. Um possível problema: quando um sistema em tempo real não é considerado para esse tipo de tarefa, qualquer alteração na topologia da rede ou em sua disponibilidade pode afetar o processo de aprendizado do analisador, já que o histórico pode não representar o estado real da rede.

Outra diferença significativa do trabalho de Fernandes et al. (2015) é que o tráfego de dados foi utilizado como medida analítica, não havendo qualquer distinção de atributos individuais que poderiam representar diferentes situações operacionais da rede. Nesse projeto, os atributos de rede são tratados individualmente.

Para o desenvolvimento do analisador de tráfego de dados para detecção de comportamento anômalo em redes de computadores, foi necessário transformar os dados do *log* do *proxy* Squid, que é originalmente formatado em arquivos de texto simples, em um sistema de banco de dados relacional. A primeira etapa na obtenção dos dados foi converter cada um dos campos tabulados do arquivo de texto para o formato CSV (*comma-separated values*), o que pode ser feito com aplicativos de planilha comuns. A partir do arquivo obtido, foi possível gerar um banco de dados relacional. O framework *Hibernate* foi utilizado para realizar o mapeamento objeto-relacional (ORM), no qual o objetivo principal é reduzir a complexidade envolvida no desenvolvimento de aplicações que necessitam interagir com bancos de dados relacionais (BABU; GUNASINGH, 2016). Nesse caso, o banco de dados é convertido em objetos e pode ser acessado sem a necessidade de chamadas SQL explícitas, fazendo chamadas nativas. O analisador desenvolvido realizou o monitoramento em dois estágios distintos, definidos a seguir:

a) Avaliação sistêmica e detecção de intervalos críticos;
b) Detecção específica de anomalias de rede em intervalos críticos.

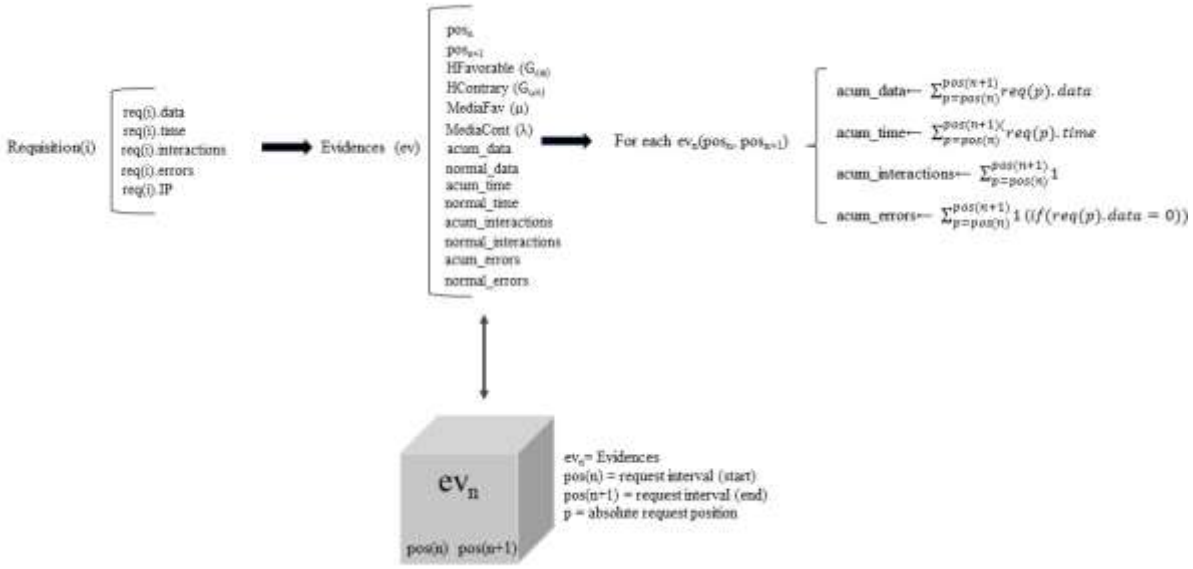
3.3 Avaliação sistêmica e detecção de intervalos críticos

Considerando que não se sabe em que momento um *host* pode realizar um comportamento não compatível com padrões normais, o primeiro passo foi realizar

uma avaliação abrangente da operação da rede. O intervalo inicial foi definido em 15 minutos, até o limite de 48 horas de operação contínua, limitado apenas à capacidade de processamento disponível. Inicialmente, observou-se que valores menores que 15 minutos não foram suficientes para garantir um processo de aprendizado aceitável do sistema para gerar resultados confiáveis, considerando que os dados ainda não eram suficientemente representativos. Por outro lado, um intervalo maior que 48 horas gerou um impacto negativo em termos de desempenho. Dadas as necessidades de desempenho, intervalos de 30 minutos foram usados neste projeto.

Nesse estágio, ainda não há indícios de quais *hosts* poderiam ser os potenciais geradores de problemas, uma vez que a ênfase está na detecção de um ou mais intervalos críticos em que comportamentos anômalos poderiam ocorrer. Como pode ser verificado, as evidências compreendem os campos dos valores acumulados de cada atributo de rede adquirido das requisições, bem como seus valores normalizados. Todos os atributos são contabilizados por acumulação ou contagem. Os *hosts* inativos não foram considerados, já que seus atributos sem valores, se computados, acabariam influenciando de maneira indesejada o processo de normalização de valores. O esquema de aquisição de atributos dos intervalos críticos é representado na Figura 6.

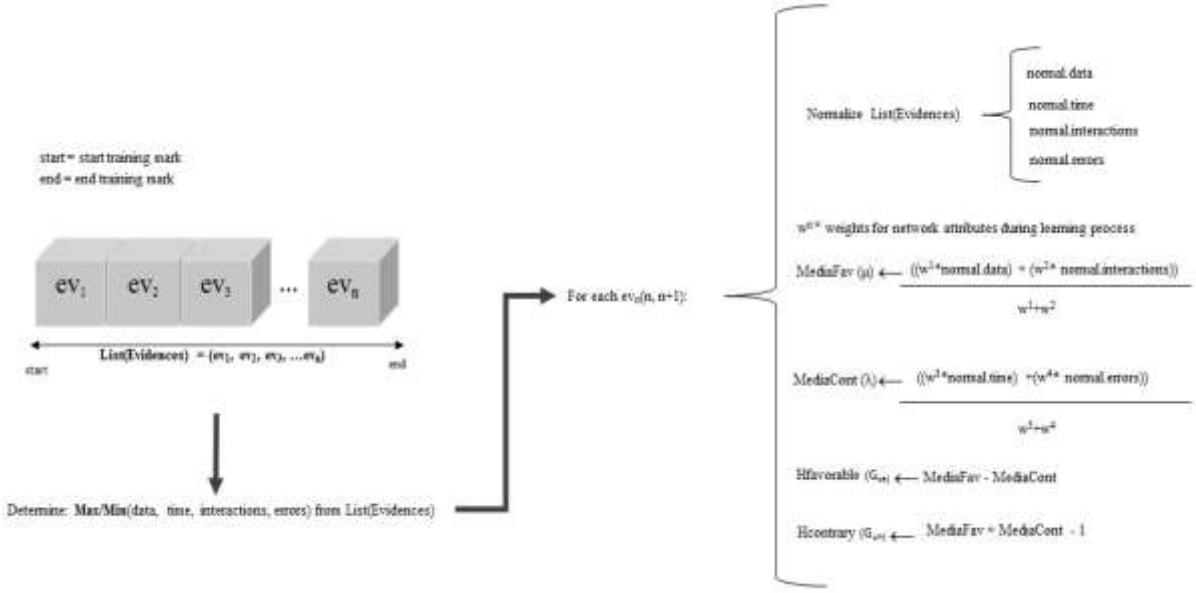
Figura 6 – Esquema de aquisição dos intervalos críticos para as evidências



Fonte: Autor.

Inicialmente, uma “Lista de Evidências” é gerada com base em um determinado intervalo. Assim, é possível obter os valores absolutos dos atributos sob análise. O passo seguinte foi normalizar os valores da lista gerada e determinar as evidências favoráveis (μ) e contrárias (λ), bem como os graus de certeza (G_{ce}) e incerteza (G_{un}), dos vários intervalos de tempo, como apresentado na Figura 7.

Figura 7 – Evidências favoráveis (μ) e contrárias (λ) e os graus de certeza (Gce) e incerteza (Gun) para a Lista de Evidências



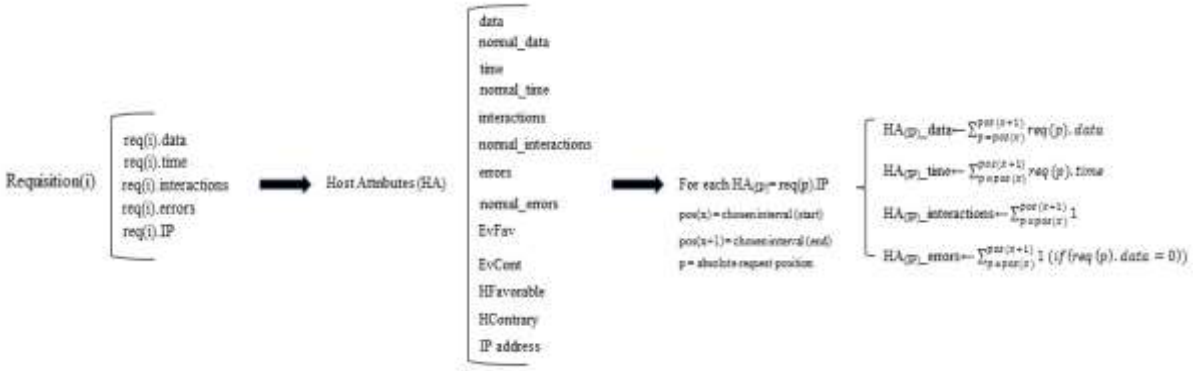
Fonte: Autor.

Uma abordagem possível e interessante pode ser a de considerar um ou mais intervalos com valores extremos de acordo com os conceitos da Lógica Eτ. Com o resultado do intervalo, o próximo passo foi a busca específica pelo *host* de comportamento anômalo.

3.4 Detecção específica de anomalias de rede em intervalos críticos

Após a determinação do intervalo, a primeira etapa foi adquirir todos os atributos de rede para cada um de seus *hosts*. Novamente, *hosts* inativos não foram considerados. Inicialmente, uma “Lista dos Atributos do Host” foi gerada com base nesse intervalo dado. Este esquema é apresentado na Figura 8:

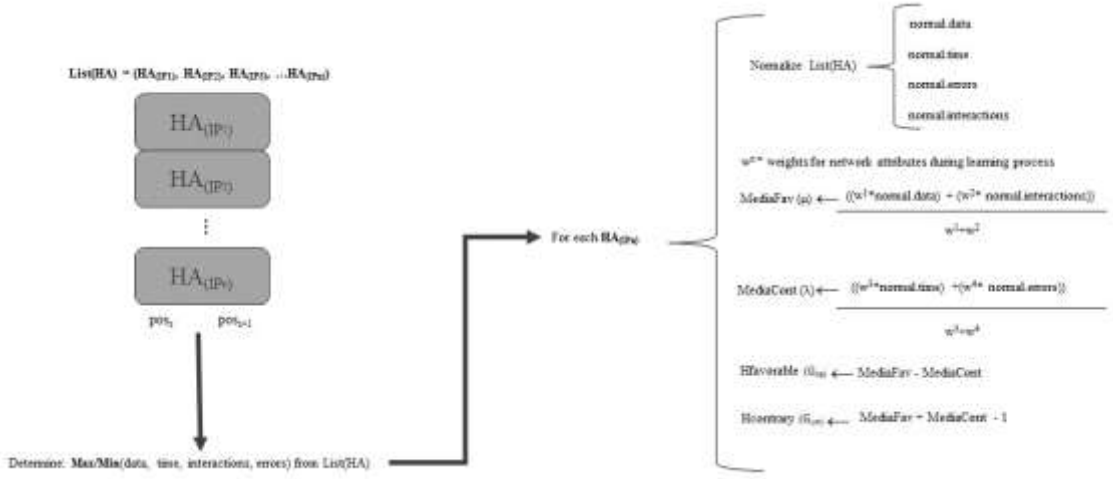
Figura 8 – Esquema de aquisição dos hosts da rede para os Atributos do Host



Fonte: Autor.

O passo seguinte foi normalizar os valores da lista e determinar as evidências favoráveis (μ) e contrárias (λ), bem como os graus de certeza (G_{ce}) e incerteza (G_{un}) dos *hosts* da rede, conforme apresentado na Figura 9:

Figura 9 – Evidências favoráveis (μ) e contrárias (λ) e os graus de certeza (Gce) e incerteza (Gun) para a Lista de Atributos do Host



Fonte: Autor.

4 RESULTADOS E DISCUSSÕES

4.1 Artigo 1

Trabalho apresentado no APMS 2015 International Conference Advances in Production Management Systems, Organizado pela International Federation of Information Processing, Musashi University, Shigeki Umeda, Prof., Egota-campus,1-26 Toyotama-kami Nerima, Tokyo 176-8534, Japan, September 5-9, 2015.

Publicado em: PIMENTA JR, A. P.; ABE, J. M. An Analyzer of Computer Networks Logs Based on Paraconsistent Logic. **IFIP Advances in Information and Communication Technology**, v. 460, p. 620–627, 2015. Disponível em <https://link.springer.com/chapter/10.1007/978-3-319-22759-7_71>

De acordo com o objetivo específico 1, neste artigo, foram determinados, segundo a opinião de especialistas na área de redes de computadores e infraestrutura, quais atributos e respectivos valores podem ser de interesse na avaliação de desempenho de uma rede de computadores. A importância dessa pesquisa decorre do fato de que, nos últimos anos, problemas de vulnerabilidade têm cada vez mais chamado a atenção para a questão do gerenciamento de informações na World Wide Web. A vulnerabilidade detectada não é restrita apenas a indivíduos, mas também a empresas e governos. Ao longo da última década, as redes tornaram-se uma maneira acessível para vários serviços de informática, mas também um grande desafio para os gerentes de rede manter sua operação. O principal problema é a dificuldade de lidar com a grande quantidade de dados gerados pelas solicitações dos usuários, o que, por sua vez, gera registros de informações cada vez maiores. Além disso, a dinâmica dos serviços pode levar a detectar falsos positivos e negativos, então, a incerteza é uma questão que deve ser considerada. O emprego da lógica clássica pode não ser adequado para resolver problemas dessa natureza, e a utilização da Lógica $E\tau$ mostrou-se adequada para esta finalidade.

An Analyzer of Computer Network Logs Based on Paraconsistent Logic

Avelino Palma Pimenta Jr.(✉), Jair Minoro Abe, and Cristina Corrêa de Oliveira

Graduate Program in Production Engineering, Paulista University, R. Dr. Bacelar 1212, São Paulo 04026-002, Brazil
appimenta@gmail.com, jairabe@uol.com.br, crisolive@ig.com.br

Abstract. In recent years, the network vulnerability events draw the attention to the issue of the information management on the World Wide Web. The detected vulnerability was not only restricted to individuals, but also to enterprises and governments. Over the past decade, networks have become an affordable way for several computer services, but also a major challenge for network managers to maintain its operation. The main problem is the difficulty to deal with big amount of data generated by user requests, which in turn ultimately generate increasing information logs. Moreover, the dynamics of the services can lead to detect false positive and negative ones, so uncertainty is a theme to be considered. The employment of classical logic may not be adequate to solve problems of this nature. The aim of this paper is to present the development of a Paraconsistent analyzer, in order to extract some computer networks patterns of interest.

Keywords: Paraconsistent logic · Computer networks · Pattern recognition · Decision-making

1 Introduction

The computer networks currently constitute as the main form of transmitting data and services. Therefore, the task of monitoring the information has turn to be a key factor in technology sectors [1]. The information security issues have existed around since it has been created. However, as the technology goes further and information management systems become increasingly powerful, the issue of information security becomes also increasingly critical [2].

Considering its intrinsic nature, the network operation analysis is based on stochastic events. The argument for this type of methodology is based on the principle that human actions behave as random elements [3]. In fact, the variability of available services is considerable, and therefore the types of user behavior eventually follow this trend.

Some important elements should be considered in data traffic management, such as trustfulness, confidentiality, integrity and reliability [4, 5].

S. Umeda et al. (Eds.): APMS 2015, Part II, IFIP AICT 460, pp. 620–627, 2015.
DOI: 10.1007/978-3-319-22759-7_71

Among the mentioned elements, reliability is the main object of analysis of this article. It can be defined as the capacity to provide access to information systems as soon as they are requested [4]. A system with low reliability ultimately leads to dissatisfaction and low user productivity.

The establishment of a set of criteria should be done to avoid false positives [6], which in turn may even lead to problems of a legal nature. For instance, a significant loss of network data packets can either be interpreted as a malicious attack, as may represent an intense use of the computer network.

It is possible to gather information from network logs of the data packets that pass through the network devices. Data extraction can provide the manager an important tool in decision making.

Some data may be considered interesting to the analysis of the packet traffic, among which are: the origin logical IP address, request time, response waiting time, type of obtained result, the amount of response data in the transaction and the destination logical IP address [7].

Due the stochastic behavior of the networks, the analysis methods based on classical logic may not be a suitable tool for this scenario [8]. A new logical system is needed to deal with it. Therefore, the Paraconsistent annotated evidential logic Eτ has a structure that becomes a natural technique to look for evidence of problems, whether caused both by the standard operation of the network or intentional elements [9]. In the latter case, it may be constituted by users or malicious application [10].

Once again, the use of Paraconsistent logic Eτ arises as a feasible alternative to take decisions under uncertainty, inconsistency and contradiction, in several areas such as robotics, electronics, traffic control, among others [11].

2 Methodology

The development of the proposal is based on the analysis of network data communication over five days and three ranges (mornings, afternoons and evenings), of five hours each. For each range, several parameters were obtained, among which: date and time of the request, the source IP address, destination IP address, type of connection made, the result of the request operation, response waiting time, amount of data response and total transactions.

From the network requests log, it was possible to extract network usage information expressed in Table 1.

Some significant information can be obtained considering the parameter "Standard Deviation" in association with "Average Response Time" as a measure of dispersion and "Average Packet Size". In this case, it is possible to make an association between the lowest standard deviation (86841.53 ms), its average response time (12579.59 ms) and average packets size (20589.08 bytes), which leads to believe that in the period from 13:00 to 17:59 on Tuesday presented the network operating normally, with low response time, even though with a considerable amount of data in transit. On Wednesday, from 18:00 to 22:59, the network had its worst performance, having

Table 1. Network parameters obtained from transactions logs

Day of week	Range	Events	Total transactions	Average response time (ms)	Standard deviation of average response time (ms)	Average packet size (bytes)
Monday	8:00 - 12:59	1 até 78637	78637	13421.17734	110597.16	40972.0802
	13:00 - 17:59	78638 até 175333	96771	17349.60840	125964.0304	16631.58440
	18:00 - 22:59	175334 até 253121	80787	27850.16981	179844.283	22746.42033
Tuesday	8:00 - 12:59	1 até 44670	44670	18734.32171	123942.0828	35573.27189
	13:00 - 17:59	44671 até 112314	69643	12776.39023	86046.23460	23559.08378
	18:00 - 22:59	112315 até 148376	33061	34218.35856	117616.3025	20740.04520
Wednesday	8:00 - 12:59	1 até 53908	53908	14365.10390	102527.2689	21270.15991
	13:00 - 17:59	53909 até 103969	33061	16172.74776	158488.8083	44461.45029
	18:00 - 22:59	103969 até 110815	54646	29514.48540	246460.6714	26382.09081
Thursday	8:00 - 12:59	1 até 52219	52219	29918.62059	110934.34	52906.31
	13:00 - 17:59	52220 até 159662	107543	16836.37154	89652.21535	64440.55643
	18:00 - 22:59	159663 até 196237	36574	33557.5653	375607.2837	27913.58398
Friday	8:00 - 12:59	1 até 37178	37178	17748.37259	888744.7718	37854.26786
	13:00 - 17:59	37179 até 140238	106839	9947.793332	122515.1212	69712.03777
	18:00 - 22:59	140239 até 196949	56610	36828.41366	317288.393	51803.5317

obtained the largest delay in average response time (29514.48 ms) and slightly higher average packets size compared to the previous example (26382.09 bytes), with a standard deviation slightly below the maximum limit obtained (246460.67 ms). In this case, it may be viable to conclude that the network had dealt with operations problems.

However, during the computer network operation, handle dynamic and highly stochastic events may be a high complexity task. Therefore, a logical analyzer – Para-analyzer [12] will be used upon the data obtained to make an analysis under the light of an artificial intelligence tool. Four parameters shall be used as factors: average response time (R), its standard deviation (D), average packets size (P) and the total transactions (T).

The number of intervals that were selected for each parameter is based on the occurrence of significant variances in the evaluations of favorable and unfavorable evidences by the specialists. A larger number of intervals often presented very close or even repeated values, which in turn would generate unnecessary redundancy in this study.

It is considered that a low response time is a good indicator because it suggests that the network did not suffer consequences of a possible congestion and was able to answer its requests in an acceptable time. For this, three intervals shall be considered, based on the minimum and maximum values obtained from the network log: R1, R2 and R3.

A low standard deviation of the average response time also leads to the belief of a homogeneous network operation. In other words, no significant discrepancies between the hosts in operation were detected. Along with the previous factor, three intervals shall be considered: D1, D2 and D3.

The average packet size is also an important factor, but it has an element of uncertainty that must be considered. Networks with low average size packets may indicate little use, which can be considered a plus. Moreover, networks that suffer attacks should also have this tendency, since the data packets used for this purpose are individually small. Four intervals will be considered: P1, P2, P3 and P4.

Finally, the number of transactions may be considered a significant factor since a high value may suggest problems relating to malicious attacks or high degree of utilization of the network. Once again, four intervals shall be used: T1, T2, T3, and T4.

The concepts of Paraconsistent logic Eτ will be used from this point. According to Abe [12]: "The atomic formulas of the logic Eτ are of the type p(μ, λ), where (μ, λ) ∈ $[0, 1]^2$ and [0, 1] is the real unitary interval (p denotes a propositional variable)". Therefore, p(μ, λ) can be intuitively read: "It is assumed that p's favorable evidence is μ and contrary evidence is λ.". This will lead to the following conclusion:

- $p_{(1.0,\ 0.0)}$ can be read as a true proposition,
- $p_{(0.0,\ 1.0)}$ as false,
- $p_{(1.0,\ 1.0)}$ as inconsistent,
- $p_{(0.0,\ 0.0)}$ as paracomplete, and
- $p_{(0.5,\ 0.5)}$ as an indefinite proposition.

To determine the uncertainty and certainty degrees, the formulas are [10]:

- Uncertainty degree: $G_{un}(\mu, \lambda) = \mu + \lambda - 1$ $(0 \leq \mu, \lambda \leq 1)$;
- Certainty degree: $G_{ce}(\mu, \lambda) = \mu - \lambda$ $(0 \leq \mu, \lambda \leq 1)$;

An order relation is defined on $[0, 1]^2$: $(\mu_1, \lambda_1) \leq (\mu_2, \lambda_2) \Leftrightarrow \mu_1 \leq \mu_2$ and $\lambda_1 \leq \lambda_2$, constituting a lattice that will be symbolized by τ.

With the uncertainty and certainty degrees, it is possible to manage the following 12 output states, showed in the Table 2.

Table 2. Extreme and nn-extreme states

Extreme States	Symbol	Non-extreme states	Symbol
True	V	Quasi-true tending to Inconsistent	QV→T
False	F	Quasi-true tending to Paracomplete	QV→⊥
Inconsistent	T	Quasi-false tending to Inconsistent	QF→T
Paracomplete	⊥	Quasi-false tending to Paracomplete	QF→⊥
		Quasi-inconsistent tending to True	QT→V
		Quasi-inconsistent tending to False	QT→F
		Quasi-paracomplete tending to True	Q⊥→V
		Quasi-paracomplete tending to False	Q⊥→F

All states are represented in Fig. 1.

Initially, for each analyzed factor, the opinions of two experts in the field of networks shall be considered, both senior professional with a large experience in the field. For each factor, intervals will be taken and rated, with a certain degree of favorable evidence (represented by μ) and unfavorable evidence (represented by λ).

Also weights to each factor/intervals will be applied, considering the importance degree that each expert deems appropriate. The data from which the Paraconsistent algorithm will be applied is applied can be expressed in Table 3.

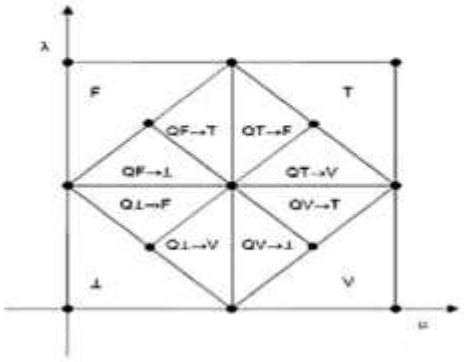


Fig. 1. All states in Lattice τ

Table 3. Distribution of factors and grades for the Para-analyzer algorithm

Factor	Interval	Values	Tender Specialist 1		Tender Specialist 2	
			μ	λ	μ	λ
Response Time	R1	< 16666 ms	0.95	0.3	0.9	0.1
	R2	16667 - 23333 ms	0.65	0.45	0.7	0.4
	R3	> 23333 ms	0.45	0.65	0.55	0.75
Standard Deviation of the Average Response Time	D1	< 147999 ms	0.9	0.1	0.9	0.1
	D2	148000 - 233999 ms	0.55	0.3	0.55	0.45
	D3	> 234000	0.2	0.8	0.3	0.8
Average Packets Size	P1	< 19999 bytes	0.7	0.2	0.75	0.3
	P2	20000 - 29999 bytes	0.6	0.4	0.65	0.45
	P3	30000 - 39999 bytes	0.3	0.6	0.35	0.55
	P4	> 40000 bytes	0.2	0.7	0.3	0.85
Transactions	T1	< 39999	0.9	0.2	0.9	0.25
	T2	40000 - 59999	0.7	0.35	0.8	0.5
	T3	60000 - 79999	0.55	0.5	0.6	0.45
	T4	> 80000	0.3	0.8	0.25	0.8

To study the proposition: "The computer network is functioning within normal operational limits", values were tabulated and applied for the Para-analyzer algorithm, as seen in Table 4.

The factors listed above are not able to lead to important conclusions alone. In this case, the combined influence of the factors, with their respective applied weights, could contribute to a more appropriate response to the initial proposition. This is determined by the global analysis of the points that represent the Cartesian plane [13].

The global analysis is calculated considering the favorable evidences (μ) multiplied by their respective weights, and finally added. The same is done to the unfavorable evidence (λ) [13]. Considering the tabulated values, the global analysis obtained was 0.63 of favorable evidence and 0.48 of unfavorable evidence. With a minimum demand level of 0.5, it was observed that the factors were proved feasible for the R1 response time, D1 standard deviation of average response time, and T1 transactions. No average size of packets (P) interval showed viable result, as seen in Fig. 2.

Table 4. Favorable and unfavorable evidences and weights of first scenario

Factor analysis	Interval	Weight	Favorable Evidence Degree	Unfavorable Evidence Degree
Response Time	R1	2	0.9	0.1
	R2	2	0.7	0.45
	R3	2	0.55	0.75
Standard Deviation of the Average Response Time	D1	2	0.9	0.1
	D2	1	0.55	0.5
	D3	2	0.3	0.8
Average Packets Size	P1	1	0.75	0.3
	P2	1	0.65	0.45
	P3	1	0.35	0.6
	P4	1	0.3	0.9
Transactions	T1	2	0.9	0.25
	T2	2	0.8	0.35
	T3	2	0.6	0.5
	T4	2	0.3	0.8

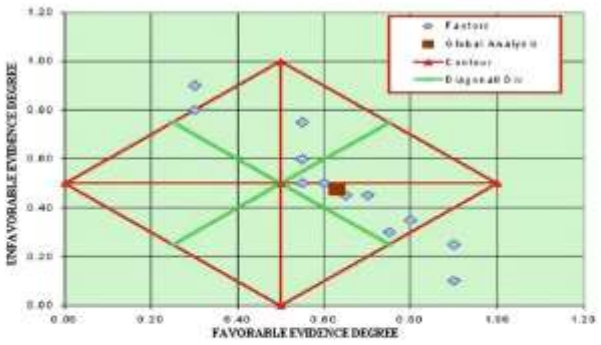


Fig. 2. Analysis of first scenario result by the Para-analyzer algorithm.

For comparison, another set of weights can be used where a higher weight is applied to each extreme position of the analyzed factor interval. The objective of this approach is to balance the weight factor to each other while applying a slightly lower relative weight in the intermediate intervals that may generate a higher level of uncertainty, as seen in Table 5.

In this second scenario, the obtained global analysis was 0.62 of favorable evidence and 0.49 of unfavorable evidence, which is slightly less than in the first scenario. With a minimum demand level of 0.5, it was observed that the factors that were viable remain the same: R1 response time, D1 standard deviation of average response time, and T1 transactions. Again, no average packets size factor interval (P) presented viable result, as can be seen in Fig. 3.

Table 5. Favorable and unfavorable evidences and weights of second scenario.

Factor analysis	Interval	Weight	Favorable Evidence Degree	Unfavorable Evidence Degree
Response Time	R1	2	0.9	0.1
	R2	1	0.7	0.45
	R3	2	0.35	0.75
Standard Deviation of the Average Response Time	D1	2	0.9	0.1
	D2	1	0.55	0.5
	D3	2	0.3	0.8
Average Packets Size	P1	2	0.75	0.3
	P2	1	0.45	0.45
	P3	1	0.55	0.6
	P4	2	0.3	0.9
Transactions	T1	2	0.9	0.25
	T2	1	0.8	0.35
	T3	1	0.6	0.5
	T4	2	0.3	0.8

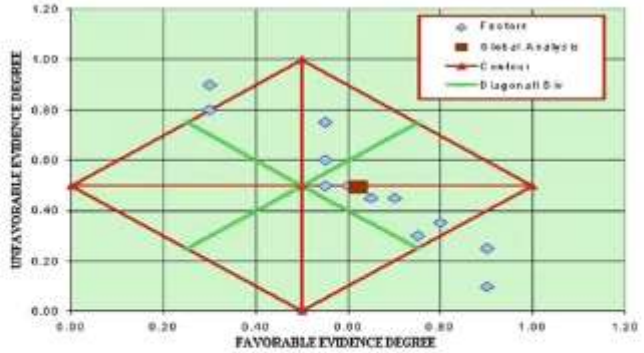


Fig. 3. Analysis of second scenario result by the Para-analyzer algorithm.

3 Analysis of the Results

From the obtained results, it can be observed that among the analyzed factors, the intervals R1, D1 and T1 gathered a common standard of viability. On the other hand, there was no significant influence on the factor P, in any of the intervals. All the evaluated scenarios showed inconclusive results.

The interpretation of the results leads to the belief that a network with reduced response time (R1), a low standard deviation of the average response time (D1) and small number of transactions (T1) are conditions that reflect the behavior of the computer network within normal limits. However, the average size factor package does not follow the same line of reasoning, and can be proven by its own data in the log, where a significant amount of data in transit was verified with a reduced response time. Therefore, it can be concluded that the average of the data packets may not be indicative of problems in the network, only an indication of intensive use of the infrastructure.

References

1. Lin, Y.K., Huang, C.F.: Stochastic computer network under accuracy rate constraint from QoS viewpoint. Inf. Sci. (Ny) **239**, 241–252 (2013)
2. White, D., Rea, A.: A backpropagation neural network for computer network security. J. Comput. Sci. **2**, 710–715 (2006)
3. Ben-Porat, U., Bremler-Barr, A., Levy, H.: Computer and network performance: graduating from the "age of innocence". Comput. Netw. **66**, 68–81 (2014)
4. Kurose, J.F., Ross, K.W.: Computer Networking A Top-Down Approach Featuring the Internet. Pearson Education, London (2005)
5. Rosen, R.: Linux kernel networking advanced topics : neighboring and IPsec (2008)
6. Fossaceca, J.M., Mazzuchi, T.A., Sarkani, S.: MARK-ELM: application of a novel multiple kernel learning framework for improving the robustness of network intrusion detection. Expert Syst. Appl. **42**, 4062–4080 (2015)
7. Rousskov, A., Soloviev, V.: A performance study of the Squid proxy on HTTP/1.0. World Wide Web **2**, 47–67 (1999)
8. Fernandez-Prieto, J.A., Canada-Bago, J., Gadeo-Martos, M.A., Velasco, J.R.: Optimisation of control parameters for genetic algorithms to test computer networks under realistic traffic loads. Appl. Soft Comput. J. **12**, 1875–1883 (2012)
9. Abe, J.M.: Foundations of annotated logics. PhD thesis, University of São Paulo, Brazil (1992). (in Portuguese)
10. Misra, A.K., Verma, M., Sharma, A.: Capturing the interplay between malware and anti-malware in a computer network. Appl. Math. Comput. **229**, 340–349 (2014)
11. Da Silva Filho, J.I., Torres, G.L., Abe, J.M.: Uncertainty Treatment Using Paraconsistent Logic - Introducing Paraconsistent Artificial Neural Networks, vol. 211, p. 328. IOS Press, Holanda (2010). doi:10.3233/978-1-60750-558-7-I. ISBN 978-1-60750-557-0
12. Abe, J.M.: Paraconsistent logics and applications. In: Proceedings of 4th International Workshop on Soft Computing Applications, Arad, Romênia 1–18, ISBN 9781424479832, IEEE CFP1028D-CDR (2010)
13. Da Silva Filho, J.I., Abe, J.M.: Paraconsistent analyzer module. Int. J. Comput. Anticipatory Syst. **9**, 346–352 (2001). ISSN 1373-5411, ISBN 2-9600262-1-7

4.2 Artigo 2

Trabalho apresentado no APMS 2016 International Conference Advances in Production Management Systems, organizado pelo Working Group 5.7 (WG5.7) in IFIP (International Federation of Information Processing) Iguassu Falls, Shigeki Umeda, Prof., Conference Theme**:** “Production Management Initiatives for a Sustainable World”, Hotel Bourbon Cataratas, Brazil, September 3-7, Sep., 2016.

Publicado em: PIMENTA JR, A. P.; ABE, J. M. Determination of operating parameters and performance analysis of computer networks with Paraconsistent Annotated Evidential Logic Eτ. **IFIP Advances in Information and Communication Technology**, v. 1, p. 1–9, 2016. Disponível em <https://link.springer.com/chapter/10.1007/978-3-319-51133-7_1>

De acordo com o objetivo específico 2, este artigo, considerado como a melhor apresentação do APMS 2016, teve como objetivo parametrizar e avaliar os elementos operacionais de redes heterogêneas partindo-se da análise de atributos representativos fornecidos pela própria operação da rede, com base em conceitos de LógicaEτ.

A necessidade do desenvolvimento dessa pesquisa decorre do fato de que as redes de computadores têm duas características importantes: a grande diversidade de dispositivos de conexão e uma grande variabilidade da distribuição física de equipamentos. Portanto, a análise de desempenho de uma rede específica baseada em referências absolutas ou de terceiros pode não ser aplicável em todas as circunstâncias, especialmente em redes altamente complexas e heterogêneas. De fato, esse tipo de análise traz consigo um alto grau de incerteza e a lógica clássica pode não ser apropriada para lidar com isso.

Determination of Operating Parameters and Performance Analysis of Computer Networks with Paraconsistent Annotated Evidential Logic Eτ

Avelino Palma Pimenta Junior(✉), Jair Minoro Abe, and Genivaldo Carlos Silva

Graduate Program in Production Engineering, Paulista University, R. Dr. Bacelar 1212, São Paulo 04026-002, Brazil
appimenta@gmail.com, jairabe@uol.com.br, gcsilva@ig.com.br

Abstract. Computer networks have two important characteristics: the vast diversity of connecting devices and a great variability of the physical distribution of equipments. Therefore, the performance analysis of a specific network based on absolute references or third parties may not be applicable in all circumstances, especially in highly complex and heterogeneous networks. Indeed, it carries a high degree of uncertainty, and the classical logic may not be appropriate to deal it. This paper aims to parameterize and evaluate the operating elements of heterogeneous networks, from the analysis of representative attributes, based on concepts of Paraconsistent Annotated Evidential Logic Et. AQ1

Keywords: Paraconsistent logic · Computer networks · Network parameterization · Pattern recognition

1 Introduction

Computer networks are currently used in most companies, and represent an important means of interoperability and data communication. As the World Wide Web and users are explicating at a very rapid rate, the performance of World Wide Web systems become rapidly high [1]. Since its inception, the foundation for the deployment of networks pointed to a variety of devices from different manufacturers and architectures, and often operates at varying speeds. The different links in the local area network can operate at different speeds and can run at different medias, such as 1 Gbps or 100 Mbps, copper or fiber [2]. The copper-based communications encode data via electrical impulses, unlike the optical fiber that uses light signals for this purpose. The existence of these two physical means of data communication, in varying degrees of use, must be rendered compatible. However, the communication of heterogeneous systems is not always an easy task, and it is not always possible to obtain optimal and predictable results.

A computer network consists of several connected hosts, which can be represented by a desktop, a laptop, a smartphone, among others. In such an heterogeneous client

I. Nääs et al. (Eds.): APMS 2016, IFIP AICT 488, pp. 1–9, 2016.
DOI: 10.1007/978-3-319-51133-7_1

environment, efficient content adaptation and delivery services are becoming a major requirement for the new Internet service infrastructure [3]. However, many of these equipments may have different architectures, and also use different operating systems and applications.

All the previous elements are part of computer networks, but also constitute as conflict elements, which makes it even more difficult to measure the performance of a network. Typical evaluation methods, such as benchmark performance, however, are limited in applicability. Often they are not representative of the traffic characteristics of any customer facility [4]. The issue of uncertainty, therefore, should be considered. A possible solution could be the analysis of experts in the field of computer networks. This approach may not be suitable for all cases, since not always the professional knows profoundly the network to be analyzed. Moreover, although differences exist, some elements are common in network communications. For the establishment of network communication, there must always be a request from the side of the "client". It is a typical protocol of request-response, which controls the data transfer between server and client (such as a web browser) [5].

This request, when answered by the side of the "server" – typically a proxy, produces a corresponding response. Proxy servers are designed with three goals: decrease network traffic, reduce user (client) perceived lag, and reduce loads on the origin servers [6].

Every request from the client passes through the proxy server, which in turn may or may not modify the client request based on its implementation mechanism [7]. This response is accompanied by several attributes that can be used to analyze network performance. The most representative attributes may be used as a means of determining the network operating parameters. This work aims to analyze and detect problems in a computer network from a public university with about two hundred hosts, divided into two different departments (academic and administrative) with the aid of Paraconsistent Logic. In the academic department, there are six computer labs with twenty hosts each, plus two coordination rooms, with the total of ten hosts each. In the administrative department, five operating rooms, with approximately fifty hosts, as well as servers, routers and switches, all connected by copper or fiber optic links, and operating for fifteen hours a day, five days a week. Each department has different needs and use different services and applications. Therefore, it is clear the high degree of heterogeneity and uncertainty of the analyzed scenario, which makes it appropriate to use a non-classical logic, the subject of this paper.

2 Methodology

Responsive service plays a critical role in determining end-user satisfaction. In fact a customer who experiences a large delay after placing a request at a business's web server often switches to a competitor who provides faster service [8]. Network infrastructure needs to be constantly improving to satisfy QoS (Quality of Service) users demand, including both technology aspects (e.g. fastest links, proxies and servers) and related software [9].

To parameterize the operation of the network, a day of operation shall be monitored, during 15 h, divided into 30-minute intervals. Some of the most significant attributes shall be used, such as:

- Total network packets (bytes).
- Total response time (ms).
- Average speed (bytes/ms).
- Number of requests.
- Number of zero bytes responses.

From the network logs, it is possible to extract the values of the attributes, shown in Table 1:

The first attribute is used to analyze the response time (in milliseconds) related to the conducted requests. The second attribute is related to the volume of data (in bytes) that was requested in a given interval. At first, one might think that the higher the value, the more efficient the network operation. However, this attribute is loaded of uncertainty, considering that it can also denote network congestion. The third attribute range is calculated based on the first two, by simple arithmetic average, to calculate the use of network bandwidth. The fourth attribute is the number of requests that occurred in a given interval. This attribute itself is not enough to determine the level of the network quality. A network with many requests may indicate either a good performance or a high rate of retransmissions, which is considered undesirable. The fifth attribute is especially important when considered in conjunction with the fourth attribute, as it allows differentiating situations where there is large number of retransmissions. The obtained values of the attributes are then tabulated and normalized in the range from 0 to 1. For a contextualized view, the image of Fig. 1 can give a good idea of network operation from two significant parameters: average speed and number of zero bytes responses:

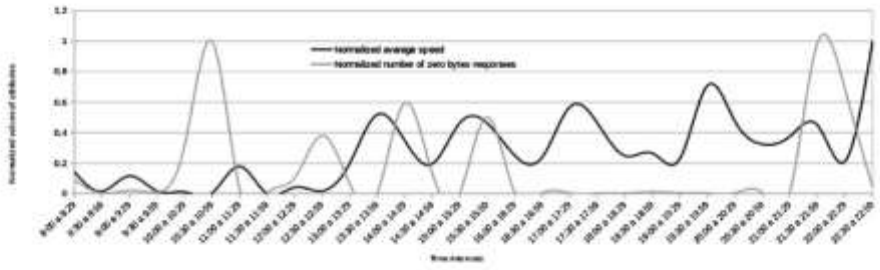


Fig. 1. Comparison between average speed and number of zero bytes responses

With the values obtained, it is possible to analyze specific scenarios in the operation of a network, through the development of a ranking of the evidence (favorable or unfavorable) using the Paraconsistent Annotated Evidential Logic Eτ.

The concepts of Paraconsistent Logic Eτ will be used from this point. According to Abe [10]: "The atomic formulas of the logic Eτ are of the type $p(\mu, \lambda)$, where $(\mu, \lambda) \in [0, 1]^2$ and [0, 1] is the real unitary interval (p denotes a propositional variable)".

Table 1. Atributes values obtained from a day operation of a computer network

Hour interval	Total network packets (bytes)	Total response time (ms)	Average speed (bytes/ms)	Number of requests	Number of zero bytes responses
8:00 a 8:29	101550313	186703410	0,5439124706	3311	779
8:30 a 8:59	101317599	384871739	0,2632502954	4515	32
9:00 a 9:29	144107833	296218480	0,4864917037	5020	201
9:30 a 9:59	149058945	558951986	0,2666757588	10348	84
10:00 a 10:29	153643549	603540143	0,2545705547	13126	2705
10:30 a 10-59	129625661	535538428	0,2420473569	18442	8644
11:00 a 11:29	113215036	181009325	0,6254652129	6829	296
11:30 a 11:59	98916878	429472435	0,2303218319	2671	40
12:00 a 12:29	89950808	281068865	0,3200312066	5051	894
12:30 a 12:59	93957712	348408989	0,2696764864	6844	3304
13:00 a 13:29	40352244	60526974	0,6666819987	1489	568
13:30 a 13:59	34759397	25246230	1,3768153503	1786	7
14:00 a 14:29	82984378	82816003	1,0020331215	8493	5147
14:30 a 14:59	103544699	156568116	0,6613396242	5180	1180
15:00 a 15:29	97323535	77590646	1,2543204628	4090	19
15:30 a 15:59	111349090	88934444	1,2520356005	9973	4345
16:00 a 16:29	116516110	148779326	0,7831471827	8299	59
16:30 a 16:59	134981701	177338304	0,7611536704	9268	43
17:00 a 17:29	101774848	98992388	1,0281078177	6730	36
17:30 a 17:59	84745862	67398212	1,2573903593	3868	28
18:00 a 18:29	63605693	81593640	0,7795422903	5449	38
18:30 a 18:59	92411148	113160272	0,8166395005	5153	109
19:00 a 19:29	91532492	124104104	0,7375460525	2359	55
19:30 a 19:59	200608215	111540378	1,798525508	4727	37
20:00 a 20:29	255225540	199250269	1,2809294626	5517	49
20:30 a 20:59	184581912	194732439	0,9478744936	4061	44
21:00 a 21:29	159659251	150403821	1,0615371999	3676	146
21:30 a 21:59	119997798	98105026	1,2231564772	12739	8554
22:00 a 22:29	126283972	180791028	0,6985079591	10007	5917
22:30 a 22:59	170579432	69887729	2,4407636997	4500	398

Therefore, $p(\mu, \lambda)$ can be intuitively read: "It is assumed that p's favorable evidence is μ and unfavorable evidence is λ.". This will lead to the following conclusion:

- $p_{(1.0,\ 0.0)}$ can be read as a true proposition,
- $p_{(0.0,\ 1.0)}$ as false,
- $p_{(1.0,\ 1.0)}$ as inconsistent,
- $p_{(0.0,\ 0.0)}$ as paracomplete, and
- $p_{(0.5,\ 0.5)}$ as an indefinite proposition.

To determine the uncertainty and certainty degrees, the formulas are [11]:

- Uncertainty degree: $G_{un}(\mu, \lambda) = \mu + \lambda - 1$ $(0 \leq \mu, \lambda \leq 1)$;
- Certainty degree: $G_{ce}(\mu, \lambda) = \mu - \lambda$ $(0 \leq \mu, \lambda \leq 1)$;

An order relation is defined on $[0, 1]^2$: $(\mu_1, \lambda_1) \leq (\mu_2, \lambda_2) \Leftrightarrow \mu_1 \leq \mu_2$ and $\lambda_2 \leq \lambda_1$, constituting a lattice that will be symbolized by τ.

With the uncertainty and certainty degrees, it is possible to manage the following 12 output states, showed in the Table 2.

Table 2. Extreme and non-extreme states

Extreme states	Symbol	Non-extreme states	Symbol
True	V	Quasi-true tending to Inconsistent	QV→T
False	F	Quasi-true tending to Paracomplete	QV→⊥
Inconsistent	T	Quasi-false tending to Inconsistent	QF→ T
Paracomplete	⊥	Quasi-false tending to Paracomplete	QF→⊥
		Quasi-inconsistent tending to True	QT→V
		Quasi-inconsistent tending to False	QT→F
		Quasi-paracomplete tending to True	Q⊥→V
		Quasi-paracomplete tending to False	Q⊥→F

All states are represented in Fig. 2:

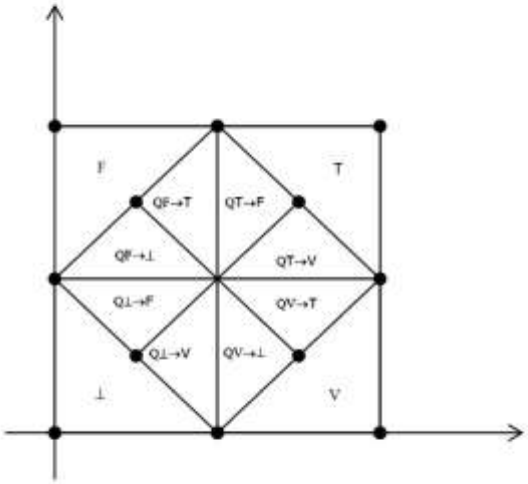


Fig. 2. Decision-making states of lattice τ

Based on the values of the attributes, obtained from one day operation of the computer network, two different scenarios from two time intervals on another day of operation will be analyzed in order to verify the operation of the network.

In the selected intervals, the following values were obtained, as shown in Table 3:

Table 3. Network attributes from two assessed scenarios

Scenarios	Total network packets (bytes)	Total response time (ms)	Average speed (bytes/ms)	Number of requests	Number of zero bytes responses
Scenario1	99646060	228119138	0,4368158712	4086	40
Scenario2	126428976	76538921	1,6518259514	11238	5532

A computer network that is operating at high speeds within its parameters is taken as favorable evidence. Therefore the average speed attribute can be considered a directly proportional greatness. This argument can also be applied to the number of requests attribute, since it indicates that the network has been operated in full working capacity to meet the user demands. In what concerns the zero byte responses attribute, the opposite occurs, as a network with high non responses indicates that the searched resources could not be found, thus it can be considered an inversely proportional greatness.

In both evaluated scenarios, the attribute values shall be normalized based on the operating values of the computer network. These values shall be used as degrees of favorable evidence for the average speed and number of requests attributes, as directly proportional greatnesses. The opposite shall be applied to the number of zero bytes responses attribute. In this case, the favorable evidence shall be defined as its denial. The favorable (μ) and unfavorable (λ) degree evidences are taken from the normalized values of the attributes, and are presented in Table 4:

Table 4. Normalized values and favorable (μ) and unfavorable (λ) evidences of the attributes

Scenarios	Normalized average speed (attribute 1)	Normalized number of requests (attribute 2)	Normalized number of zero bytes responses (attribute 3)	Attribute 1 evidences		Attribute 2 evidences		Attribute 3 evidences	
				μ	λ	μ	λ	μ	λ
Scenario1	0,093417539	0,1565117821	0,0038207711	0,9	0.91	0,15	0,85	100	0
Scenario2	0,643085955	0,5875369132	0,3696885493	0,64	0,36	0,58	0,42	0,36	0,64

After the parameterization of the network attributes, the proposition "The computer network is functioning within its normal operating values?" shall be analyzed. For this purpose, the Para-analyzer will be applied, representing scenarios 1 and 2, respectively in Figs. 3 and 4:

AQ2

The global analysis is calculated considering the favorable evidences (μ) multiplied by their respective weights (all equal, in both scenarios), and finally added. The same is done to the unfavorable evidence (λ) [11].

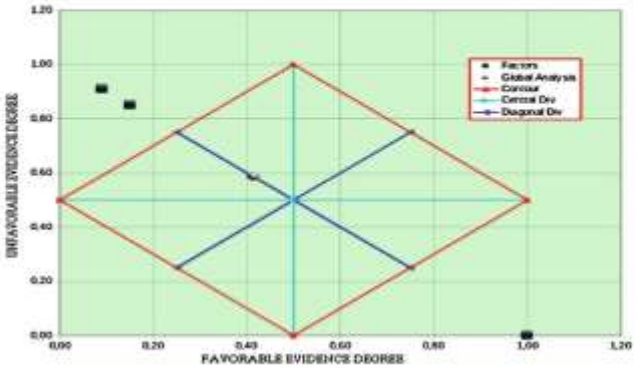


Fig. 3. Analysis of scenario 1 result by the Para-analyzer algorithm

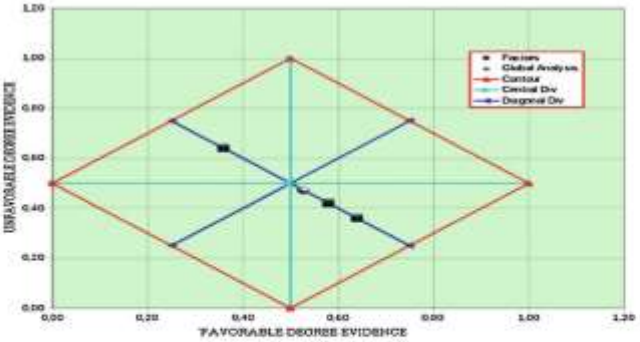


Fig. 4. Analysis of scenario 2 result by the Para-analyzer algorithm

3 Analysis of the Results

In scenario 1, the global analysis presents a quasi-false result tending to paracomplete and inconsistent to the normal network performance. Although the number of zero bytes responses attribute has high favorable evidence, this was not enough to represent a standard operation, since the other two attributes have not been sufficient to support the results. Diagnosis: the analyzed network in scenario 1 is not congested due to the low number of requests and is able to locate the searched resources. Abnormally, it still functions in low speed, which leads to the conclusion that the network is underutilized, or the network infrastructure project was oversized.

In scenario 2, the global analysis presents a quasi-true result, tending to paracomplete and inconsistent to the normal network performance. The high average speed and number of requests presents a situation of full use of the network capacity. However, it is observed that it begins to show clear signs of degradation due to the high number of zeros bytes responses. Diagnosis: the analyzed network in scenario 2 operates in a high degree of utilization, with early congestion signals and performance degradation.

4 Conclusion

As seen in both presented scenarios, the determination of the parameters in a computer network is a complex task. By their uncertainty and contradictory characteristics, and its dynamic operation, the Paraconsistent Annotated Evidential Logic Eτ emerges as an important tool for analysis of this type of environment.

Some possible solutions for scenario 1:

- Downsizing: sale or exchange of network devices (adapters, switches, routers) whose nominal capacity is beyond the need of the network.
- When possible, sharing or assignment of the installed infrastructure to another company or institution.
- Outsourcing services for companies that do not wish to have their own infrastructure.

Some possible solutions for scenario 2:

- Determining whether the congestion problem is systemic or occurs at only a few hosts. This can be done with the use of the Para-analyzer in different hosts of the network, and comparing the results with those obtained initially from the operating parameters.
- If the problem occurs at only few hosts, the solution is the physical or logic correction of the affected host(s). This task is usually simple, and its resolution is performed by a computer technician.
- If the problem is systemic, the analysis shall consider the possibility of upgrading (where possible) or even exchange of switches or routers by other with higher capacity.

References

1. Benadit, P.J., Francis, F.S.: ScienceDirect improving the performance of a proxy cache using very fast decision tree classifier. Procedia Comput. Sci. **48**, 304–312 (2015)
2. Kurose, J.F., Ross, K.W.: Computer Network: a Top-Down Approach, 6th edn. Addison-Wesley, Boston (2013)
3. Canali, C., Cardellini, V., Lancellotti, R.: Content adaptation architectures based on squid proxy server. World Wide Web **9**, 63–92 (2006)
4. Davison, B.D., Wu, B.: Implementing a web proxy evaluation architecture (2004)
5. Sysel, M., Doležal, O.: An educational HTTP proxy server. Procedia Eng. **69**, 128–132 (2014)
6. Romano, S., ElAarag, H.: A neural network proxy cache replacement strategy and its implementation in the squid proxy server. Neural Comput. Appl. **20**, 59–78 (2010)
7. Agarwal, T., Leonetti, M.A.: Design and Implementation of an IP based authentication mechanism for Open Source Proxy Servers in Interception Mode (2013)
8. Austin, I.B.M., Road, B., Tx, A., Rajamony, R., Elnozahy, M.: Measuring client-perceived response times on the WWW. In: 3rd Conference on USENIX Symposium on Internet Technologies Systems. 16 (2001)

9. Cárdenas, L.G., Sahuquillo, J., Pont, A., Gil, J.A.: The multikey web cache simulator: a platform for designing proxy cache management techniques. In: Proceedings of the 12th Euromicro Conference on Parallel, Distributed and Network-Based Processing, pp. 390–397 (2004)
10. Abe, J.M., Akama, S., Nakamatsu, K.: Introduction to Annotated Logics - Foundations for Paracomplete and Paraconsistent Reasoning. Springer, Heidelberg (2015)
11. Abe, J.M.: Paraconsistent logics and applications. In: 4th International Workshop on Soft Computing Applications, pp. 11–18. IEEE (2010)

4.3 Artigo 3

Submetido ao INFOCOMP – Journal of Computer Science

ISSN 1807-4545

Publisher: Departamento de Ciência da Computação da Universidade Federal de Lavras

Country: Brazil

De acordo com o objetivo específico 3, este artigo teve como propósito a busca por *malwares* em redes de computadores com a Lógica Eτ, com base em uma abordagem não intrusiva e sistêmica.

Essa pesquisa se justifica devido ao fato de que a detecção de *malware* nas empresas é sempre um desafio devido ao grande número de dispositivos de rede, bem como a questão da privacidade quando o acesso direto ao equipamento é necessário. Os programas antivírus nem sempre são uma boa opção, considerando os custos de aquisição para versões efetivas. Uma possível abordagem é a busca de comportamentos anômalos na rede a partir de fontes de informação indiretas.

Como comportamentos anômalos, as situações de inconsistência podem ser entendidas em várias medidas de desempenho dos dispositivos analisados, em que muitas vezes a lógica clássica pode não ser a mais adequada para esse fim. Os vírus de computador, bem como as ferramentas de intrusão de rede, podem produzir comportamentos inesperados em uma rede de computadores devido ao tráfego de dados incomum em comparação com situações operacionais normais.

Uma possível abordagem em situações dessa natureza é a utilização de uma lógica não clássica. Entre as existentes, a Lógica Eτ tem sido utilizada em várias áreas por sua flexibilidade e facilidade, podendo determinar vários estados adicionais, além de verdadeiro ou falso, para lidar com situações de inconsistência.

Systemic and non-intrusive detection of malwares in computer networks with Logic Eτ

Avelino P. Pimenta Júnior, Jair M. Abe

Graduate Program in Production Engineering - Paulista University

1212,Dr. Bacelar Street, 04026-002 São Paulo, Brazil

Abstract. Malware detection in enterprises is always a challenge due to the large number of network devices, as well as the issue of privacy when direct access to the equipment is required. Antivirus programs are not always a good option, considering the acquisition costs for effective versions. A possible approach is the search for anomalous behaviors in the network from indirect information sources. As anomalous behaviors, one can understand those situations of inconsistency in various performance measures of the analyzed devices, in which, often, the Classical logic may not be the most suitable for this purpose. Computer viruses, as well as network intrusion tools, can produce unexpected behavior on a computer network by unusual data traffic compared to normal operating situations. A possible approach in situations of this nature is the use of a non-classical one. Among the existing ones, the Logic Eτ has stood out in several areas for its flexibility and easiness, being able to determine several additional states, other than true or false, to deal with situations of inconsistency. This survey looks for the location of malwares in computer networks with the Logic Eτ, by using a non-intrusive approach.

Keywords: computer networks; Logic Eτ; anomalous behavior; malware.

4.3.1 Introduction

4.3.1.1 Objectives

The purpose of this survey has beenthe search for situations considered anomalous, suggestive of malware infection, of devices connected to a computer network.

Once these situations that require attention were detected, it has been possible to apply the necessary corrections in the equipment involved. The exact type of malware is beyond the scope of this survey; however, it has been possible to generate meaningful indicators in search of the exact cause and determine where these problems occur.

4.3.1.2 Justification

The computer networks currently constitute the main form of transmitting data and services. Therefore, the task of monitoring the information has turned out to be a key factor in technology sectors [1] and is part of the backbone of information technology in various educational institutions and in companies of different natures. A computer network consists of several connected hosts, which can be represented by a desktop, a laptop, a smartphone, wearable devices, biomedical sensors, among others [2] [3]. In such a heterogeneous client environment, efficient content adaptation and delivery services are becoming a major requirement for the new internet service infrastructure [4].

By its very nature of decentralization and heterogeneity, it is often difficult to determine in a feasible time when and where a device fails. Different network links can operate at different speeds and can be implemented in various media, such as 1 Gbps or 100 Mbps, copper or optical fibers [5]. This situation may lead to a significant delay in correcting any problems, which, over the time, can become worse. Errors are often caused by anomalies, which can be understood as unexpected behaviors in a network [6] [7]. Often the discovery of one (or more) spot of failure occurs late, and its correction may be difficult, expensive, and impractical in some cases.

In critical systems, increasingly present in companies and institutions, data loss is not an acceptable option. Often, the lost data is no longer available for recovery, since many times the backuppolicies are poorly implemented or inexistent, even in great corporations. Often, these situations can lead to financial losses [8], and frequently the operating costs arising from data loss cannotbe estimated. Furthermore, with the growing number of World Wide Web users, the need for satisfactory performance becomes increasingly relevant [9].

Thus, early detection of failure spots within computer networks can certainly bring benefits and avoid significant losses in corporations.

4.3.2 Materials and methods

An analysis of an operation dayof a computer network has been held. The origin for this information can be gathered from different sources, obtained from network devices, such as routers or proxies. In this case, it has been acquired from

an open-source Linux based proxy Squid. The computer network under consideration is composed of approximately 90 hosts and comprises the administrative and academic network of a public education institution in Brazil. Their distribution and setup, as well as the network topology, are highly heterogeneous, and no longer follow the original project and requisites, meeting the current demands of the physical and logical changes that have taken place in the institution.

The data of the obtained attributes has been tabulated according to variable and parameterized time intervals. These may vary between 15 and 30 minutes. Initially, it has been observed that inferior values (less than 15 minutes) were often not enough to generate reliable results, considering that the volume of data generated in the analyzed network was still not sufficiently representative. On the other hand, superior values (more than 30 minutes) have rendered the search for errors less efficient in terms of speed and required higher computational processing.

The attributes were normalized for each host that composes the network, considering the analyzed range. Next, concepts of Paraconsistent Annotated Evidential Logic $E\tau$ have been applied for each of the involved equipment and have determined the favorable and contrary evidences for each of the attributes. With the aid of a data traffic analyzer application, it has been possible to obtain the network hosts behavior within a specific time interval. In the end, an overall analysis has been achieved, considering the various possible logical states contemplated by the Logic $E\tau$

4.3.3 Theory

As mentioned by [10]: “The amount of information that travels across the internet has increased dramatically in the past few decades because of the huge growth in the number of internet users”. Therefore, responsive service plays a critical role in determining end-user satisfaction. In fact, a customer who experiences a large delay after placing a request at a business’s web server often switches to a competitor who provides faster service [11]. Network infrastructure needs to be constantly improving to satisfy QoS (Quality of Service) users’ demand, including both technology aspects (e.g. fastest links, proxies, and servers) and related software [12].

To reach the user satisfaction issue, some important elements should be considered in data traffic management, such as reliability, integrity, and availability [5].

Reliability can be understood as the ability of the computer network to successfully transmit data from a specific source to a destination [13], and a performance index to evaluate the capability of a computer network [14].

Integrity is the ability to keep information intact, to ensure the quality of the information [5].

Availability is the ability to provide access to information systems as soon as they are requested [5].

Computer networks make use of routers, which are responsible for interconnecting two or more distinct networks. As mentioned by [15], “Routing is the process of sending data packets from the host of origin to the destination host, which is performed by the routers”. Generally, such devices are called gateways, which operate in the man-in-the-middle function. Many other features may be added to a router, such as access control, firewall, bandwidth managers.

For the establishmentof network communication, there must always be a request from the side of the "client". It is a typical protocol of request-response, which controls the data transfer between server and client (such as a web browser) [16].

This request, when answered by the side of the "server", typically a proxy, producesa corresponding response. Proxy servers are designed with three goals: decrease network traffic, reduce user (client) perceived lag, and reduce loads on the origin servers [17]. Every request from the client passes through the proxy server, which, in turn, may or may not modify the client’s request based on its implementation mechanism [18].

Some elements may be interesting to the packet traffic analysis, such as the logical address associated with the request for the resource, request time, response time, type of result obtained, the amount of response data in the transaction, and destination request [19]. As the gateway or proxy forwards packets to other networks, it is possible to audit data traffic information. In this survey, this information has been called *attributes*.

There are several analyzable attributes with different levels of importance. In this case, the following have been considered:

a) Trafficked Data (D)

b) Response time (RT)

c) Requisitions (R)

d) Errors (E)

When a resource is requested, the agent responsible for locating and searching the internet is the gateway, which can be represented by an ordinary router or a more sophisticated device, such as a proxy. When receiving this information, it can be registered in logs, which are plain text files that record each of the performed operations with their attribute values.

An example of a log used by the developed program is generated by the open source Squid proxy/cache software, originally developed for Linux operating systems, widely used in internet providers.

The first attribute (trafficked data) corresponds to the volume of information that has been requested by a given network device in each period. This attribute is measured in bytes, obtained from the response to a requested resource, which may comprise a Web page, a file, a stream of data, among others.

The second attribute (response time) corresponds to the total response time spent to obtain the resources requested in each period. This attribute is measured in milliseconds. In this case, it is also considered the time of not located resources, or that had an error in response since these two situations should also be considered for analysis purposes.

The third attribute (requisitions) corresponds to the number of requests made by a network device in each period. The higher the number of requisitions, the more active the equipment may be considered.

The fourth and last attribute (errors) corresponds to the number of zero-byte responses received after a request is made, which can be translated as an error or problem in the location of a requested resource. When searching for a certain piece of information on the internet, even if it is not available, a zero-size response is sent to the local network, informing that the operation was not successful. This is an undesirable situation, since processing and time resources have been spent without a favorable counterpart to the request made.

One of the problems in monitoring computer networks is the issue of high levels of uncertainty and unpredictability in their operation.

As mentioned before, several other attributes could be used, but, for this survey, these ones have been considered the most significant.

The concepts of Logic $E\tau$ will be used from this point on. According to [20]: "The atomic formulas of the Logic $E\tau$ are the type $p(\mu, \lambda)$, where $(\mu, \lambda) \in [0, 1]^2$ ([0, 1] is the real unit interval) and p denotes a propositional variable". Therefore, among several readings, $p(\mu, \lambda)$ can be intuitively read: "It is assumed that the favorable evidence of p is μ and the contrary evidence of p is λ". Thus, we have, for instance, the following particular readings:

a) $p_{(1.0, 0.0)}$ can be read as a true proposition;

b) $p_{(0.0, 1.0)}$ as false;

c) $p_{(1.0, 1.0)}$ as inconsistent;

d) $p_{(0.0, 0.0)}$ as paracomplete; and

e) $p_{(0.5, 0.5)}$ as an indefinite proposition.

The uncertainty and certainty degrees associated to (μ, λ) are defined [21] [22]:

a) Uncertainty Degree: $G_{un}(\mu, \lambda) = \mu + \lambda - 1$ $(0 \leq \mu, \lambda \leq 1)$;

b) Certainty Degree: $G_{ce}(\mu, \lambda) = \mu - \lambda$ $(0 \leq \mu, \lambda \leq 1)$;

An order relation is defined on $[0, 1]^2$: $(\mu_1, \lambda_1) \leq (\mu_2\ \lambda_2) \Leftrightarrow \mu_1 \leq \mu_2$ and $\lambda_2 \leq \lambda_1$, forming a lattice which is symbolized by τ.

With the degree of certainty and uncertainty, one can determine the following 12 output states, shown in Table 2:

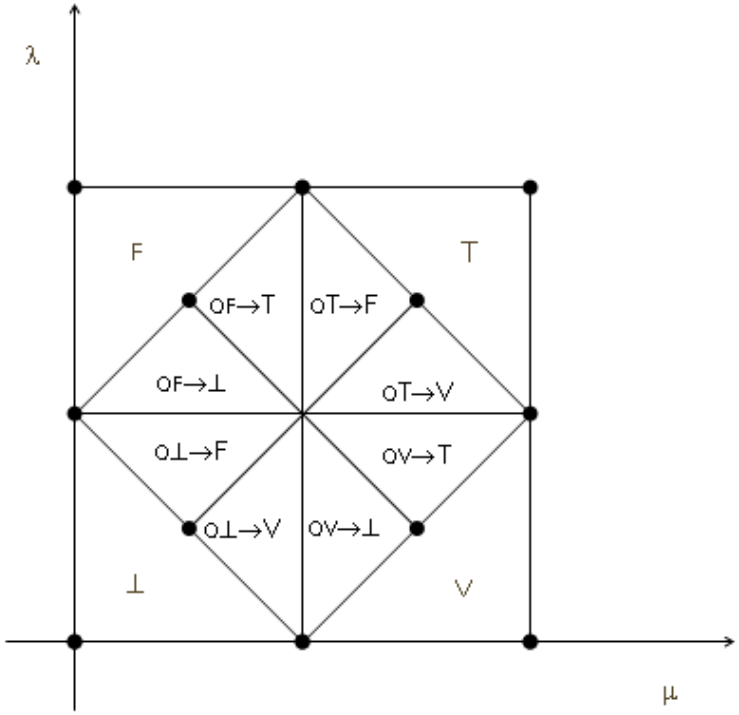
Table 2 – Extreme and non-extreme states

Extreme states	**Symbol**	**Non-extreme states**	**Symbol**
True	V	Quasi-true tending to Inconsistent	$QV \rightarrow T$
False	F	Quasi-true tending to Paracomplete	$QV \rightarrow \perp$
Inconsistent	T	Quasi-false tending to Inconsistent	$QF \rightarrow T$
Paracomplete	$\perp$	Quasi-false tending to Paracomplete	$QF \rightarrow \perp$
		Quasi-inconsistent tending to True	$QT \rightarrow V$
		Quasi-inconsistent tending to False	$QT \rightarrow F$
		Quasi-paracomplete tending to True	$Q\perp \rightarrow V$
		Quasi-paracomplete tending to False	$Q\perp \rightarrow F$

Source: Abe (2015).

Extreme and non-extreme states are shownin Figure 10:

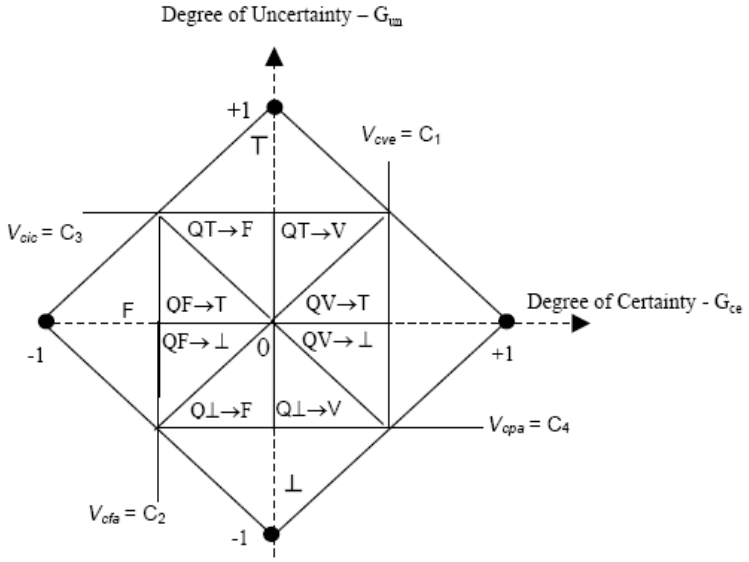
Figure 10 – Extreme and non-extreme states of the Lattice τ



Source: Abe (2015).

Also, in Figure 11, the states are shown together, with certainty and uncertainty degrees, as well as the control values.

Figure 11 – Certainty / Uncertainty degrees with decision states of the Lattice τ



Source: Abe (2015).

4.3.4 Calculation

For the development of the software for detecting anomalous behavior in computer networks, it has been necessary to transform the data obtained from the Squid application logs, which were originally formatted in plain text files, into a relational database system.

The location of the file that represents the accesses made to a Proxy server may vary, but, in general, they are located in the /var/log/squid/log directory of a GNU/Linux system. The file, access.log, generally consists of (at least) 10 columns separated by one or more spaces, whose sequential formatting of the fields in the text file follows this pattern [23]:

Time - Elapsed – Client Address- Code/Status Bytes -Method - URL - RFC931 –Peer status Type

In detail, each field represents the following information:

a) **Time:** a Unix timestamp as UTC seconds with a millisecond resolution. This is the time when Squid started to log the transaction;

b) **Elapsed:** the elapsed time considers for how many milliseconds the transaction busied the cache;

c) **Client address:** the IP address of the requesting instance;

d) **Code/Status:** this column is made up of two entries separated by a slash. This column encodes the transaction result. The cache result of the request contains information on the kind of request, how it has been satisfied, or in what way it failed;

e) **Bytes:** the size of the amount of data delivered to the client;

f) **Method:** the request method to obtain an object;

g) **URL:** this column contains the URL requested;

h) **RFC931:** the user identity for the requesting client;

i) **Peer status:** the hierarchy information consists of three items:

- Any hierarchy tag may be prefixed with *TIMEOUT_* if the timeout occurs waiting for all ICP replies to return from the neighbors. The timeout is either dynamic, if the icp_query_timeout has not been set, or the time configured there has run up;

- A code that explains how the request has been handled, e.g., by forwarding it to a peer, or going straight to the source;
- The IP address or hostname from where the request (if a miss) has been forwarded. For requests sent to origin servers, this is the origin server's IP address. For requests sent to a neighbor cache, this is the neighbor's hostname.

j) Type: the content type of the object as seen in the HTTP reply header.

Considering the development of a Java based data traffic analyzer application, it has been necessary to perform the conversion of the data from a plain text file to a relational database. Each of the log files varies in size, ranging from 60000 to 150000 lines, each representing an external resource request sent to the server by a given network host.

The first step in obtaining the data has been converting each of the tabulated fields from the text file to the CSV (comma-separated values) format, which can be done with usual spreadsheet applications. From the newly obtained file, it has been possible to generate the relational database based on the MySQL Community Server.

Considering the representative fields and the performed data conversion, it was possible to create the table from the following command line:

```
CREATE TABLE `day_table` (
 `col_day_week` char(5) DEFAULT NULL,
 `col_month` char(5) DEFAULT NULL,
 `col_day` int(11) DEFAULT NULL,
 `col_hour` char(10) DEFAULT NULL,
 `col_year` int(11) DEFAULT NULL,
 `col_response` int(11) DEFAULT NULL,
 `col_origin` char(20) DEFAULT NULL,
 `col_type` char(25) DEFAULT NULL,
 `col_size` int(11) DEFAULT NULL)
)
```

It has not been necessary to use any external tools to import the data from the CSV file, whose data loading was possible from the instruction:

```
LOAD DATA INFILE day_table.csv' INTO TABLE day_table FIELDS TERMINATED BY ',' IGNORE 1 LINES;
```

The Hibernate framework has been used to perform object-relational mapping (ORM), in which the main objective is to reduce the complexity involved in the development of applications that need to interact with relational databases [24]. In this case, the database is converted into objects and can be accessed without the need of explicit SQL calls, thus making native calls. The developed application carries out the monitoring in two distinct stages, defined as follows:

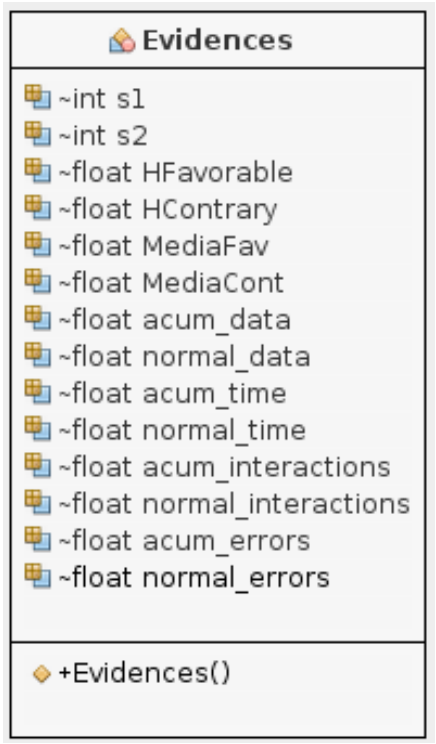
a) Systemic evaluation and detection of critical intervals;
b) Specific detection of network anomalies at critical intervals.

4.3.4.1 Systemic evaluation and detection of critical intervals

Considering that it is not known at what time a host can exhibit a behavior that is not compatible with normality patterns, the first step is to perform a more comprehensive search of the network operation. Random variations have been tested between 10 and 60 minutes, and it has been observed that the results obtained up to the limit of 30 minutes had little significant variation. It has also been noted that smaller values produced a high number of interactions, which would be associated with a worse and undesired computational performance of the system. On the other hand, higher values could mathematically dilute situations of abnormality of certain equipment, making it difficult to detect them.

At this stage, there is still no hint of the hosts that may be the potential problem generators, since the emphasis is on the detection of one or more critical intervals in which anomalous behaviors may be occurring. For this, an ArrayList of the class Evidences has been generated, as shown in Figure 12:

Figure 12 – UML representation of the Evidence class



Source: Author.

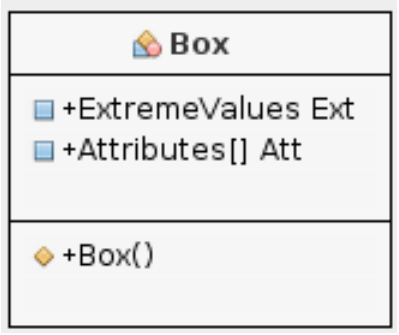
As it can be verified, the Evidence class has both the fields of the accumulated values of each network attribute, as well as their normalized values. It is a **publicEvidencesget_Evidences (String start, String end)** method, in which all attributes are accounted by accumulation or counting. Inactive hosts are not considered, since their attributes without values, if computed, would end up undesirably influencing the process of normalization of values.

Initially, the **public List<Evidences>get_Total (String start, String end, int interval)** method is invoked, which is capable of generating an ArrayList of the Evidences class and should invoke the previous method within the time interval. Thus, it is possible to obtain the absolute values of the attributes under analysis. The next step is to normalize the values from the generated ArrayList, with the invocation of the method **public void NormalizeList_Intervals (List<Evidences> L)**, which may, with the Evidences class, determine the Favorable and Contrary Evidences, as well as the Certainty and Uncertainty Degrees of the various time intervals. As a way of a faster search, a possible and interesting approach may consider one or more intervals with extreme values according to the concepts of Logic $E\tau$. With the result of the interval, the next step is the specific search for the anomalous behavior host.

4.3.4.2 Specific detection of network anomalies at critical intervals.

The first step consists of generating an ArrayList of the class named Box, as represented in Figure 13:

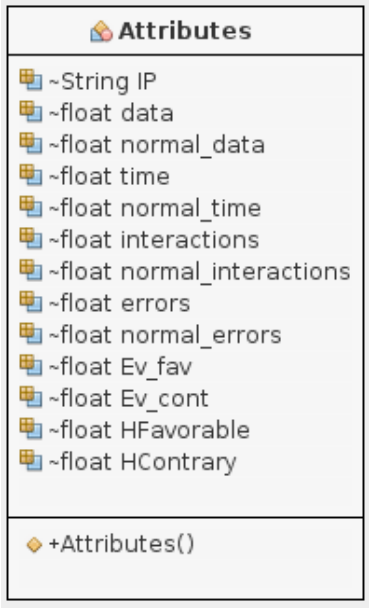
Figure 13 – UML representation of the Box class



Source: Author.

The class Box is composed by two other classes: the first, named Attributes, consists of the network attributes obtained from the network logs, represented by Figure 14:

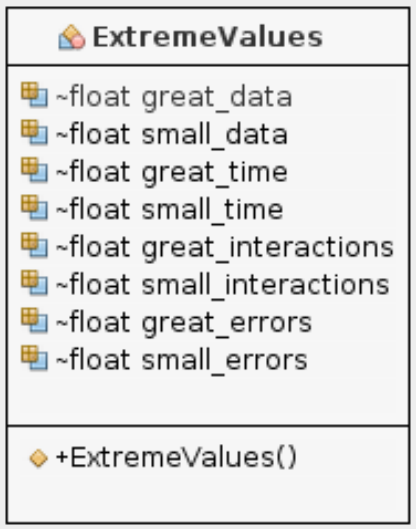
Figure 14 – UML representation of the Attributes class



Source: Author.

The second class, ExtremeValues, aims to aid in the process of normalizing the network attributes and also stores their minimum and maximum values. It is represented by Figure 15:

Figure 15 – UML representation of the ExtremeValues class



Source: Author.

The method **publicBoxget_Attributes (String start, String end)** has been invoked, in which all network attributes have been obtained for each of the network hosts, comprising the intervals determined by the parameters “start” and “end”. Again, inactive hosts whose attributes without values, if computed, would undesirably inflate in the process of normalization of values, have not been considered.

Then, the method **public List<Box>get_MinMax (String start, String end, int interval)** has been invoked, which is responsible for generating the ArrayLists of the Box class for each of the intervals, and also the extreme values (minimum and maximum) of the network attributes, represented in the ExtremeValues class, which has later been used for future normalization.

Then, it has been possible to normalize the network attributes, considering that an ArrayList of the Box class has been generated, comprising the operation data of all the network hosts in several intervals, as well as determining the extreme values of each attribute. The next step has been the invocation of the **public void NormalizeList_IPs(List<Box> L)** method, which, in addition to normalizing, has also been able to determine the Favorable and Contrary Evidences, as well as the Certainty and Uncertainty Degrees, from the Logic $E\tau$.

This part of the process should be repeated as many times as necessary, with a gradual decrease of the search intervals, until the expert can verify a clear and undoubted scenario of his search, that is, the host with anomalous behavior.

On the day the analysis had been carried out, the time interval considered varied from 8:00 to 23:00, which included the period in which the computer labs were

available to the general public, students, professors and the administrative staff. There is no defined standard use of the laboratory resources, since they depend on the classes that will be taught, in the specific case of the academic members, as well as the utilization of the administrative area, which depends on the month period in which a specific task is most often. Much bureaucratic work developed in the institution already makes use of automated systems, considering that it is part of a conglomerate with more than seventy other units.

Therefore, there is no clear operation pattern of the computer network, and thus the predictability and ease of detection of any problems or errors are compromised. This difficulty tends to worsen even more, considering the large number of network devices, since any problems that may occur in a small number of devices may be hidden by the others. The computer networks tend to be resilient, that is, they can keep functioning even if errors occur. The decentralized physical distribution also contributes to the difficulty in detecting any problems. Therefore, the access to centralized information is one of the paths that have been used to locate the problems.

4.3.4.3 Search for anomalous behavior in time intervals

The first step is to determine the favorable and contrary evidences of the attributes in the considered time intervals. This range is parameterizable, and, therefore, it can be adjusted according to the presented scenario. As an example, in situations where data traffic is considered low, it may be worthwhile to increase the time interval to be analyzed in order to obtain a more significant sampling of the object to be analyzed. If data traffic is heavy, this range may be decreased so as not to compromise the analyzer’s performance.

The scenario to be analyzed uses IPv4 Class C addresses, and at the time the network was the object of study, the original project was no longer being followed. Therefore, the allocation of the addresses is set up randomly to the requesting devices. However, only active hosts have been considered to calculate the evidences.

The analysis of the attributes has been made from 8:00 to 8:29, 8:30 to 9:00, and subsequently until 23:00, for determining the favorable and opposite evidences. Thus, several scenarios of network traffic could be evaluated.

The first step was to totalize the analyzed attributes by the interval, shown in Table 3:

Table 3 – Absolute values of the analyzed attributes obtained from network logs

Time interval	Data Trafficked (bytes)	Response Time (ms)	Requisition (un)	Errors (un)
8:00:00 - 8:29:59	9.9093632 E7	4.4279896E7	3281.0	13.0
8:30:00 - 8:59:59	1.60600704E8	8.4619144E7	4459.0	22.0
9:00:00 - 9:29:59	2.1030584E8	7.6781648E7	1857.0	30.0
9:30:00 - 9:59:59	8.5182512E7	6.1862668E7	2957.0	20.0
10:00:00 - 10:29:59	2.51689872E8	8.2077576E7	4756.0	16.0
10:30:00 - 10:59:59	8.3945624E7	5.8526308E7	3516.0	25.0
11:00:00 - 11:29:59	5.8782236E7	6.5617692E7	2892.0	10.0
11:30:00 - 11:59:59	6.42468E7	6.2325128E7	3746.0	28.0
12:00:00 - 12:29:59	1.16160048E8	5.7763048E7	5515.0	34.0
12:30:00 - 12:59:59	2.81489696E8	6.598046E7	4197.0	39.0
13:00:00 - 13:29:59	7.557472E7	5.226914E7	4708.0	29.0
13:30:00 - 13:59:59	1.04744872E8	6.3228104E7	6350.0	40.0
14:00:00 - 14: 29:59	2.27949008E8	1.3698472E8	8186.0	68.0
14:30:00 - 14:59:59	9.3282032E7	9.4350152E7	6643.0	2500.0
15:00:00 - 15:29:59	1.25987664E8	9.7167072E7	13203.0	6874.0
15:30:00 - 15:59:59	2.73367616E8	1.05816296E8	9099.0	1519.0
16:00:00 - 16:29:59	4.11387136E8	1.89746144E8	17008.0	6752.0
16:30:00 - 16:59:59	2.65305024 E8	1.23623696E8	21968.0	14575.0
17:00:00 - 17:29:59	2.19906656E8	1.15870672E8	12279.0	8820.0
17:30:00 - 17:59:59	2.93159488E8	7.8449592E7	6620.0	2809.0
18:00:00 - 18:29:59	9.3969776E7	8.4985488E7	2006.0	16.0
18:30:00 - 18:59:59	3.9746752E7	6.5820184E7	5360.0	11.0
19:00:00 - 19:29:59	3.7131136E7	6.9051264E7	3462.0	4.0
19:30:00 - 19:59:59	6.9230912E7	1.23596128E8	2040.0	29.0
20:00:00 - 20:29:59	1.1345296E8	1.1629288E8	3089.0	19.0
20:30:00 - 20:59:59	9.8718096E7	1.06047968E8	2510.0	41.0
21:00:00 - 21:29: 59	5.4427928E7	9.1542016E7	6974.0	4559.0
21:30:00 - 21:59:59	8.2702984E7	8.2027728E7	10871.0	8167.0
22:00:00 - 22:29:59	5.2289248E7	8.289784E7	5774.0	2363.0
22:30:00 - 22:59:59	1.03395056E8	1.30959304E8	14509.0	10060.0

Source: Author.

For each of the attributes, the normalization process of the values in the intervals between 0 and 1has been applied. This process is necessary for the determination of the favorable and opposite evidences of the attributes.

The highest (V*max)* and lowest (V*min)* values for each column of the involved variables have been determined. Then, for each V value, the formula has been applied:

$$Vn = (V - Vmin) / (Vmax - Vmin)$$

After the normalization process, the following modified values were determined, as shown in Table 4:

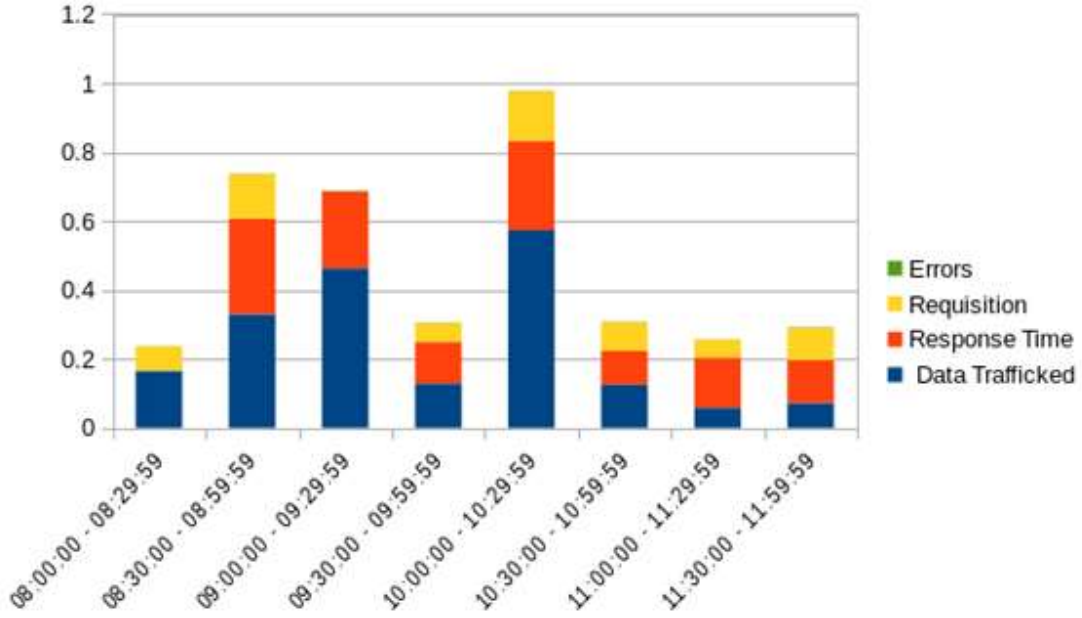
Table 4 – Normalized values of the analyzed attributes obtained from network logs

Time interval	Normalized Data Trafficked (bytes)	Normalized Response Time (ms)	Normalized Requisition (un)	Normalized Errors (un)
08:00:00 - 08:29:59	0.1655618	0.0	0.07080702	6.1766524E-4
08:30:00 - 08:59:59	0.3299067	0.27731004	0.12938192	0.0012353305
09:00:00 - 09:29:59	0.46271724	0.22343159	0.0	0.0017843662
09:30:00 - 09:59:59	0.12839173	0.12087184	0.054696433	0.0010980715
10:00:00 - 10:29:59	0.57329404	0.25983816	0.14414997	8.2355365E-4
10:30:00 - 10:59:59	0.12508681	0.097936206	0.082492165	0.0014412189
11:00:00 - 11:29:59	0.057851043	0.14668556	0.05146437	4.1177683E-4
11:30:00 - 11:59:59	0.07245218	0.124051	0.093928695	0.0016471073
12:00:00 - 12:29:59	0.21116272	0.092689216	0.1818905	0.0020588841
12:30:00 - 12:59:59	0.6529182	0.14917938	0.116354235	0.0024020313
13:00:00 - 13:29:59	0.10272002	0.05492164	0.14176321	0.0017157367
13:30:00 - 13:59:59	0.18066172	0.13025846	0.22341007	0.002470661
14:00:00 - 14:29:59	0.5098592	0.6372944	0.3147034	0.004392286
14:30:00 - 14:59:59	0.15003338	0.34420535	0.23797922	0.17129916
15:00:00 - 15:29:59	0.23742178	0.3635701	0.5641689	0.47148445
15:30:00 - 15:59:59	0.6312163	0.42302874	0.36010143	0.10397365
16:00:00 - 16:29:59	1.0	1.0	0.7533688	0.46311167
16:30:00 - 16:59:59	0.6096733	0.5454448	1.0	1.0
17:00:00 - 17:29:59	0.4883703	0.49214703	0.5182239	0.6050374
17:30:00 - 17:59:59	0.68409956	0.23489778	0.23683557	0.19250566
18:00:00 - 18:29:59	0.15187103	0.27982846	0.0074088806	8.2355365E-4
18:30:00 - 18:59:59	0.006988842	0.14807758	0.17418328	4.8040628E-4
19:00:00 - 19:29:59	0.0	0.17028946	0.07980707	0.0
19:30:00 - 19:59:59	0.08576957	0.54525524	0.009099497	0.0017157367
20:00:00 - 20:29:59	0.20392945	0.49504948	0.061260007	0.0010294421
20:30:00 - 20:59:59	0.16455838	0.42462134	0.032469794	0.0025392903
21:00:00 - 21:29:59	0.046216473	0.32490095	0.25443786	0.31260723
21:30:00 - 21:59:59	0.121766515	0.25949547	0.44821241	0.5602224
22:00:00 - 22:29:59	0.04050199	0.26547703	0.19476902	0.16189691
22:30:00 - 22:59:59	0.17705506	0.595873	0.6291084	0.6901379

Source: Author.

As a way of presenting the values in Table 4 in perspective, it is possible to create three stacked bar graphs that represent the three periods of the network operating day. In Figure 16, it can be observed that the largest sum of all the normalized attributes is less than 1. Although it is not possible, with the data of this period, to point out any abnormality in the network, considering that the values are still insufficient and not representative, when it is noted that the “Error” attribute could not even be quantified, the probability of functioning within normality patterns is higher.

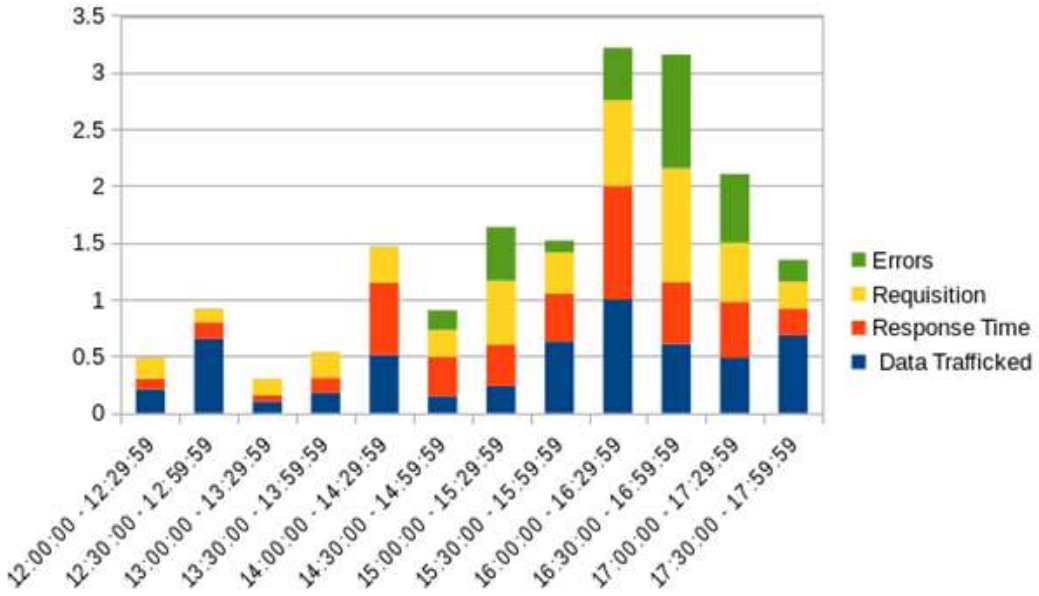
Figure 16 – Bar graph representing the network operating from 08:00 – 11:59



Source: Author.

Figure 17 performs a similar behavior to Figure 16's until approximately 16:00, when the sum of the normalized attributes abruptly exceeds the value 3, and this behavior prevailed until 16:59. This change particularly draws attention, since usually, in this period, there are few connected users, and, therefore, little use of the network capacity. In addition, the intervals from 16:00 to 16:29 and 16:30 to 16:59 have significant error rates, which may require a more careful analysis in the search for anomalous events.

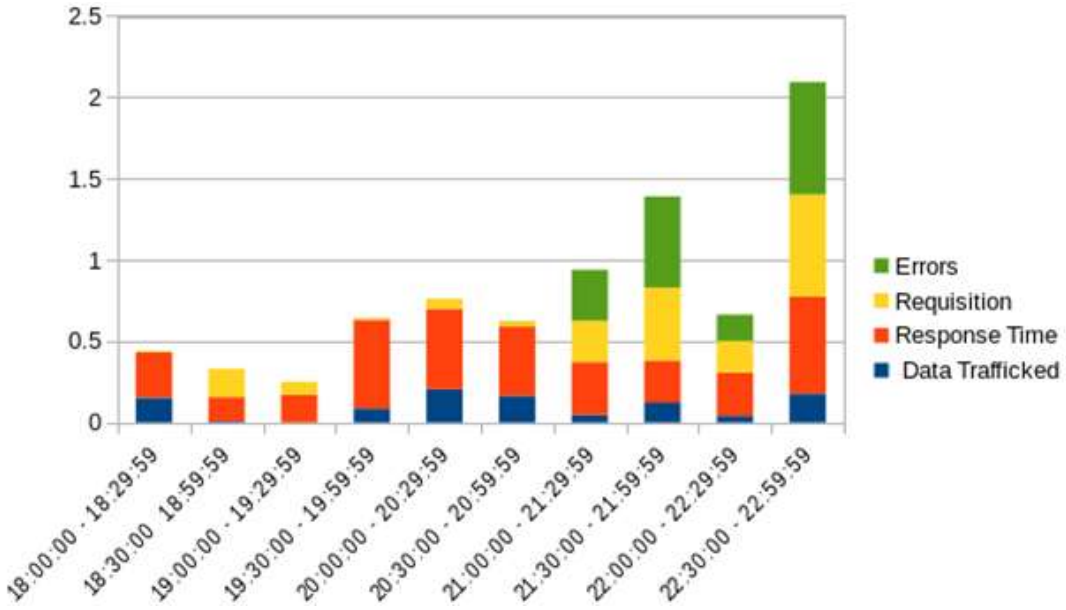
Figure 17 – Bar graph representing the network operating from 12:00 – 17:59



Source: Author.

In Figure 18, which covers the period with the greatest use of the network, it is possible to observe that, except for the interval from 22:30 – 22:59, the sum of the normalized attributes does not even exceed the value of 1.5. Although the last interval presents a sum that slightly exceeds the value 2, it is still lower than those observed in Figure 17.

Figure 18 – Bar graph representing the network operating from 18:00 – 22:59



Source: Author.

Considering that the search for anomalies may be simpler if compared to extreme situations, a good choice to start tracing them would be the intervals indicated in Figure 17.

Among the analyzed attributes, it is desirable a computer network that is capable of forwarding a significant amount of data satisfactorily, attending the user requests. Therefore, the attributes Trafficked Data (D) and Requisitions (R) can be considered favorable evidences.

On the other hand, it is important that the response time be as low as possible, in order to obtain higher system usability and user satisfaction. A small number of errors is also desired, which implies fewer retransmissions. High values of the two attributes are not desirable. Therefore, the attributes Response Time (RT) and Errors (E) can be considered contrary evidences.

For each analyzed interval, the favorable (μ) and contrary (λ) evidences have been determined, considering the mentioned attributes, with the following formulas:

$$\mu = (w^1D + w^2R) / (w^1+w^2)$$
$$\lambda = (w^3RT + w^4E) / (w^3+w^4)$$

For the first analysis, the values of the weights (w^n) of each of the attributes have been considered equal, since there is no initial condition to determine higher or lower levels of importance of one attribute in comparison to the other. However, calibrations can be performed according to the Certainty degree G_{ce} (μ, λ) and Uncertainty degree G_{un} (μ, λ) obtained from the analyzed scenarios.

Therefore, it is possible, with the aid of the Logic Eτ, to present the following results for the time intervals in the analyzed network, according to the Table 5:

Table 5 – Favorable/Contrary Evidence and Certainty/Uncertainty Degrees for each interval

Time interval	Favorable Evidence μ	Contrary Evidence λ	Certainty Degree G_{ce}	Uncertainty Degree G_{un}
08:00:00 - 08:29:59	0.1181844	3.0883262E-4	0.11787557	-0.88150674
08:30:00 - 08:59:59	0.22964431	0.13927269	0.09037162	-0.631083
09:00:00 - 09:29:59	0.23135862	0.11260798	0.11875064	-0.6560334
09:30:00 - 09:59:59	0.09154408	0.060984958	0.030559119	-0.847471
10:00:00 - 10:29:59	0.358722	0.13033086	0.22839114	-0.5109471
10:30:00 - 10:59:59	0.10378949	0.04968871	0.05410078	-0.8465218
11:00:00 - 11:29:59	0.054657705	0.07354867	-0.018890962	-0.8717936
11:30:00 - 11:59:59	0.08319044	0.06284905	0.020341389	-0.8539605
12:00:00 - 12:29:59	0.19652662	0.04737405	0.14915256	-0.75609934
12:30:00 - 12:59:59	0.38463622	0.07579071	0.30884552	-0.5395731
13:00:00 - 13:29:59	0.122241616	0.028318688	0.09392293	-0.8494397
13:30:00 - 13:59:59	0.2020359	0.06636456	0.13567135	-0.73159957
14:00:00 - 14:29:59	0.4122813	0.32084334	0.091437966	-0.2668754
14:30:00 - 14:59:59	0.1940063	0.25775224	-0.063745946	0.5482415
15:00:00 - 15:29:59	0.40079534	0.41752726	-0.016731918	-0.1816774
15:30:00 - 15:59:59	0.49565887	0.2635012	0.23215768	-0.24083996
16:00:00 - 16:29:59	0.8766844	0.7315558	0.14512861	0.60824025
16:30:00 - 16:59:59	0.80483663	0.77272236	0.032114267	0.577559
17:00:00 - 17:29:59	0.5032971	0.5485922	-0.04529512	0.0518893
17:30:00 - 17:59:59	0.46046758	0.21370173	0.24676585	-0.3258307
18:00:00 - 18:29:59	0.07963996	0.14032601	-0.06068605	-0.78003407
18:30:00 - 18:59:59	0.09058606	0.074278995	0.016307063	-0.835135
19:00:00 - 19:29:59	0.039903536	0.08514473	-0.045241192	-0.8749517
19:30:00 - 19:59:59	0.047434535	0.27348548	-0.22605094	-0.67908
20:00:00 - 20:29:59	0.13259473	0.24803945	-0.11544472	-0.6193658
20:30:00 - 20:59:59	0.09851409	0.21358031	-0.11506622	-0.6879056
21:00:00 - 21:29:59	0.15032718	0.31875408	-0.1684269	-0.5309187
21:30:00 - 21:59:59	0.28498948	0.40985894	-0.124869466	-0.30515158
22:00:00 - 22:29:59	0.1176355	0.21368697	-0.09605147	-0.6686775

Source: Author.

The next step, based on the Certainty and Uncertainty degrees, has been the search for one of the time intervals that may represent a significantly anomalous network operation. Among the candidate intervals for analysis, there was a high degree of uncertainty between 16:00 and 16:29. In fact, the G_{ce} has been calculated at 0.14512861 and the G_{un} at 0.60824025, revealing a significant inconsistency profile.

From the “divide-to-conquer” strategy, a new analysis has been held, considering only the interval between 16:00 and 16:29, but with a 10-minute variation. The following values have been obtained with the respective G_{ce} and G_{un}, according to the Table 6:

Table 6 – Certainty / Uncertainty Degrees of the attributes from 16:00 to 16:30

Time interval	Certainty Degree G_{ce}	Uncertainty Degree G_{un}
16:00:00 - 16:09:59	-0.48813787	-0.48813784
16:10:00 - 16:19:59	0.32670146	-0.67150617
16:20:00 - 16:29:59	0.4484431	0.5515568

Source: Author.

With the new analysis carried out, another source of inconsistency has been obtained from the interval between 16:20 and 16:29 hours. Once again, a new analysis has been performed, with another time reduction: 2 minutes, as shown in Table 7:

Table 7 – Certainty / Uncertainty Degrees of the attributes from 16:20 to 16:30

Time interval	Certainty degree G_{ce}	Uncertainty Degree G_{un}
16:20:00 - 16:21:59	0.23057377	0.2281071
16:22:00 - 16:23:59	0.5198167	-0.39736778
16:24:00 - 16:25:59	-0.49263892	-0.1695013
16:26:00 - 16:27:59	0.0058194995	0.3685069
16:28:00 - 16:29:59	0.08172488	-0.9182751

Source: Author.

Among the intervals, an inconsistency between 16:20 and 16:22 has been observed. A final analysis, with a 1-minute variation is shown in Table 8:

Table 8 – Certainty / Uncertainty Degrees of the attributes from 16:26 to 16:28

Time interval	Certainty Degree G_{ce}	Uncertainty Degree G_{un}
16:26:00 - 16:26:59	0	-1
16: 27:00 - 16:27:59	0	1

Source: Author.

Among the two intervals, the one that presented the highest level of inconsistency has been the one between 16:27 and 16:28. Thus, a specific interval, with a 1-minute variation, has been determined, in which the potentially problem-generating equipment in the network can be tracked. From a 15-hour operating scenario, with the application of the Logic Eτ, it has been possible to reduce the scope of the analysis to only 1 minute.

4.3.4.4 Search for anomalous behavior of the network hosts

Based on the time interval, between 16:27 and 16:28, a specific search has been held to locate one or more hosts that might be responsible for the anomalous behavior in the computer network. Each of the operating network hosts within the defined time interval, and identified by its source IP address, had its respective favorable and contrary evidences calculated for the attributes. The following results have been obtained, according to the Table 9:

Table 9 – Certainty / Uncertainty Degrees of the hosts from 16:27 and 16:28

IP host address	Certainty degree G_{ce}	Uncertainty Degree G_{un}
192.168.1.6	0.0712372	-0.881337
192.168.1.15	-0.022222161	0.07037747
192.168.1.17	-0.05768591	-0.9401534
192.168.1.38	-0.12034002	-0.87494224
192.168.1.65	-0.082016654	-0.91380996
192.168.1.74	0.049036026	0.049036026
192.168.1.79	-0.01788491	-0.97344714
192.168.1.82	-0.010851777	-0.97934717
192.168.1.90	-0.08030032	-0.77098423
192.168.1.104	0.058846354	-0.79822236

Source: Author.

Ten candidates have been determined from 254 possible sources. Among these candidates, one that represents the most evident anomalous behavioris the IP address host 192.168.1.74.

4.3.5 Results and Discussion

From the obtained result, a specific host analysis has been carried out, in which the level of inconsistency has been continuously maintained throughout the analysis. It is important to point out that other devices could also have presented problems of inconsistency simultaneously, also becoming candidates for analyzes of this nature. However, specifically, in this case, only one host presented this type of problem, and it underwent a more accurate verification.

For this purpose, a quantitative analysis of the host attributes has been performed on the same day. It has been determined that it was a connected device in the administrative sector of the institution. This host had operated on that day only in the interval between 13:30 and 16:30. It was not a backbone device, such as a server, but a desktop used by trainees in the administrative sector. Further analysis revealed that the computer had been infected with various types of malware.

Considering the interval, the favorable and contrary evidenceshave been calculated, according to the following Table 10:

Table 10 – Normalized attributes values of the host from 13:30 to 16:30

Time interval	Data	Interactions	Response Time	Errors
13:30:00 - 13:59:59	0.113430575	0.014455948	0.19705948	0.0053908355
14:00:00 - 14: 29:59	1	0.120868206	0.5505165	0.0062621143
14:30:00 - 14:59:59	1	0.01741149	0.92535114	0.0008268449
15:00:00 - 15:29:59	1	0.023295393	1	0.0010223787
15:30:00 - 15:59:59	0.035427984	0.04673564	0.65221035	0.0048908954
16:00:00 - 16:29:59	0.98902106	0.11062907	1	0.12903225

Source: Author.

From the device operating information, although it occurred within only three hours, it has been possible to observe a significant anomalous behavior.

In the period from 14:00 to 14:29, from 14:30 to 14:59, and from 15:00 to 15:29, the host has been the largest user of the network band, although the number of requests was considered small, presenting an inconsistent behavior. In fact, considering the three intervals, it has been observed that the last two responded by only 1% and 2%, respectively, of the interactions, which, by itself, already makes the device behavior, at least, unexpected.

In all three intervals, the error rate remained low, which shows that the network was being able to respond to requests. In the first interval, the response time corresponds to only 55% of that observed in the interval, and immediately rises to 92% and 100% for the last intervals, causing the search for network resources to become significantly slower.

From 15:30 to 15:59, there was a sudden drop in data traffic, having fallen to approximately 3.5% of the total. Even so, the equipment responded for 65% of the response time, an unexpected behavior since the error rate remained low.

Previously, it had already been determined that the interval between 16:00 and 16:29 comprised the one with the worst performance in the network, and this was confirmed by the analysis of the attributes. In this specific range, the network practically reached the highest utilization of the network (98% of the traffic data), considering only 11% of the requests made. The response time was the highest among the other hosts, and, in this interval, the error rate increased to 12%, significantly higher than the others, ranging from 0% to 0.4%.

4.3.6 Conclusion

From the study, it has been possible not only to determine which host presented a contradictory and unexpected behavior about the expected network parameters, but also the exact moment in which this occurred, within a 1-minute interval. Considering that the network operates from 8:00 to 23:00, one can consider as a success the reduction of 15 hours of analysis to only 1 minute. This reduction would be even more evident in scenarios with greater number of connected devices, where the prospecting of errors would be even more complex.

In any case, it has been possible to determine that the reason for the malfunction of the host did not go through problems such as congestion or network failure. In fact, it has been possible to observe the so-called "misuse" of the equipment by the team of trainees. It has also been possible to verify that the equipment, originally installed for the exclusive use of the administrative sector, was full of unauthorized applications, being heavily used for access to not reliable internet addresses, which unfortunately brought several of the so-called malicious software into the system. These, on an autonomous basis, sent and received data to the

internet, contributing significantly to the emergence of security breaches in the analyzed network.

It is important to emphasize the fundamental role of Logic $E\tau$ in the search and location of the problem, considering that Classical Logic, in general, would only give us the possibility to determine if the host was or was not operating expectedly, disregarding possible contradictions.

It is also important to emphasize that the search was not for malfunctioning devices due to physical problems, but rather to anomalous situations generated by computers operating within normality patterns and without apparent failures.

4.3.7 References

[1] Y. K. Lin and C. F. Huang, “Stochastic computer network under accuracy rate constraint from QoS viewpoint,” *Inf. Sci. (Ny).*, vol. 239, pp. 241–252, 2013.

[2] L. Da Xu, W. He, and S. Li, “internet of Things in Industries: A Survey,” *IEEE Trans. Ind. Informatics*, vol. 10, no. 4, pp. 2233–2243, Nov. 2014.

[3] Zhuming Bi, Li Da Xu, and Chengen Wang, “internet of Things for Enterprise Systems of Modern Manufacturing,” *IEEE Trans. Ind. Informatics*, vol. 10, no. 2, pp. 1537–1546, May 2014.

[4] C. Canali, V. Cardellini, and R. Lancellotti, “Content Adaptation Architectures Based on Squid Proxy Server,” *World Wide Web*, vol. 9, no. 1, pp. 63–92, Mar. 2006.

[5] J. F. Kurose and K. W. Ross, *Computer Network: a top-down approach - 6th ed.*, vol. 1. 2013.

[6] J. N. Fidalgo and J. A. Lopes, “Load Forecasting Performance Enhancement When Facing Anomalous Events,” *IEEE Trans. Power Syst.*, vol. 20, no. 1, pp. 408–415, Feb. 2005.

[7] C. Brighenti and M. A. Sanz-Bobi, “Auto-Regressive Processes Explained by Self-Organized Maps. Application to the Detection of Abnormal Behavior in Industrial Processes,” *IEEE Trans. Neural Networks*, vol. 22, no. 12, pp. 2078–2090, Dec. 2011.

[8] Y.-J. Lee, Y.- R.Yeh, and Y.- C. F. Wang, “Anomaly Detection via Online Oversampling Principal Component Analysis,” *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 7, pp. 1460–1470, Jul. 2013.

[9] J. Benadit P and S. Francis F, “Science Direct Improving the Performance of a Proxy Cache Using Very Fast Decision Tree Classifier,” *Procedia – Procedia Comput. Sci.*, vol. 48, no. 48, pp. 304–312, 2015.

[10] K. Masuda, S. Ishida, and H. Nishi, “Cross-site Recommendation Application Based on the Viewing Time and Contents of Webpages Captured by a Network Router.”

[11] I. B. M. Austin, B. Road, A. Tx, R. Rajamony, and M. Elnozahy, “Measuring client-perceived response times on the www,” *3rd Conf. USENIX Symp. internet Technol. Syst.*, no. March, p. 16, 2001.

[12] L. G. Cárdenas, J. Sahuquillo, A. Pont, and J. A. Gil, “The Multikey Web Cache Simulator: a Platform for Designing Proxy Cache Management Techniques,” *Parallel, Distrib. Network-Based Process. 2004. Proceedings. 12th Euromicro Conf.*, pp. 390–397, 2004.

[13] Y. K. Lin and C. T. Yeh, “Using minimal cuts to optimize network reliability for a stochastic computer network subject to assignment budget,” *Comput. Oper. Res.*, vol. 38, no. 8, pp. 1175–1187, 2011.

[14] C. T. Yeh and L. Fiondella, “Optimal redundancy allocation to maximize multi-state computer network reliability subject to correlated failures,” *Reliab. Eng. Syst. Saf.*, 2016.

[15] L. Zazuli and A. Mardedi, “Developing Computer Network Based on EIGRP Performance Comparison and OSPF,” *International Journal of Advanced Computer Science and Applications*, vol. 6, no. 9.pp. 80–86, 2015.

[16] M. Sysel and O. Doležal, “An Educational HTTP Proxy Server,” *Procedia Eng.*, vol. 69, pp. 128–132, 2014.

[17] S. Romano and H. El Aarag, “A neural network proxy cache replacement strategy and its implementation in the Squid proxy server,” *Neural Comput. Appl.*, vol. 20, no. 1, pp. 59–78, Sep. 2010.

[18] T. Agarwal and M. A. Leonetti, “Design and Implementation of an IP based authentication mechanism for Open Source Proxy Servers in Interception Mode,” Feb. 2013.

[19] A. Rousskov and V. Soloviev, “A performance study of the Squid proxy on HTTP/1.0,” *World Wide Web*, vol. 2, no. 1, pp. 47–67, 1999.

[20] J. M. Abe, S. Akama, and K. Nakamatsu, *Introduction to Annotated Logics - Foundations for Paracomplete and Paraconsistent Reasoning*, 1st ed. Springer International Publishing, 2015.

[21] J. M. Abe, *Paraconsistent Intelligent Based-Systems: New Trends in the Applications of Paraconsistency*. Germany: Springer-Verlag, 2015.

[22] S. Akama, *Towards Paraconsistent Engineering*. Springer International Publishing, 2016.

[23] A. Jeffries, “Customizable Log Formats,” *Squid-cache Wiki*, 2015. [Online]. Available: http://wiki.squid-cache.org/Features/LogFormat. [Accessed: 04-Jul-2017].

[24] C. Babu and G. Gunasingh, “DESH: Database evaluation system with hibernate ORM framework,” in *2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 2016, pp. 2549–2556.

5 LIMITAÇÕES DO PROJETO

Foram observadas algumas limitações no projeto, uma vez percebida a ocorrência de falsos negativos. Verificou-se, *in loco*, a existência de equipamentos infectados por determinados tipos de *malware*, cuja análise não foi confirmada pelo analisador do tráfego de rede. Este, a partir dos atributos de rede escolhidos para análise de situações anômalas, não foi capaz de detectar a ocorrência de *malwares* que possuíam pouca ou nenhuma atividade na rede de computadores e que, portanto, não geravam valores capazes de serem quantificados e analisados. Estes programas maliciosos, denominados como *standalone,* não podem ser facilmente detectáveis pelo seu comportamento anômalo em uma rede de computadores. Por outro lado, e exatamente por essa mesma característica, possuem maior dificuldade de propagação, tendo em vista que a rede de computadores não se apresenta como a forma primária de infecção desse tipo de *malware*, o que diminui sensivelmente sua virulência.

Para esse tipo de programa malicioso, é necessário o uso de atributos diferentes daqueles utilizados nesse projeto. Trata-se, portanto, de efetuar novos levantamentos de requisitos, uma vez que a abordagem é completamente distinta, e fora do escopo do que foi discutido previamente.

Esses requisitos, uma vez estabelecidos e testados, poderão fazer parte do desenvolvimento de um sistema multifatorial e mais abrangente de busca de *malwares*, capaz de detectar programas maliciosos de naturezas distintas em uma rede de computadores. Ainda assim, é necessário discutir qual será a fonte de dados utilizada, tendo em vista que os *logs* de funcionamento da rede não poderão servir a esse propósito, em razão de os atributos extraídos dessa fonte de informação serem específicos para redes de computadores.

6 CONSIDERAÇÕES FINAIS

A implantação do analisador do tráfego de rede trouxe diversas mudanças para a faculdade de tecnologia pública que foi objeto desta pesquisa durante seu período de utilização.

Como é comum em algumas instituições de ensino público, o número de técnicos envolvidos na manutenção de equipamentos é insuficiente e não acompanha o aumento na quantidade de equipamentos e na complexidade da sua respectiva infraestrutura. Dessa forma, a abordagem dos responsáveis pelo setor de TI diante de problemas na rede basicamente possui um caráter corretivo, quase sempre após a abertura de um chamado técnico de um docente ou funcionário com alguma reclamação. Portanto, em função da falta de tempo disponível, a possibilidade de prospecção preventiva de problemas dessa natureza ocorre quase sempre no período de férias, ocasião em que, devido à pequena atividade desenvolvida, quase não ocorrem problemas de maior gravidade ou complexidade.

Com a utilização do analisador do tráfego de rede, foi possível realizar diariamente uma rápida verificação prévia da utilização dos equipamentos de rede, cujos comportamentos poderiam sugerir a ocorrência de problemas de alguma natureza, ainda que não houvesse uma reclamação formal a respeito de seu funcionamento. Tendo em vista a necessidade de correção rápida dos problemas e retorno dos equipamentos à operação, estes tinham seus sistemas operacionais reinstalados e seus aplicativos novamente customizados, sem determinar a princípio a fonte de problema. Dessa forma, esses mesmos equipamentos eram rapidamente recolocados para a utilização dos usuários. Portanto, a causa do problema em determinado dispositivo não se constituía em prioridade para o técnico responsável.

Porém, como forma de extrair informações das anomalias detectadas, foi criado um grupo de controle, a partir de equipamentos distribuídos por diversos pontos da rede. Os computadores foram escolhidos de forma sequencial, assim que o analisador determinava a ocorrência de um comportamento não esperado em cada um deles. Ao longo de quatro semanas, foram feitos levantamentos das causas dos problemas, cujos resultados podem ser expressos na Tabela 11:

Tabela 11 – Determinação de ocorrência de falhas na instituição de ensino

Semana	Equipamentos com falha	Setor Administrativo	Setor Acadêmico	Tipo de Problema Detectado
Primeira	41	5	36	Vírus, *Worms*, *Spywares*::39 Falhasde *hardware* / meio físico: 2
Segunda	32	1	31	Vírus, *Worms*, *Spywares*:: 31Falhas de *hardware* / meio físico: 1
Terceira	10	1	9	Vírus, *Worms*, *Spywares*:: 10 Falhas de *hardware* / meio físico: 0
Quarta	11	0	11	Vírus, *Worms*, *Spywares*:: 11Falhas de *hardware* / meio físico: 0

Fonte: Autor.

Ainda de acordo com a Tabela 11, dois fatos interessantes podem ser observados. O primeiro diz respeito às falhas de *hardware* ou meio físico. Considerando-se que são problemas que dependem pouco da ação dos usuários, pode-se notar que poucos casos foram verificados na primeira semana. A partir da segunda semana, e com os respectivos reparos efetuados, o número progressivamente cai até não haver qualquer ocorrência na terceira e quarta semanas.

O segundo fato diz respeito à ocorrência dos vírus, *worms* e *spywares*, que praticamente dominam a maior parte das ocorrências ao longo das quatro semanas. Pode-se observar uma grande prevalência desses *malwares* no setor acadêmico, e sensivelmente menor no setor administrativo. É importante ressaltar que todos os equipamentos possuíam antivírus coorporativos instalados e atualizados no momento em que as anomalias foram detectadas, o que demonstra que a simples presença deles não é capaz de evitar ocorrências dessa natureza. Levando-se em conta que a rede da instituição possui um número aproximado de 100 equipamentos, pode-se julgar alto o número de computadores afetados na primeira semana. Observa-se também que a detecção precoce e reinstalação dos equipamentos foi suficiente para fazer com que os números declinassem sensivelmente da primeira até a quarta semana, tarefa que os antivírus instalados não foram capazes de realizar.

Foram sugeridas ações preventivas junto aos docentes, discentes e funcionários com o objetivo de minimizar a chance de entrada de *malwares* na rede. Dentre as sugestões, de acordo com Zuben (2012), podem ser citadas:

a) Manter o computador com todas as atualizações aplicadas e com todos os programas instalados com as versões mais recentes;
b) Utilizar mecanismos de segurança, tais como *firewall* pessoal, *antimalware*, *antiphishing*, *antispam*, complementos, extensões e plugins;
c) Usar apenas programas originais e as configurações de segurança já disponíveis;
d) Ter cautela ao instalar aplicativos desenvolvidos por terceiros;
e) Utilizar senhas contendo uma grande quantidade e diferentes tipos de caracteres, além de números aleatórios;
f) Evitar a utilização de sequências de teclado e de dados pessoais, tais como: nome, sobrenome, contas de usuário, números de documentos, placas de carros e números de telefones.

Por fim, pode-se concluir que, ainda que nem todos os *malwares* existentes na rede de computadores pudessem ser detectados pelo analisador, o índice de acertos na busca por programas maliciosos que fazem uso da rede foi alto, justamente aqueles cuja taxa de replicação é significativa. Isso corrobora com a ideia de que o objetivo específico 1 foi atingido, tendo em vista que os atributos determinados pelos especialistas estavam corretos para a análise da rede. Por outro lado, o objetivo específico 2 também foi alcançado, pois a própria informação produzida pela operação da rede permitiu a parametrização do seu funcionamento, através dos atributos determinados previamente. Por fim, o objetivo específico 3 também pode ser considerado cumprido, ainda que com as ressalvas discutidas anteriormente, já que os *malwares* que atendem ao perfil abordado nesse estudo foram detectados e a observação *in loco* e diminuição gradativa de infecção dos equipamentos envolvidos corroboraram com esta percepção.

7 TRABALHOS FUTUROS

Este trabalho utilizou como fonte de dados os *logs* consolidados de uma rede de computadores, ou seja, os dados obtidos ao final de um período de operação da rede. Dessa forma, diversas situações do funcionamento de uma rede de computadores são contempladas, permitindo a quantificação mais precisa do que ocorreuao longo do tempo nos diversos dispositivos conectados, assim como a diluição de situações anômalas pontuais. É importante considerar o descarte dessas situações, uma vez que o ambiente de trocas de dados não é hermético e isolado. Ao contrário, ele depende de outras redes externas também sujeitas a situações de congestionamento ou problemas de fluxo, cujas consequências acabam por impactar indiretamente a rede sob monitoramento.

Porém, mesmo que esse método permita a determinação de valores de operação precisos, ele tem como desvantagem a demora na localização do problema porque o sistema necessita “aprender” o que é considerado ou não normal. Muitas vezes, dependendo da natureza e do quão crítica é a atividade desenvolvida, este retardo não é aceitável.

Em ambientes que necessitam de respostas expressas, é necessário que seja implementado um sistema em tempo real que permita a localização mais rápida do ponto em que o problema ocorre. Considerando que mesmo nesse modelo é necessário que haja a aprendizagem do analisador do tráfego de dados do que é ou não esperado dentro de uma rede, uma possível abordagem é a utilização de atributos de rede que não foram considerados neste trabalho, tais como latência e largura de banda. Dessa forma, o número de fatores a serem apontados deverá ser maior, ainda que a metodologia possa ser mantida praticamente a mesma.

De forma diferente à proposta original, cujo foco é interno e compreendendo a rede local da empresa ou instituição, outra possibilidade, pensando-se ainda na busca por situações anômalas, é a utilização do analisador do tráfego de dados na busca por problemas externos, ou seja, entidades que fazem uso de programas maliciosos com o objetivo de invadir/atacar a rede local. Como essas intrusões também podem ficar registradas nos mesmos dispositivos, em seus respectivos *logs,* uma nova janela de oportunidade pode ser aberta em uma das áreas mais sensíveis das redes de computadores.

Outra possibilidade é a utilização da Lógica $E\tau$ para tratar a limitação de projeto apresentada anteriormente, ao se buscarem *malwares* de comportamento *standalone*. Nesse caso, não são aplicáveis os atributos relativos à rede, pois, para esse tipo de programa malicioso, eles não são representativos, como foi verificado localmente ao se detectarem falsos negativos. Uma possível abordagem é utilizar atributos relativos à leitura e à gravação anômalas nos sistemas de arquivos, tendo em vista que o processo de replicação e comportamento anômalo de alguns desses tipos de *malwares* ocorrem nas mídias em que estão armazenados. O problema dessa abordagem é que ela deixa de ter um caráter sistêmico e não invasivo, já que esse comportamento ocorre localmente em cada computador, portanto, de forma descentralizada, e não pode ser quantificado a partir de uma fonte única e externa aos dispositivos envolvidos.

REFERÊNCIAS BIBLIOGRÁFICAS

ABE, J. M. **Fundamentos da Lógica Anotada, Tese de Doutorado**. [s.l.] FFLCH - USP, 1992.

ABE, J. M. **Paraconsistent Intelligent Based-Systems: New Trends in the Applications of Paraconsistency**. Germany: Springer-Verlag, 2015.

ABE, J. M.; AKAMA, S.; NAKAMATSU, K. **Introduction to Annotated Logics - Foundations for Paracomplete and Paraconsistent Reasoning**. 1. ed. [s.l.] Springer International Publishing, 2015.

AKAMA, S. **Towards Paraconsistent Engineering**. [s.l.] Springer International Publishing, 2016.

AUSTIN, I. B. M. et al. Measuring client-perceived response times on the www. **the 3rd conference on USENIX Symposium on internet Technologies and Systems**, n. March, p. 16, 2001.

BABU, C.; GUNASINGH, G. **DESH: Database evaluation system with hibernate ORM framework**. 2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI). **Anais**...IEEE, set. 2016 Disponível em: <http://ieeexplore.ieee.org/document/7732441/>. Acesso em: 2 set. 2017.

BEN-PORAT, U.; BREMLER-BARR, A.; LEVY, H. Computer and network performance: Graduating from the “age of Innocence”. **Computer Networks**, v. 66, p. 68–81, 2014.

BENADITP, J.; FRANCISF, S. ScienceDirect Improving the Performance of a Proxy Cache Using Very Fast Decision Tree Classifier. **Procedia - Procedia Computer Science**, v. 48, n. 48, p. 304–312, 2015.

BRIGHENTI, C.; SANZ-BOBI, M. A. Auto-Regressive Processes Explained by Self-Organized Maps. Application to the Detection of Abnormal Behavior in Industrial Processes. **IEEE Transactions on Neural Networks**, v. 22, n. 12, p. 2078–2090, dez. 2011.

CANALI, C.; CARDELLINI, V.; LANCELLOTTI, R. Content Adaptation Architectures Based on Squid Proxy Server. **World Wide Web**, v. 9, n. 1, p. 63–92, 2 mar. 2006.

CERT. **Sobre o CERT**. Disponível em: <https://www.cert.br/sobre>. Acesso em: 31 out. 2017a.

CERT. **Estatísticas dos incidentes reportados ao CERT**. Disponível em: <https://www.cert.br/stats/incidentes/>. Acesso em: 20 out. 2017b.

CERT. **Incidentes Reportados ao CERT.br -- Janeiro a Dezembro de 2016**. Disponível em: <https://www.cert.br/stats/incidentes/2016-jan-dec/total.html>. Acesso em: 20 out. 2017c.

CGI.BR. **Cartilha de Segurança para internet, versao 4.0 / CERT.br**. Disponível em: <http://cartilha.cert.br/privacidade/>. Acesso em: 22 out. 2017.

CHIAPPA, J. N. **The ARPANET in December 1969**. Disponível em: <http://mercury.lcs.mit.edu/~jnc/tech/arpageo.html>. Acesso em: 9 nov. 2017.

COMER, D. **internetworking with TCP/IP**. 4. ed. Upper Saddle River, New Jersey: Alan Apt, 2000.

DA SILVA FILHO, J.I., G. L. T. & J. M. A. **Uncertainty Treatment Using Paraconsistent Logic - Introducing Paraconsistent Artificial Neural Networks**. [s.l: s.n.].

DAVISON, B. D.; WU, B. Implementing a web proxy evaluation architecture. n. December, 2004.

FERNANDES, G. et al. **Statistical, forecasting and metaheuristic techniques for network anomaly detection**. Proceedings of the 30th Annual ACM Symposium on Applied Computing - SAC '15. **Anais**...New York, New York, USA: ACM Press, 2015. Disponível em: <http://dl.acm.org/citation.cfm?doid=2695664.2695852>. Acesso em: 18 mar. 2018.

FERNANDEZ-PRIETO, J. A. et al. Optimisation of control parameters for genetic algorithms to test computer networks under realistic traffic loads. **Applied Soft Computing Journal**, v. 12, n. 7, p. 1875–1883, 2012.

FIDALGO, J. N.; LOPES, J. A. Load Forecasting Performance Enhancement When Facing Anomalous Events. **IEEE Transactions on Power Systems**, v. 20, n. 1, p. 408–415, fev. 2005.

FOSSACECA, J. M.; MAZZUCHI, T. A.; SARKANI, S. MARK-ELM: Application of a novel Multiple Kernel Learning framework for improving the robustness of Network Intrusion Detection. **Expert Systems with Applications**, v. 42, n. 8, p. 4062–4080, 2015.

GARSHASBI, M. S. Fault localization based on combines active and passive measurements in computer networks by ant colony optimization. **Reliability Engineering & System Safety**, v. 152, p. 205–212, 1 ago. 2016.

GÓMEZ-CORRAL, A. On the applicability of the number of collisions in p-persistent CSMA/CD protocols. **Computers & Operations Research**, v. 37, n. 7, p. 1199–1211, 1 jul. 2010.

GORRY FAIRHURST. **CSMA/CD**. Disponível em: <http://www.erg.abdn.ac.uk/users/gorry/course/lan-pages/csma-cd.html>. Acesso em: 5 nov. 2017.

GRANA, J. et al. A likelihood ratio anomaly detector for identifying within-perimeter computer network attacks. **Journal of Network and Computer Applications**, v. 66, p. 166–179, 2016.

J.B.SHUKLA et al. Modeling and analysis of the effects of antivirus software on an infected computer network. **Applied Mathematics and Computation**, v. 227, p. 11–18, 15 jan. 2014.

KUROSE, J. F.; ROSS, K. W. **Computer Network: a top-down approach - 6th ed.** [s.l: s.n.]. v. 1

LEE, Y.-J.; YEH, Y.-R.; WANG, Y.-C. F. Anomaly Detection via Online Oversampling Principal Component Analysis. **IEEE Transactions on Knowledge and Data Engineering**, v. 25, n. 7, p. 1460–1470, jul. 2013.

LIN, Y. K.; HUANG, C. F. Stochastic computer network under accuracy rate constraint from QoS viewpoint. **Information Sciences**, v. 239, p. 241–252, 2013.

LIN, Y. K.; YEH, C. T. Using minimal cuts to optimize network reliability for a stochastic computer network subject to assignment budget. **Computers and Operations Research**, v. 38, n. 8, p. 1175–1187, 2011.

MASUDA, K.; ISHIDA, S.; NISHI, H. Cross-site Recommendation Application Based on the Viewing Time and Contents of Webpages Captured by a Network Router. 2013.

MISRA, A. K.; VERMA, M.; SHARMA, A. Capturing the interplay between malware and anti-malware in a computer network. **Applied Mathematics and Computation**, v. 229, p. 340–349, 2014.

OBAIDAT, M. S.; NICOPOLITIDIS, P.; ZARAI, F. Modeling and Simulation of Computer Networks and Systems. In: [s.l.] Elsevier, 2015. p. 187–223.

PENA, E. H. M. et al. Anomaly detection using digital signature of network segment with adaptive ARIMA model and Paraconsistent Logic. **Proceedings - International Symposium on Computers and Communications**, 2014.

PIMENTA, A. P.; ABE, J. M.; DE OLIVEIRA, C. C. **An analyzer of computer network logs based on paraconsistent logic**. [s.l: s.n.]. v. 460

PIMENTA JR, A. P.; ABE, J. M. An Analyzer of Computer Networks Logs Based on Paraconsistent Logic. **IFIP Advances in Information and Communication Technology**, v. 460, p. 620–627, 2015.

PIMENTA JR, A. P.; ABE, J. M. Determination of operating parameters and performance analysis of computer networks with ParaconsistentAnnotated Evidential Logic Eт. **IFIP Advances in Information and Communication Technology**, v. 1, p. 1–9, 2016.

ROSEN, R. **Linux Kernel Networking - advanced topics : Neighboring and IPsec**. [s.l: s.n.].

ROUSSKOV, A.; SOLOVIEV, V. A performance study of the Squid proxy on HTTP/1.0. **World Wide Web**, v. 2, n. 1, p. 47–67, 1999.

SANTOS, R. R. DOS; MOREIRA, A. M.; ROCHA, A. S. DA. **Curso IP v6 Básico**. 1. ed. São Paulo, Brazil: [s.n.].

TANENBAUM, A. S. **Redes de Computadores**. 4. ed. São Paulo, Brazil: Campus, 2003.

WHITE, D.; REA, A. A Backpropagation Neural Network for Computer Network Security. **Journal of Computer Science**, v. 2, n. 9, p. 710–715, 2006.

XU, L. DA; HE, W.; LI, S. internet of Things in Industries: A Survey. **IEEE Transactions on Industrial Informatics**, v. 10, n. 4, p. 2233–2243, nov. 2014.

YAACOB, A. H. et al. **ARIMA Based Network Anomaly Detection**. 2010 Second International Conference on Communication Software and Networks. **Anais**...IEEE, 2010Disponível em: <http://ieeexplore.ieee.org/document/5437603/>. Acesso em: 20 mar. 2018

YEH, C. T.; FIONDELLA, L. Optimal redundancy allocation to maximize multi-state computer network reliability subject to correlated failures. **Reliability Engineering and System Safety**, 2016.

ZHU, B.; SASTRY, S. **Revisit Dynamic ARIMA Based Anomaly Detection**. 2011 IEEE Third Int'l Conference on Privacy, Security, Risk and Trust and 2011 IEEE Third Int'l Conference on Social Computing. **Anais**...IEEE, out. 2011Disponível em: <http://ieeexplore.ieee.org/document/6113293/>. Acesso em: 20 mar. 2018.

ZHUMING BI; LI DA XU; CHENGEN WANG. internet of Things for Enterprise Systems of Modern Manufacturing. **IEEE Transactions on Industrial Informatics**, v. 10, n. 2, p. 1537–1546, maio 2014.

ZUBEN, M. VON. **Spywares, Worms, Bots e Boas Práticas de SegurançaCentro de Estudos, Resposta e Tratamento de Incidentes de Segurança no Brasil**, 2012. Disponível em: <https://www.cert.br/docs/palestras/certbr-puccamp2012.pdf>. Acesso em: 2 nov. 2017