UNIVERSIDADE FEDERAL DE SANTA MARIA
CENTRO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Anderson Monteiro da Rocha

MÉTODO AGNÓSTICO DE DETECÇÃO DA QUEBRA DA NEUTRALIDADE NA INTERNET PELOS ISPS

Santa Maria, RS
2018

Anderson Monteiro da Rocha

MÉTODO AGNÓSTICO DE DETECÇÃO DA QUEBRA DA NEUTRALIDADE NA INTERNET PELOS ISPS

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação(PPGCC) da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para obtenção do título de **Mestre em Ciência da Computação**.

Orientador: Prof. Dr Carlos Raniery Paula dos Santos

Santa Maria, RS

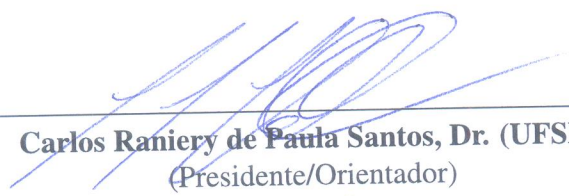
2018

Anderson Monteiro da Rocha

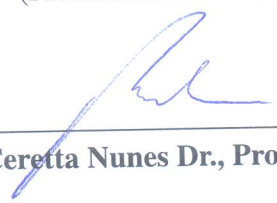
MÉTODO AGNÓSTICO DE DETECÇÃO DA QUEBRA DA NEUTRALIDADE NA INTERNET PELOS ISP'S

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação (PGCC) da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para obtenção do título de **Mestre em Ciência da Computação**.

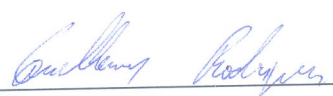
Aprovado em 28 de fevereiro de 2018:



Carlos Raniery de Paula Santos, Dr. (UFSM)
(Presidente/Orientador)



Raul Ceretta Nunes Dr., Prof. (UFSM)



Guilherme da Cunha Rodrigues Dr., Prof. (IFSUL)

Santa Maria, RS
2018

DEDICATÓRIA

Dedico este trabalho à minha família e aos meus verdadeiros amigos...

AGRADECIMENTOS

Se você está lendo esta página é porque eu consegui. E não foi fácil chegar até aqui. Das várias tentativas no processo seletivo, passando pela aprovação até a conclusão do Mestrado, foi um longo caminho percorrido. Nada foi fácil, mas enfim consegui!

Primeiramente, um agradecimento mais do que especial para minha esposa, Fabiana (FAFI) foi a pessoa que me proporcionou uma sustentação e apoio incondicional em toda esta caminhada. Não sei se existe felicidade perfeita, mas ao seu lado sou perfeitamente feliz.

Aos meus filhos Larissa e Matheus. Larissa, minha revisora oficial dos textos e das ideias. Matheus, o responsável por proporcionar os momentos de brincadeiras e alegrias para esquecer um pouco dos estudos.

Agradeço a minha mãe, Salimar, meu exemplo de vida, minha inspiração, o meu muito obrigado por me fazer o que sou hoje. Aos meus irmãos Sabrina, Henrique, o meu padrastro Almeida, aos sogros Ori e Marileira pela convivência e ajuda quando necessário.

Agradeço ao meu orientador, Carlos Raniery pela orientação, ensinamentos, conselhos, paciência e sentido prático com que sempre me orientou.

Um agradecimento muito especial e de coração aos meus amigos colegas de mestrado que conquistei durante esses dois anos. Foram inúmeros momentos divididos, muito café, poucas cerveja e raras caipirinhas (#sqn). Amigos estes que tornaram mais leve e tranquilo meu trabalho. Um abraço especial ao Nilton Batista, Thales Tavares, Vinícius Fulber e ao Leonardo Marcuzzo, saibam que foram importantes na minha vida acadêmica e na aquisição de novos conhecimentos. Que o NoSTRUn tenha muito sucesso e nos proporcione publicações e viagens.

Um agradecimento a todos aqueles que me deram força, coragem e motivação. Agradeço ao apoio incondicional da família, base de tudo, que durante estes dois anos torceram por mim. E por fim, um muito obrigado aos meus colegas de Instituto Federal Farroupilha - Câmpus São Vicente do Sul, são nos pequenos detalhes que se faz grandes amizades.

"Talvez não tenha conseguido fazer o melhor, mas lutei para que o melhor fosse feito. Não sou o que deveria ser, mas graças a dedicação e a abdicação, não sou o que era antes."

(MARTHIN LUTHER KING)

RESUMO

MÉTODO AGNÓSTICO DE DETECÇÃO DA QUEBRA DA NEUTRALIDADE NA INTERNET PELOS ISPS

AUTOR: ANDERSON MONTEIRO DA ROCHA
ORIENTADOR: CARLOS RANIERY PAULA DOS SANTOS

Atualmente o debate sobre neutralidade da rede está cada vez mais pertinente, vários países estão discutindo normas e leis afim de estabelecer como deve ser tratado o tráfego de dados pelos ISPs. Neste contexto, é importante compreender as técnicas, as motivações e os vários tipos de discriminações que os ISPs podem realizar. A partir desta constatação, esta dissertação apresenta a criação de um método que identifique a quebra da neutralidade de maneira agnóstica em um ambiente controlado. Um detalhe importante do método é ser agnóstico na identificação da quebra da neutralidade. Sendo assim, independentemente do tipo de protocolo, da aplicação, do serviço, do tamanho do pacote ou de qualquer outra informação que o fluxo possuir o método funciona da mesma maneira. Outro fator a ser ressaltado é a utilização de múltiplas métricas de desempenho (latência, *jitter*, *throughput* e perda de pacotes) para identificação da quebra de neutralidade. O método proposto também é capaz de distinguir entre diferenciação de tráfego e a degradação que ocorre sobre os fluxos. Todas as amostras capturadas de cada métrica foram transformadas em um índice de valor 0 (zero) ou 1 (um), utilizou-se o controle estatístico de processos (CEP) na transformação desses índices. Para avaliação do método, desenvolveu-se um ambiente controlado com quatro tipos diferentes de cenários. Nestes cenários o roteador simula as políticas de configuração de um ISP. As simulações foram de tráfego neutro, de tráfego com descarte de pacotes, de *traffic shaping* no tráfego e por último um cenário onde o roteador executa *delay* em determinados fluxos. Com os resultados obtidos foi possível avaliar que o método agnóstico proposto por esta dissertação demonstrou-se eficaz nos quatro cenários.

Palavras-chave: Neutralidade da Rede. Métricas de Desempenho. Agnóstico.

ABSTRACT

AGNOSTIC METHOD OF DETECTING THE BREAKDOWN OF INTERNET NEUTRALITY BY ISP

AUTHOR: ANDERSON MONTEIRO DA ROCHA
ADVISOR: CARLOS RANIERY PAULA DOS SANTOS

Nowadays the discussion about network neutrality is more and more necessary, many countries are debating norms and laws to establish how the data traffic must be treated by the ISP. In this context, it is important understand the techniques, motivations and the types of traffic discrimination that can be realized by the ISP. From this knowledge, this Master dissertation presents an agnostic method to detected the network neutrality breaking in a controlled environment. Being an agnostic method for neutrality breaking detection, independently of the protocol, application, service, packet size or any other information of the flow, this method works in the same way. Other important factor is the utilization of multiple performance metrics (latency, jitter, throughput and packet loss) to identify the neutrality breaking. The proposed method also is capable to distinguish between traffic discrimination and natural degradation forms. All the captured samples of each metric were transformed in an index with value between 0 (zero) and 1 (one), a simple statistic process (CEP) was used in the indexes transformation. To evaluate the method, a controlled environment was developed with four different scenarios. In these scenarios, the router simulates the configuration policies of an ISP. The simulations were of neutral traffic, with packet discarding, with traffic shaping and with the router delaying the packets. With the obtained results was possible evaluate that the proposed agnostic method demonstrated itself effective in all tested scenarios.

Keywords: Network neutrality. perfomarce metric. agnostic.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

AF	Assured Forwarding
BE	Best Effort
CEP	Controle Estatístico de Processos
D-ITG	Distributed Internet Traffic Generator
DNS	Domain Name System
DPI	Deep Packet Inspections
DoS	Denial of Service
EF	Expedited Forwarding
FCC	Federal Communications Commission
FIFO	First In First Out
FTP	File Transfer Protocol
GB	GigaBytes
HTTP	Hypertext Transfer Protocol
ICMP	Internet Control Message Protocol
IDS	Intrusion Detection System
IETF	Internet Engineering Task Force
IJT	Índice por Janela de Tempo
IMJ	Índice por Métrica da Janela de tempo
IP	Internet Protocol
ISP	Internet Service Provider
ITF	Índice Total por Fluxo
KVM	Kernel Virtual Machine
LC	Linha Central
LIC	Limite Inferior de Controle
LSC	Limite Superior de Controle
M-LAB	Measurement Lab
Mbps	Megabits por segundo
MCI	Marco Civil da Internet
MPLS	MultiProtocol Label Switching
P2P	Peer-to-Peer
PPlive	Peer-to-Peer Streaming Video Network
QoE	Quality of Experience

QoS	Quality of Service
RED	Random Early Drop
RSPV	Resource Reservation Protocol
RtN	Roteador Neutro
RtNN1	Roteador Não-Neutro que realiza descarte de pacotes
RtNN2	Roteador Não-Neutro que realiza traffic shaping
RtNN3	Roteador Não-Neutro que realiza delay
RTT	Round-Trip Time
SMTP	Simple Mail Transfer Protocol
TCP	Transfer Control Protocol
VoIP	Voice of Internet Protocol
WFQ	Weighted Fair Queueing

SUMÁRIO

1 INTRODUÇÃO

O tráfego de dados na Internet vem se multiplicando a cada ano (FLACH et al., 2016) (CISCO, 2015), assim como o número de usuários e, principalmente, a diversidade de serviços e aplicações (PEITZ; SCHUETT, 2016). Os *Internet Service Providers* (ISPs) possuem função importante neste crescimento, pois através deles os usuários são conectados à Internet.

Originalmente, a Internet era neutra e projetada para seguir o princípio de ponta-a-ponta (WU, 2003). No entanto a evolução da arquitetura da Internet proporcionou o surgimento de novas tecnologias, interesses econômicos e políticos (HOSEIN et al., 2015). Estas novas tecnologias demandaram mais recursos de rede (CALLADO et al., 2009), afetando diretamente os ISPs com relação ao roteamento do tráfego de dados. Com a intenção de melhorar a segurança e os serviços prestados, os ISPs começaram a realizar uma inspeção profunda de pacotes, assim como a priorização de serviços (CROWCROFT, 2007). Com isto, surgiu o debate sobre a neutralidade de rede que nos últimos anos ganhou uma atenção da sociedade tornando-se assim uma discussão de abrangência global (GROVE; AGIC; SEDLMEIR, 2012).

A responsabilidade principal dos ISPs é de analisar exclusivamente o cabeçalho dos pacotes para definir o caminho que estes devem seguir. Entretanto, os provedores de serviço utilizam mecanismos de discriminação quando os pacotes trafegam através de suas infraestruturas de comutação (DECKER; EIDENBENZ; WATTENHOFER, 2013). Neste cenário, os ISPs possuem uma tendência em aplicar configurações de modelagem de tráfego como forma de discriminar um conteúdo ou aplicação na Internet, entre elas podemos citar: (i) bloquear determinados tipos de pacotes; (ii) reduzir a velocidade dependendo da aplicação; (iii) cobrar um preço diferente pelo acesso a um determinado conteúdo. Com essas ações os ISPs podem configurar seus equipamentos para degradar diversos serviços, por exemplo: BitTorrent (DISCHINGER et al., 2008), jogos *on-line* (WU, 2003) e *streaming* de vídeos (YOO, 2014).

Existem também uma série de outros motivos para que um ISP possua interesses em realizar diferenciação de tráfego em seus clientes (VAN SCHEWICK, 2012). Um ISP pode querer aumentar seu domínio de monopólio e desejar elevar seus lucros ao limitar acesso a um conjunto de serviços que poderiam começar a ser tratados de forma diferenciada. Assim como, o ISP poderia fazer acordos e cobrar de outros ISPs e empresas de que fornecem conteúdo ou serviço para ter um tratamento preferencial dos seus dados.

A medida que cresce a quantidade de usuários na Internet também aumentam os países

que estão debatendo sobre o tema da neutralidade de rede. Nesta perspectiva, violações da neutralidade e casos reais são relatados em trabalhos científicos, assim como também em denuncias da imprensa e de usuários. Em 2005, a provedora Telus do Canadá bloqueou o acesso à páginas da *Web* com serviços de voz (AUSTEN, 2005), da mesma forma em 2011 o ISP MobileVikings da Bélgica bloqueou o *Domain Name System* (DNS) à paginas *Web* (RESPECTMYNET, 2009).

Similarmente, na Nova Zelândia em 2013 (ESNAASHARI, 2014) e nos EUA em 2007 provedores interromperam o tráfego de dados referentes a aplicativos P2P (TOPOLSKI, 2009). Outras operadoras que violaram a neutralidade com degradação de tráfego foram a T-Mobile da Alemanhã que degradava tráfego da aplicação Skype (KENDRIC, 2009), e a British Telecommunications da Inglaterra que degradava tráfego de vídeos da página *Web* da emissora BBC.

Além disso, existem outros relatos de operadoras que realizaram o *traffic shaping* em seus clientes, como por exemplo em 2011 o ISP Free da França (RESPECTMYNET, 2009), em 2012 nos EUA e no Canadá (MUELLER; ASGHARI, 2012) e no Chile em 2013 (BUSTOS-JIMÉNEZ; FUENZALIDA, 2014). Em 2014, houve uma redução na taxa de velocidade entre o fornecedor de conteúdo Netflix e os provedores Verizon e Comcast, ambos ISPs dos EUA (BRODKIN, 2014). No Brasil, em 2016 a operadora Claro recebeu inumeras reclamações e foi notificada por estar realizando *traffic shaping* em um jogo *on-line* (IDEC, 2016).

Neste contexto da neutralidade de rede é fundamental compreender as técnicas, as motivações e os vários tipos de discriminação, como eles são realmente aplicados na prática. As discussões de quebra da neutralidade precisam ser transparentes aos usuários (COMMISSION et al., 2011). O usuário tem o direito de saber se seu ISP é capaz de priorizar tráfego de fluxos diferentes, ou até mesmo bloquear ou proibir certos tipos de pacotes, bem como saber se seu ISP está configurado com *Deep Packet Inspection* (DPI), *Traffic Shaping*, ou Qualidade de Serviço (*Quality of Service* - QoS).

Para responder tais indagações é necessário comprovar a discriminação de tráfego da Internet pelos ISPs (DISCHINGER et al., 2010). Ao compreender como um ISP gerencia e controla a transferência de informações, será possível entender melhor como essas práticas afetam diretamente o consumidor. Para uma melhor análise da neutralidade é preciso distinguir a violação da neutralidade de outras causas de degradação (congestionamento e configuração incorreta) (PATHANIA; KALRA, 2012) .

Contudo, os trabalhos existentes sobre detecção da quebra da neutralidade são específi-

cos para uma aplicação, protocolo ou a um mecanismo de discriminação em particular (BASSO; SERVETTI; DE MARTIN, 2011), (DISCHINGER et al., 2010), (HOSEIN et al., 2015), (KANUPARTHY; DOVROLIS, 2010), (KREIBICH et al., 2010), (TARIQ et al., 2009), (ZHANG; MARA; ARGYRAKI, 2014) e (ZHANG; MAO; ZHANG, 2009). Além disso, os estudos se limitam em apontar uma degradação no desempenho da rede, sem informar se este ocorre devido à configuração do ISP ou às condições naturais da rede, tais como congestionamento.

Para realizar uma detecção mais efetiva é fundamental que sejam avaliadas múltiplas métricas de desempenho a fim de detectar ou não a quebra da neutralidade (COMMISSION et al., 2011). Contudo, os trabalhos disponíveis na literatura até então não realizam tal tipo de análise por estes serem mais simples, concentrando-se apenas em um conjunto limitado de métricas. Entre os trabalhos que abordam as soluções de detecção nenhum apresenta uma ferramenta única de medição capaz de analisar diversas métricas de desempenho (*e.g.*, *throughput*, latência, *jitter* e perda) simultaneamente.

A partir desta constatação, a dissertação tem como objetivo a criação de um método que identifique a quebra da neutralidade em um ambiente controlado. O método proposto foi avaliado em um ambiente onde tráfego sintético é gerado entre *hosts* interligados por meio de um roteador que implementa métodos diferentes de discriminação. Um detalhe importante do método é ser agnóstico na identificação da quebra de neutralidade. Sendo assim, independentemente do tipo de protocolo, da aplicação, do serviço, do tamanho do pacote ou de qualquer outra informação que o fluxo possuir o método funciona da mesma maneira.

Este trabalho está organizado da seguinte forma. O Capítulo 2, *Background*, relata sobre os tópicos que fundamentam a neutralidade de rede. No Capítulo 3 é realizada a Revisão da Literatura dos trabalhos mais relevantes. No Capítulo 4 é apresentanda a Solução Proposta por esta pesquisa. O Capítulo 5 descreve a Metodologia utilizada. O Capítulo 6 mostra a avaliação e discussão dos resultados. Por fim, o Capítulo 7 apresenta a Conclusão.

2 BACKGROUND

O tema neutralidade da rede tem sido predominante no debate sobre as políticas de Internet (BASSO; SERVETTI; DE MARTIN, 2011). A discussão básica é compreender se os ISP's devem ser autorizados a diferenciar o tráfego da Internet que passa pela sua infraestrutura ou se a neutralidade da rede deve ser explicitamente protegida, o que consagra a característica da Internet desde seu início. Desta forma, este Capítulo apresenta o conceito da Neutralidade de Rede, ressalta a importância para Qualidade de Serviço (Qos) na Internet e finaliza tratando de Anomalias de Tráfego.

2.1 NEUTRALIDADE DE REDE

O termo Neutralidade de Rede foi introduzido por Tim Wu (2003), definindo características do tratamento do tráfego de dados realizado por provedores da Internet. A neutralidade de rede é um princípio de arquitetura que trata os pacotes de dados que trafegam nas redes de forma isonômica, não os discriminando em razão de seu conteúdo, origem e destino (WU, 2003).

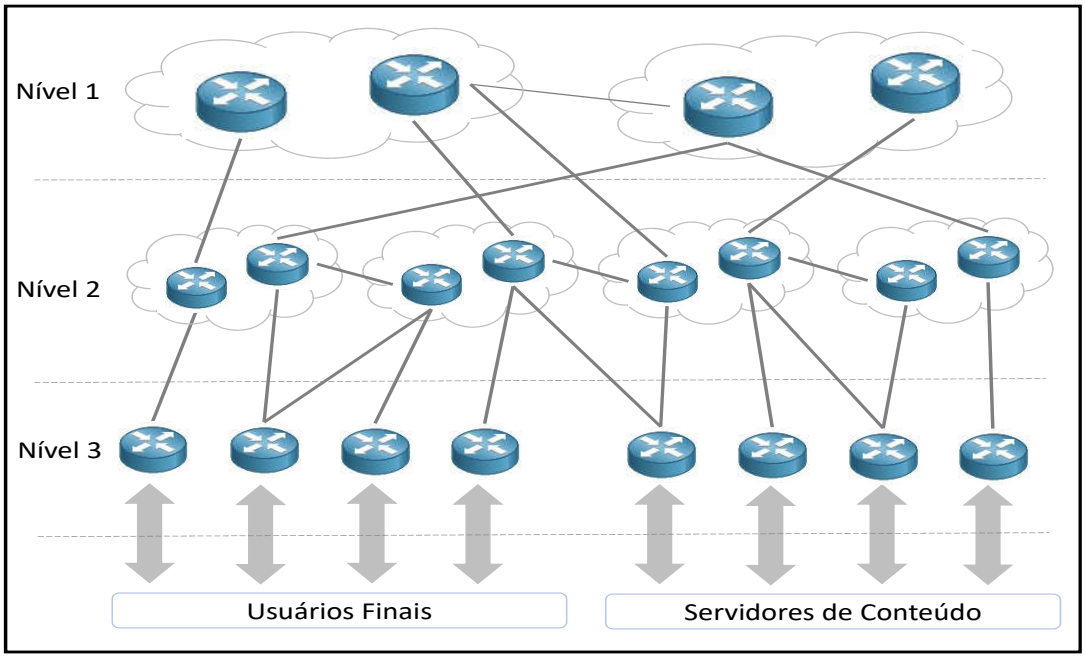
Scott Jordan (2007) utiliza o termo Neutralidade de Rede para caracterizar o tratamento igualitário de aplicações, conteúdos e serviços que trafegam pela internet e a importância da transparência nas práticas de gerenciamento de tráfego. Uma outra definição utilizada pelo Órgão Regulador da União Europeia (2011) afirma que a neutralidade de rede é um princípio a partir do qual todos os pacotes são tratados de forma igual através de uma infraestrutura *Internet Protocol* (IP).

O conceito de Barbara Schewick (2012) pressupõe que a inteligência das redes está localizada em suas extremidades (borda), isto é, nos computadores dos usuários finais, e não em seu centro (núcleo). Essa afirmação vai ao encontro deste trabalho por tratar do tema da quebra da neutralidade na borda. A topologia da Internet é hierárquica, onde os usuário são conectados a ISPs locais (borda), que por sua vez são conectados a ISPs regionais/nacionais, e por fim estes últimos são conectados a ISPs internacionais. A Figura 1, ilustra a topologia da Internet e seus níveis.

A quebra de neutralidade é comum, seja por interesses comerciais, seja por interesses de controle estatal, como acontece, nesse último caso, em países como China e Irã (MARSDEN, 2011). O Estados Unidos tem uma situação clara de cisão entre a banda larga fixa e a móvel.

Enquanto a banda larga fixa norte-americana teve alguma garantia de neutralidade até recentemente, a móvel já se apresentava explicitamente não neutra, com gerenciamento de dados e diferenciação de conteúdos de maneira bastante sensível, incluindo favorecimento de parcerias comerciais com as empresas fornecedoras do acesso.

Figura 1 – Topologia da Internet



Fonte: acervo pessoal.

As configurações impostas pelos ISPs em seus equipamentos são em sua grande maioria inacessíveis. Entretanto, pesquisa científicas (VAN SCHEWICK, 2012), (WU, 2003), (KANUPARTHY; DOVROLIS, 2010), (TARIQ et al., 2009) e (ZHANG; MAO; ZHANG, 2009) que os ISPs utilizam mecanismos de discriminação quando os pacotes trafegam através de suas infraestruturas de comutação, possibilitando assim a diferenciação e a priorização de tráfego. Por exemplo, ISPs podem configurar seus equipamentos para discriminar o serviço de vídeo sob-demanda simplesmente diminuindo a prioridade de pacotes que tenham como origem os servidores deste serviço.

Esse tipo de prática não é algo novo e ocorre também em outros países. Desde o início dos anos 2000, quando a Internet entra em uma fase de maior expansão no Brasil, já existiam serviços que praticavam o *traffic shaping* (SANTOS et al., 2016). Apesar dos menores índices, a

prática ainda é adotada por algumas empresas no mundo, conforme pesquisas do Measurement-Lab (ZHANG; MARA; ARGYRAKI, 2014).

Nos Estados Unidos, a *Federal Communications Commission* (FCC) aprovou a chamada *Open Internet Order* em fevereiro de 2015 (COMMISSION; COMMISSION et al., 2015). Um documento que substitui regras anteriores, onde aplica o princípio da neutralidade da rede a serviços de banda larga. O documento enfatiza claramente as situações de bloqueios, limitações e prioridades.

a) **Bloqueio** – ISPs não podem bloquear conteúdo ou serviços que estejam dentro da legalidade;

b) **Limitação** – Os ISPs não podem realizar degradação de tráfego de conteúdo, serviços ou por dispositivos;

c) **Priorização** – É vedada a priorização de conteúdo ou serviços legais sobre outros, dessa forma, não são criadas *fast lanes* [1] e os ISPs ficam impossibilitados de priorizar conteúdo de seus clientes.

Um outro ponto a ser discutido na Neutralidade incide sobre QoS implementada pelos ISP's. A simples configuração de QoS em um ISP pode ser considerada uma violação da neutralidade, porque pode se tratar de discriminação do tráfego de dados em seus equipamentos. Porém, para a estabilidade da Internet os ISP's precisam configurar QoS em seus roteadores (VAN SCHEWICK, 2015). Sobre o papel importante e indispensável que a Qualidade de Serviço tem na Internet, Barbara Schewick (2015) afirma que existe QoS do bem e QoS do mal, o que diferencia é o objetivo adotado pelo ISP.

Desde que foi sancionada a lei do Marco Civil da Internet, os ISPs estão proibidos de exercer qualquer discriminação de dados em suas redes. Desta forma, todos os dados são iguais e não podem ser diferenciados. Entretanto existem exceções descritas na lei, entre elas estão: (i) priorização de serviços de emergências; (ii) questões de tratamento de segurança da rede; (iii) situações excepcionais de congestionamento de rede. Quando houver a diferenciação do tráfego, excluídas as exceções mencionadas, então haverá a quebra da neutralidade.

[1] rotas preferenciais que priorizariam tráfego para serviços específicos de Internet

2.2 QUALIDADE DE SERVIÇO

A avaliação de desempenho em QoS é importante tanto para medir a capacidade da rede de deslocar os dados, como para os ISPs poderem gerir melhor as aplicações.

A implementação de QoS envolve dois aspectos: mecanismos de controle de tráfego implementados nos roteadores e uma metodologia de QoS, que define como os mecanismos são utilizados (JAMHOUR, 2012).

Os mecanismos de QoS alteram a forma de encaminhamento de pacotes pelos roteadores. Por padrão, a forma adotada pelos roteadores é o primeiro a entrar é o primeiro a sair (*First In First Out*-FIFO). A qualidade de serviço pode ser medida através das garantias que a rede oferece para o tráfego transportado. Originalmente, o único modo de operação suportado pelo IP era o melhor esforço (*best effort*).

As metodologias de QoS , definem como os diversos elementos da rede devem cooperar de forma a prover garantias fim-a-fim para o tráfego dos usuários. As metodologias atualmente propostas pelo IETF (*Internet Engineering Task Force*) são serviços integrados, serviços diferenciados e MPLS (*MultiProtocol Label Switching*).

a) **Serviço Integrado (IntServ)**: é uma estratégia de controle de qualidade por fluxo, isto é, cada aplicação de um usuário da rede. Esse modelo introduziu um protocolo de sinalização para QoS denominado RSVP (*Resource Reservation Protocol*). O método de serviços Integrados foi considerado complexo demais para ser adotado em grande escala na Internet;

b) **Serviços Diferenciados (DiffServ)**: é uma abordagem mais simples, que necessita de uma implementação mais simples de QoS. Nessa estratégia, existe tratamento diferenciado apenas para algumas grandes classes de tráfego. As classes do DiffServ recebem nomes padronizados como "EF (*expedited forwarding*)", "AF (*assured forwarding*)"e "BE (*best effort*)". Cada uma dessas classes é tratada de forma diferenciada pelos roteadores IP. O tráfego individualizado dos usuários é classificado nessas grandes classes. Essa metodologia não usa protocolo de sinalização;

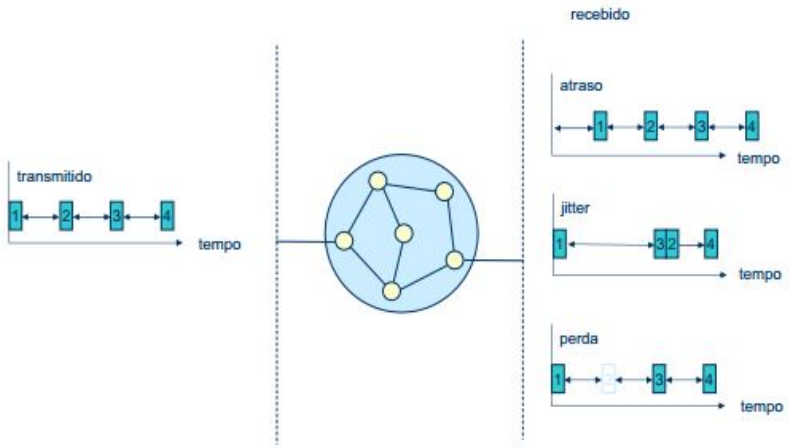
c) **MPLS (*MultiProtocol Label Switching*):** o MPLS foi considerado como uma alternativa para acelerar o processo de roteamento, que era considerado muito complexo nas redes IP. Contudo, nos últimos anos, protocolos de sinalização como o RSVP-TE e o CR-LDP

foram introduzidos, tornando o MPLS uma importante ferramenta para engenharia de tráfego. Através do MPLS, é possível definir caminhos com banda reservada ao longo dos roteadores. Isso tornou o MPLS uma das tecnologias mais importantes para criar circuitos virtuais comercializáveis.

Muitas vezes, as questões de QoS são um dos tópicos mais discutidos na relação ISPs-clientes. Até porque existem fatores determinantes nos serviços disponibilizados pelos ISPs, como exemplo o *throughput*, latência e a perda de pacotes. Esses fatores podem ser considerados fundamentais na Qualidade de Experiência (QoE) do usuário.

O QoS pode ser caracterizado por um conjunto de métricas de desempenho de rede, Esse trabalho tem ênfase para as métricas de Latência, *Jitter*, *Throughput* e Perda de Pacotes (BRADNER; MCQUAID, 1999). A Figura 2, ilustra três métricas de desempenho.

Figura 2 – Métricas de desempenho



Fonte: Jamhour, 2012.

a) Latência: ou atraso de ida e volta é o tempo total que um pacote trafega desde a origem até o destino e retorna para origem. Para medir a latência, o pacote possui uma marca de tempo (*timestamp*) que é transmitido pela rede. A marca de tempo é então analisada quando o quadro é recebido. Para que isso aconteça, o quadro precisa voltar ao local de origem por um laço de retorno (atraso ida e volta). Com uma grande latência pode haver perda de sincronização;

b) *Jitter*: é variação de tempo entre as chegadas de pacotes do endereço de origem caracteriza-se como variação de latência, ou simplesmente *jitter*. Se o tempo de chegada dos pacotes variar de forma muito diferente, diminuirá a qualidade do serviço;

c) *Throughput*: também conhecido como vazão, é simplesmente a quantidade máxima de dados, que pode ser transportada da origem ao destino por unidade de tempo. No entanto, a definição e medição da vazão são complicadas pela necessidade de se definir um nível aceitável de qualidade;

d) Perda de Pacotes: analisa o número de quadros que foram transmitidos pelo transmissor e que nunca foram recebidos em seu destino. Os pacotes podem ser perdidos ou descartados por diversas razões, incluindo erros e atraso excessivo. Uma perda de vários pacotes em uma fila afetará a qualidade, pois irá impor uma retransmissão de pacotes, retardando ainda mais a rede.

2.3 ANOMALIAS DE TRÁFEGO

O comportamento de rede é composto por uma sequência de pacotes que são trocados entre dispositivos conectados à rede. O tráfego de rede pode ser definido como normal ou anômalo (DEWAELE et al., 2007) :

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

Anomalias de tráfego podem causar graves problemas no desempenho da rede. Consequentemente estes problemas afetam a Qualidade de Serviço e Qualidade de Experiência dos usuários. Estas anomalias, são definidas como um desvio acentuado de uma determinada característica do tráfego relativo a um modelo de comportamento normal (LAKHINA; CROVELLA; DIOT, 2004). Se tratando de Neutralidade de Rede a anomalia está relacionada a diferenciação de tráfego que afetará o desempenho e a qualidade de serviço da rede.

A proposta deste trabalho na construção de um método que possa identificar a diferenciação de um tráfego é muito semelhante a um sistema de detecção de intrusão. Afirmar-se isso, porque o método tem função principal detectar alguma anomalia no tráfego de dados, assim como um IDS.

Sistemas de Detecção de Intrusão (*Intrusion Detection System - IDS*) são dispositivos implementados em *software* e *hardware* responsáveis por detectar e alertar algum ponto da rede a presença de tráfego proveniente de uma ação maliciosa (AXELSSON, 2000).

A detecção baseada em anomalias tem sido amplamente estudada pela comunidade científica. Isso pode ser visto pela extensa revisão feita pelos autores em (AHMED; MAHMOOD;

HU, 2016). Considerando-se que há várias abordagens que podem ser utilizadas em conjunto entre si, a natureza de uma detecção deve considerar o comportamento do tráfego que está sendo analisado para realizar uma posterior comparação com uma base de assinaturas ou por um desvio de comportamento (ZHANG et al., 2015).

2.4 RESUMO

O capítulo apresentou as definições e conceitos existentes sobre Neutralidade de Rede, Qualidade de Serviço e Anomalias de Tráfego. Sobre neutralidade foi descrito seus princípios a partir de três autores de relevância no tema. Também foi exposto a existência da quebra da neutralidade e seus motivos, exemplificou-se como alguns países tratam o tema. Bem como, retratou-se o funcionamento dos ISPs na Internet em sua estrutura atual de comunicação de dados descrevendo a borda e o núcleo. Por fim, foi tratado sobre o princípio da neutralidade sobre situações específicas de bloqueio, limitação e priorização.

Em relação a qualidade de serviço, tratou-se da importância da avaliação de desempenho para as aplicações do ISPs. Também foi conceituada as metodologias propostas pelo IETF. Por fim, apresentou-se qualdidade de serviço que é caracterizado por um conjunto de métricas de desempenho de rede. Sobre as quais, definiu-se para esta pesquisa as métricas de latência, *jitter*, *throughput* e perda de pacotes.

Finalizando, a anomalia de tráfego neste trabalho é considerada tráfego não-neutro aquele que possuir anomalia ou interrupção de segmento da rede. Em contra partida um tráfego neutro aquele que possuir um tráfego legítimo, sem a ocorrência de ataques ou anomalias. Por se tratar de neutralidade a anomalia de tráfego é tratada como diferenciação de tráfego, pois influciará o desempenho e a qualidade de serviço.

3 REVISÃO DA LITERATURA

Neste Capítulo são apresentados os principais trabalhos da área sobre métodos de identificação da quebra da neutralidade na Internet. Também são descritas as vantagens e desvantagens de cada trabalho. Por fim, uma discussão sobre os trabalhos e suas características.

3.1 TRABALHOS RELACIONADOS

Vários pesquisadores têm se dedicado a propor trabalhos com as mais diversas abordagens (*e.g.*, medição passiva ou ativa, análise da borda ou núcleo, tempo dos testes, periodicidade dos testes entre outras características) sobre detecção da quebra da neutralidade de rede. Essas abordagens muitas vezes são específicas para um cenário ou não contemplam ao mesmo tempo diversas métricas de desempenho de rede: latência, *jitter*, *throughput* e perda de pacotes.

O trabalho desenvolvido por Mukarram Tariq (2009) apresenta uma plataforma de medição distribuída para detectar se um determinado ISP induz a degradação do desempenho para classes específicas de serviço. A medição do trabalho proposto é passiva, onde o tráfego gerado pelo usuário é monitorado continuamente e envia periodicamente informações para o servidor. Para tal fim os autores apresentaram um cenário com três avaliações diferentes: (i) discriminação simples em tráfego *Hypertext Transfer Protocol* (HTTP), (ii) discriminação em fluxo longo de tráfego HTTP e (iii) discriminação de tráfego BitTorrent. Os resultados obtidos após os testes confirmam a detecção da quebra da neutralidade exclusivamente nos cenários propostos. Entretanto, a plataforma de Tariq (2009) precisa conhecer a política de discriminação do ISP. Caso contrário, ela não será eficiente na detecção da discriminação do tráfego.

O trabalho de Zhang (2009) apresenta um sistema que detecta a diferenciação de tráfego com base em informações de roteamento, cabeçalho do pacote e informações da camada de aplicação. O sistema proposto pelos autores se baseia em uma seleção de caminho inteligente para detectar tanto a diferenciação de conteúdo e de roteamento. O sistema publicado pelos autores relata testes com cinco aplicações diferentes: HTTP, BitTorrent, SMTP, PPLive e VoIP. Os testes utilizam sondagem ativa que realiza a detecção a partir da borda da Internet. Com os dados obtidos a partir da sondagem, eles são comparados pelas taxas de perdas agregadas de diferentes fluxos para inferir a presença de violações de neutralidade de rede em ISP. O sistema proposto por Ying Zhang (2009) está limitado a detectar a quebra da neutralidade, através da

análise da perda de pacotes com duração de duas horas para a realização dos testes.

O trabalho de Dischinger (2010) apresenta uma ferramenta onde o usuário final se conecta ao servidor Web da implementação para executar testes, que geram fluxos contendo dados de nível de aplicativo. Dessa forma é executada a medição entre o caminho do cliente e o servidor. A ferramenta realiza a diferenciação do tráfego com base nos tipos de fluxos, caracterizando diferentes fluxos através do cabeçalho IP (IP de origem e IP de destino), cabeçalho TCP (porta de origem e porta de destino) e pela carga útil do pacote. Neste trabalho os autores não conseguem detectar a discriminação entre os provedores de conteúdo e a existência de problemas de rede como congestionamento. A ferramenta desenvolvida pelos autores não está mais disponível para uso desde fevereiro de 2017.

O método apresentado por Kanuparthy e Dovrolis (2010) visa detectar se o ISP realiza discriminação de tráfego através das métricas de perda e de atraso de pacotes. A ferramenta desenvolvida pelos autores gera dois fluxos, sendo um fluxo classificado como de prioridade baixa, e outro como prioridade normal de tráfego. O trabalho proposto realiza medições ativas de máquinas clientes para os enlaces na plataforma da M-LAB[2] para detectar qualquer tipo de discriminação com base em mecanismos de tráfego. Desta forma detecta-se o tráfego de diferenciação com base na gestão de fila (*e.g.*, WFQ e RED). O contraponto, é que dois fluxos simultâneos (HTTP e BitTorrent) podem ser diferentes na taxa de perda sendo do mesmo tamanho. Outra desvantagem, é que o método utiliza respostas ICMP e a geração de tais pacotes está sujeita a taxa de limitação.

Outro método para detecção de discriminação de tráfego foi desenvolvido por Basso (2011) que apresenta uma ferramenta para medições de rede distribuída, onde um agente monitora periodicamente a latência e *throughput* da conexão do usuário e armazena os resultados em um servidor centralizado. Atualmente a ferramenta proposta suporta o teste de velocidade e teste de BitTorrent, de maneira a emular o tráfego HTTP e BitTorrent respectivamente. Uma característica importante descrita no trabalho, é o fato da ferramenta monitorar continuamente a conexão do usuário final. No entanto, a variação no desempenho pode ser o resultado de outros fatores, tais como congestionamento da rede, e não é sempre devido à discriminação do ISP. Este trabalho não é capaz de identificar a diferenciação de tráfego por aplicação, ele se detêm na diferenciação por protocolo ou por usuário.

Finalmente o trabalho desenvolvido por Ravaioli (2012), onde se reproduz o tráfego real

[2]é uma plataforma de servidores aberta e distribuída para pesquisadores desenvolverem ferramentas de medição da internet. Disponível em <https://www.measurementlab.net/>

do usuário de forma que este tráfego atinja apenas os roteadores a poucos saltos de distância do ISP do cliente. Neste método são realizadas medições de atraso e perda de pacotes para cada fluxo. Os autores afirmam que o método de medição e detecção proposto é independente de aplicações e das configurações de discriminação de tráfego utilizados pelo ISP. As medições de atraso e perda de pacotes utilizadas pela ferramenta proposta (RAVAIOLI; BARAKAT; URVOY-KELLER, 2012) são referentes a respostas ICMP: o atraso é o RTT entre envio do pacote e o recebimento da resposta ICMP. A perda de pacotes corresponde à taxa de respostas ICMP não recebidas. A desvantagem do método está relacionada com sua *base-line* que é fundamentada em um ambiente controlado, além também do método não estabelecer uma relação entre os fluxos que compõe o tráfego analisado.

3.2 DISCUSSÃO

Com o objetivo de melhor analisar as pesquisas sobre trabalhos que tratam de métodos de identificação da quebra da neutralidade, agrupou-se os trabalhos em decorrência das suas características. Além disso, esta analise auxilia na definição de detalhes de implementação do método proposto nesta dissertação.

Em relação ao tipo de medição que os trabalhos realizam, (DISCHINGER et al., 2010), (ZHANG; MAO; ZHANG, 2009), (BASSO; SERVETTI; DE MARTIN, 2011), (TARIQ et al., 2009) e (KANUPARTHY; DOVROLIS, 2010) efetuam sua medição na borda da Internet, isto é, entre o usuário e o ISP. Entretando, o trabalho de (RAVAIOLI; BARAKAT; URVOY-KELLER, 2012) realiza medições desde o primeiro ISP até chegar ao ISP final da conexão, sendo considerando assim como medição de núcleo.

Quanto a abordagem na qual ocorre a coleta das informações, (BASSO; SERVETTI; DE MARTIN, 2011) e (TARIQ et al., 2009) atuam de maneira passiva, ou seja, usam um *sniffer* de rede para capturar os dados. Ao contrário dos trabalho de (DISCHINGER et al., 2010), (KANUPARTHY; DOVROLIS, 2010) e (ZHANG; MAO; ZHANG, 2009) que realizam a captura de maneira ativa, inserindo fluxos para medição posterior. Somente o trabalho de (RAVAIOLI; BARAKAT; URVOY-KELLER, 2012) utilizada abordagem híbrida para obtenção dos resultados, isto é, utilizando a maneira passiva e ativa para medição dos fluxos.

Sobre as métricas de desempenho utilizadas para inferir a quebra da neutralidade, (DISCHINGER et al., 2010) e (ZHANG; MAO; ZHANG, 2009) usam apenas uma métrica para informar se ocorre ou não quebra da neutralidade. Outros trabalhos utilizam duas métricas de

desempenho para obtenção dos resultados que permitam identificar ou não a quebra da neutralidade (KANUPARTHY; DOVROLIS, 2010) (BASSO; SERVETTI; DE MARTIN, 2011) (TARIQ et al., 2009) e (RAVAIOLI; BARAKAT; URVOY-KELLER, 2012).

A cerca dos métodos apresentados nos trabalhos relacionados, Dischinger (2010) utiliza o *throughput* calculando o valor mínimo, máximo e a mediana das taxas de transferências medidas. O trabalho de Zhang (2009) utiliza do método Kolmogorov-Smirnov (KS) para comparar conjunto de dados para aplicação HTTP com outros conjuntos de dados. O trabalho de Kanuparthy (2010) utiliza o método de divergência de Kullback-Leibler como uma medida de igualdade de distribuições e a ferramenta de comparação entre fluxos foi o Teste Z. A aplicação do método estatístico de variáveis de confusão é utilizada no trabalho de Tariq (2009) que compara a mesma aplicação em ISPs diferentes levando em conta uma série de fatores. Ravaioli (2012) em seu trabalho utiliza uma análise estatística simples para inferir se um fluxo de dados sofreu diferenciação de tráfego. Compara-se as métricas de cada fluxo com as medidas de todo o fluxo. Os valores são obtidos através do campo *Time to Live* (TTL) em conjunto com as respostas do *Internet Control Message Protocol* (ICMP).

Outra análise, refere-se ao trabalho de Basso (2011), onde usa-se um método mais simples para obtenção dos resultados. Neste trabalho são analisados os valores obtidos pelo cliente, com outras medições passada de outras conexões e também com outros valores existentes no servidor de medição desenvolvido no trabalho.

Com o objetivo de demonstrar melhor as características de solução de cada trabalho relacionado sobre quebra da neutralidade, desenvolveu-se um resumo comparativo entre eles, conforme Tabela 1. Para cada solução a Tabela destaca: abordagem utilizada, tipo de medição, a métrica empregada e que tipo de comparação é realizada.

a) Solução: nome da solução desenvolvida nos trabalhos relacionados para diagnósticar a quebra da neutralidade;

b) Abordagem: ativa quando se injeta pacotes de testes no fluxo adicionando tráfego, ou passiva que captura o fluxo através de *sniffer* de rede;

c) Medição: de onde é realizada a leitura do tráfego, borda da Internet ou no núcleo;

d) Métrica: quais são as métricas de desempenho que são avaliadas pelos métodos;

e) Comparação: os dados dos fluxos são comparados entre o próprio tráfego ou com uma *base-line*.

Tabela 1 – Comparação entre as ferramentas

Referência	Ferramenta	Abordagem	Medição	Métrica	Comparação
Tariq (2009)	NANO	Passiva	Borda da Internet	Conforme a aplicação	Mesma aplicação em ISPs diferentes
Zhang (2009)	NetPolice	Ativa	Núcleo da Internet	Perda de pacotes	Vários protocolos X HTTP
Dischinger (2010)	Glasnost	Ativa	Borda da Internet	*Throughput*	Aplicação X Dados aleatórios
Kanuparthy (2010)	DiffProbe	Ativa	Borda da Internet	Atraso e perda de pacotes	VoIP X Dados aleatórios
Basso (2011)	Neubot	Passiva	Borda da Internet	Latência e *Througput*	Com ele mesmo e outras medições próximas geograficamente
Ravaioli (2012)	ChkDiff	Híbrida	Núcleo da Internet	Atraso e Perda de pacotes	Várias aplicações X HTTP

Fonte Autor.

Com base nas informações obtidas na leitura dos trabalhos relacionados realizou-se uma comparação entre eles com o intuito de melhor observar as definições sobre as soluções propostas e suas características.

Chegou-se à conclusão de que a abordagem passiva observa o tráfego de dados mais legítimos para descobrir a diferenciação entre os fluxos, enquanto a abordagem ativa influencia negativamente no desempenho da rede por introduzir fluxos artificiais para realizar as medições. Entretanto, a abordagem passiva através de observações não garante que a aplicação está configurada para recuperar o desempenho máximo da rede. Apesar do ponto negativo da abordagem ativa, esta foi a mais utilizada entre os trabalhos pesquisados. O principal motivo é a característica de disparar um fluxo para um servidor de monitoramento e para o serviço que se deseja avaliar a neutralidade para comparar os resultados obtidos.

Outra definição é o local de medição para estabelecer a existência da quebra da neutralidade. A borda da Internet é o local mais utilizado pelos trabalhos, pois é onde ocorre a maioria das diferenciações de tráfego, consequemente a quebra da neutralidade é provocada pelos ISPs mais próximos ao usuário. Entretanto, existe um trabalho que pesquisou a discriminação realizada no núcleo da Internet, no qual relata que é a mais difícil de ser identificada por envolver uma complexa topologia de comunicação entre os ISPs.

Em relação às métricas, fica nítida a utilização de no máximo duas delas para aferir a diferenciação de tráfego. Em nenhum trabalho utilizou-se múltiplas métricas de desempenho correlacionadas para identificar a quebra da neutralidade. Outra característica dos trabalhos relacionados está no fato de nenhum deles utilizar a métrica *jitter* como fator na detecção na diferenciação de tráfego.

Um aspecto importante e semelhante entre todos os trabalhos relacionados é a inexistênica de solução que identifica o exato momento que está ocorrendo a diferenciação de tráfego, assim como qual métrica de desempenho está em degradação e ocasionando tal fator. Os trabalhos se limitam a informar se ocorre ou não a quebra da neutralidade de maneira genérica, sem especificar quando começou e por quanto tempo durou.

4 SOLUÇÃO PROPOSTA

O objetivo deste capítulo é apresentar a solução proposta que foi criada com base na revisão da literatura, bem como observando as melhores práticas existentes nas ferramentas disponíveis. Sendo assim, são apresentados detalhadamente o processo de análise do tráfego, a proposta de método deste trabalho, além dos cálculos dos índices criados para verificação da quebra da neutralidade.

4.1 PROCESSO DE ANÁLISE DO TRÁFEGO

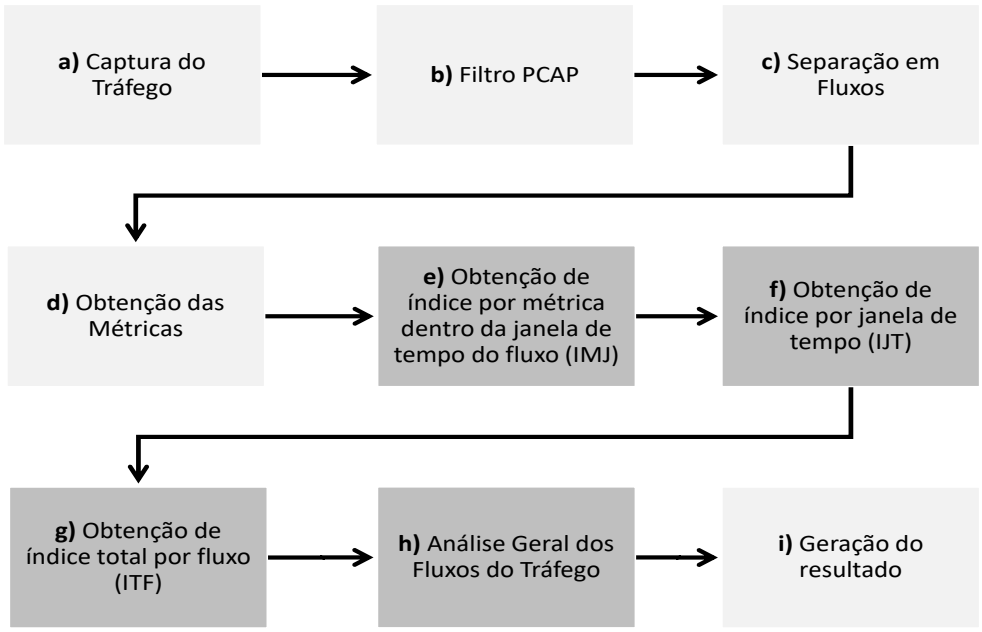
O método proposto tem como objetivo principal detectar a quebra da neutralidade de maneira agnóstica, ou seja, ser capaz de identificar a quebra da neutralidade independente do conteúdo, da aplicação, dos serviços ou do dispositivo de destino. Nesse trabalho define-se fluxo como uma sequência de pacotes que se enquadram em um determinado critério, correspondente a comunicação entre duas entidades da rede (CLAFFY; BRAUN; POLYZOS, 1995). Para formação de um fluxo definiu-se 6 campos do cabeçalho dos pacotes na definição de um fluxo: o endereço IP de origem, o endereço IP de destino, a porta de origem, a porta de destino, o protocolo da camada de transporte e o protocolo de aplicação.

A caracterização do tráfego da Internet é fundamental para várias atividades, como gerenciamento de rede, planejamento e provisionamento, engenharia de tráfego, diagnóstico, desempenho da aplicação e detecção de anomalias (CALLADO et al., 2009). As estratégias de medições podem ser vistas como uma ferramenta essencial para identificar comportamentos anômalos (*e.g.*, *Denial of Service* - DoS, problemas de roteamento e tráfego indesejado) (CALLADO et al., 2009).

Ressalta-se que avaliar a diferenciação de tráfego com o objetivo de diagnosticar a quebra da neutralidade pode ser considerado um fator relevante, ainda mais quando relacionados com as métricas de desempenho. Esta avaliação da FCC (2011) está totalmente integrada à proposta deste trabalho que utiliza diferentes métricas de desempelho (latência, *jitter*, *throughput* e perda de pacotes) para identificar a quebra da neutralidade.

O processo de análise do tráfego apresentado neste trabalho é composto por nove etapas, conforme estrutura apresentada na Figura 3. Dentre essas etapas, quatro são as mais importantes e onde de fato a detecção ocorre.

Figura 3 – Processos da análise do tráfego



Fonte: acervo pessoal.

a) **Captura do Tráfego:** ocorre durante a comunicação entre a máquina do cliente (*Client*) e os servidores (*Servers*), é executado o *sniffer* de rede e gerado um PCAP de todo tráfego capturado;

b) **Filtro PCAP:** o arquivo gerado que inclui todo tráfego é filtrado, permanecendo somente informações relevantes para analise. São descartados os pacotes que fazem a comunicação inicial do gerador de tráfego, assim como os primeiros pacotes que fazem a sincronização de cada fluxo;

c) **Separação em Fluxos:** com o PCAP filtrado, separa-se todo o tráfego em fluxos (endereço IP de origem, endereço IP de destino, porta de origem, porta de destino, protocolo da camada de transporte e protocolo de aplicação);

d) **Obtenção das Métricas:** calcula-se as métricas de latência, *jitter*, *throughput* e perda de pacotes de cada fluxo. Também calcula-se as mesmas métricas para os fluxos separados em janelas de tempo correspondentes a um segundo;

e) **Obtenção de índice por métrica dentro da janela de tempo do fluxo (IMJ):** com as métricas obtidas na etapa anterior, é calculado um índice que representa a neutralidade ou a não neutralidade para cada métrica do fluxo em cada janela de tempo;

f) **Obtenção de índice por janela de tempo (IJT):** é a soma dos índices das métricas obtidos na etapa anterior divido pelo número de métricas;

g) **Obtenção de índice total por fluxo (ITF):** é a soma dos índices que são diferentes de 0 (zero) dividido pelo número de janelas. Com o resultado obtido neste cálculo, é possível saber a porcentagem de janelas de tempo no qual foram identificadas uma diferenciação de tráfego;

h) **Análise geral dos fluxos do tráfego:** nessa etapa os resultados de cada fluxo são analisados e correlacionados com outros fluxos dentro da mesma janela de tempo. Assim, tem-se a possibilidade de verificar se uma mesma janela de tempo para fluxos diferentes possui comportamento semelhante ou não;

i) **Geração do Resultado:** Depois de todas as etapas concluídas é possível indicar ou não a quebra da neutralidade no tráfego. O método informa em qual fluxo houve a degradação do tráfego, assim como quais métricas sofreram diferenciação e em qual janela de tempo ocorre a anormalidade do fluxo.

Dentre as noves etapas apresentadas, o método proposto por este trabalho compreende apenas um sub-conjunto de 4 etapas, que são as principais.

4.2 PROPOSTA DO MÉTODO AGNÓSTICO

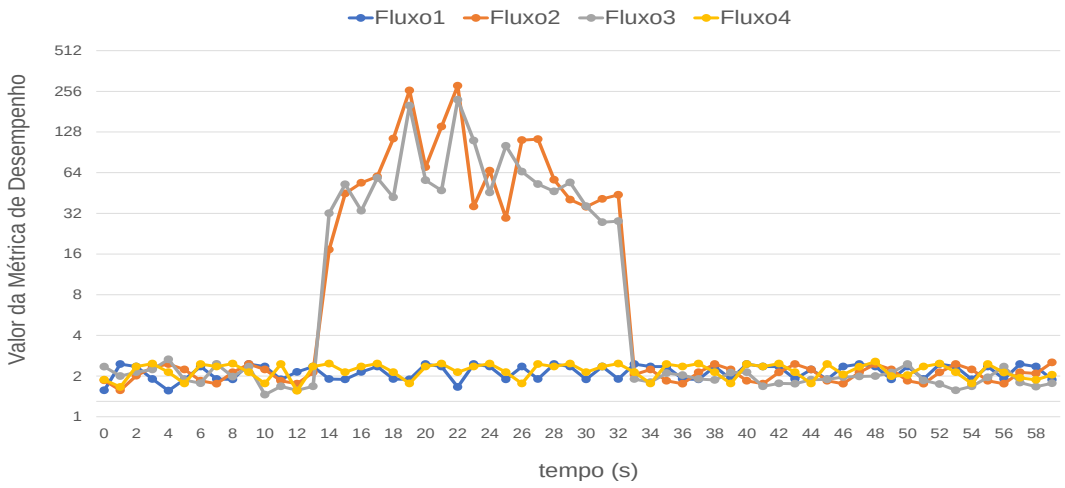
O método possui 4 etapas principais (e, f, g e h) que consistem em analisar as métricas de desempenho do tráfego de dados capturados, separar esse tráfego em fluxos diferentes, separar os fluxos em janelas de tempo e comparar as medidas obtidas entre fluxos. Para cada fluxo e janela de tempo obtém-se os valores das métricas de latência, *jitter*, *throughput* e de perda de pacotes. Dentro de cada fluxo haverá uma segmentação em janelas temporais de igual duração. O número de janelas por fluxo dependerá do tempo total de tráfego capturado. Nos testes adotou-se 60 segundos como tempo de amostragem nos experimentos.

A ideia de usar índices para identificar fluxos idealizou-se a partir do trabalho de Zhang, Mara e Argyraki (ZHANG; MARA; ARGYRAKI, 2014) onde definiram o índice 1 (um) para

tráfego diferente, e 0 (zero) para tráfego semelhante. Os autores do artigo também descrevem que a violação da neutralidade é tratada como a situação na qual os fluxos de dados entre dois equipamentos possuem desempenhos diferentes ao atravessar o mesmo *link* de rede.

Neste trabalho avalia-se a diferenciação de tráfego que acontece sobre os fluxos de um determinado protocolo. Assim sendo, a diferenciação de tráfego age exclusivamente sobre um determinado protocolo, porém, em fluxos diferentes conforme Figura 4. Entretando, se ocorrer degradação todos os fluxos sofrem comportamentos semelhantes no mesmo período de tempo conforme ilustra a Figura 5.

Figura 4 – Tráfego com diferenciação



Fonte: acervo pessoal.

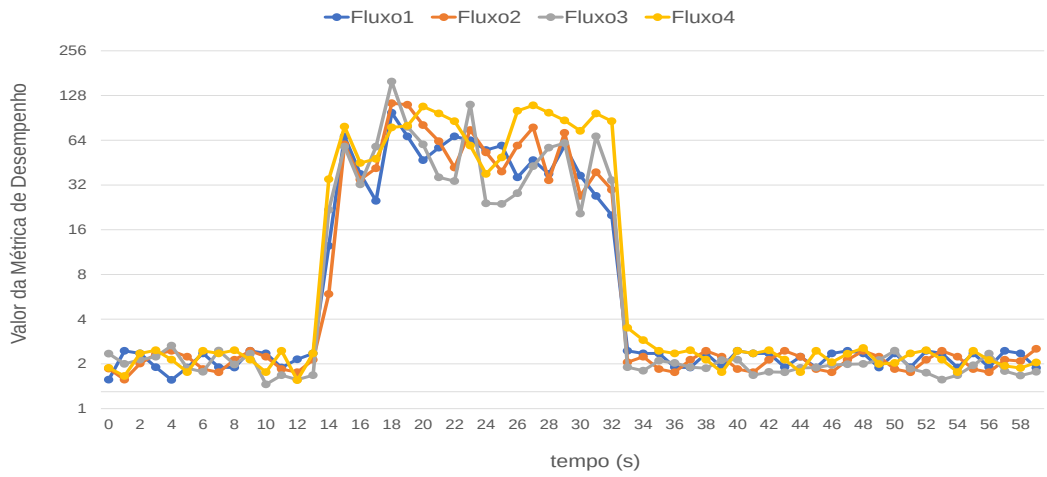
4.2.1 Obtenção de Índice por Métrica dentro da Janela de Tempo do Fluxo (IMJ)

O processo de obtenção do índice por métrica dentro da janela de tempo consiste em transformar o valor de cada métrica de desempenho em um índice (0 ou 1), no qual o valor 0 (zero) representa fluxo de dados neutro e o valor 1 (um) representa fluxo de dados não-neutro.

As fórmulas estatísticas para obtenção dos índices das métricas de latência, *jitter* e *throughput* são baseadas no Controle Estatístico de Processo (CEP), e a obtenção do índice da métrica de perda de pacotes é baseada em uma comparação simples entre um valor já estabelecido.

Conforme Montgomery (2000) e Robinson (2002) as técnicas de CEP são baseadas na construção de gráficos de controle. Um dos principais propósitos dos gráficos de controle é

Figura 5 – Tráfego com degradação



Fonte: acervo pessoal.

detectar ocorrências de mudança no processo, para que uma investigação da causa e uma ação corretiva possam ser tomadas da forma mais rápida possível (MONTGOMERY, 2000). Do mesmo modo, Mahajan (2004) relata em seu trabalho que o método CEP é uma técnica gráfica utilizada em ambientes que atuam com correlação de uma amostra com suas anteriores, e tem como objetivo de avaliar se a amostra atual está dentro de um comportamento normal baseado nos valores da totalidade das amostras passadas.

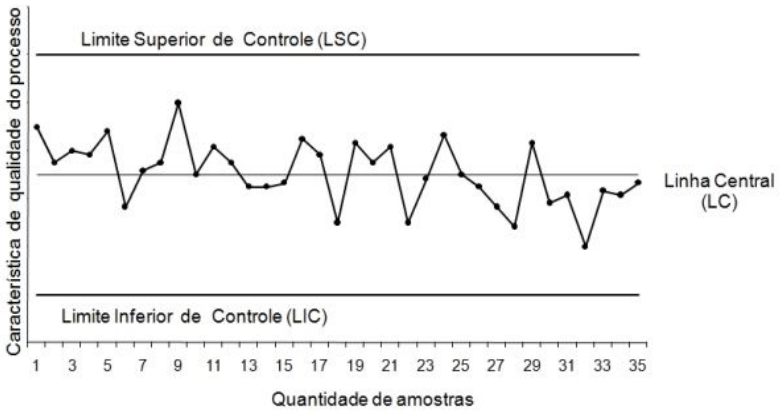
Os limites dos gráficos de controle, conforme Figura 6 são determinados com base na média μ, e no desvio padrão σ da distribuição da variável aleatória χ que representa uma medida de interesse sobre o processo monitorado. Os gráficos de controle são compostos por uma Linha Central (LC) e por linhas de Limite Superior de Controle (LSC) e Limite Inferior de Controle (LIC). A linha central representa um valor médio considerado normal para o processo monitorado.

Com as definições de LSC e LIC, é considerado fluxo normal se os valores das amostras estiverem dentro dos limites. Caso contrário, o processo é considerado anormal (COSTA; EPPRECHT; CARPINETTI, 2005) (MONTGOMERY, 2000). Portanto, o valor da amostra calculada que estiver dentro dos limites é considerado normal (neutro), e o valor que estiver fora dos limites é considerado anômalo (não-neutro).

a) **Cálculo do IMJ para latência e *jitter*:**

A fórmula para obtenção do IMJ das métricas de latência e *jitter* estabelece que as duas

Figura 6 – Exemplo de gráfico de controle padrão



Fonte: (Montgomery, 2009)

primeiras amostras (tempo 0 e tempo 1) são descartadas, não sendo consideradas para cálculo do IMJ. Consequentemente, o cálculo começa na amostra de tempo 2. A partir do valor da janela temporal 2 é calculada a média μ e o desvio padrão σ dos valores anteriores. Com esse resultado calcula-se a próxima amostra (tempo 3 segundos). Nesse cálculo adotamos um método simples de detecção de valores atípicos, muitas vezes utilizado no domínio do processo de controle de qualidade (MONTGOMERY, 2000), que consiste em assumir que todas as instâncias de dados anormais estão a uma distância da média μ , superior a 3σ ou inferiror a 3σ.

Amostras que estão fora dos limites estabelecidos são consideradas *outliers*. Nessas condições, a hipótese que guia a utilização dos gráficos de controle é que, em um processo estável, a grande maioria dos valores obtidos de uma característica de qualidade devem estar contidas no intervalo $\mu \pm 3\sigma$. Dessa forma, os limites de controle superior e inferior são usualmente escolhidos como: LSC = μ + 3σ e LIC = μ - 3σ.

$$\chi <= \mu + 3 * \sigma \qquad \in \qquad \chi >= \mu - 3 * \sigma \tag{4.1}$$

onde:

χ: valor da amostra atual

μ: valor da média da amostras anteriores

σ: valor do desvio padrão das amostras anteriores

Entretanto, a partir da amostra 4 (tempo 3 segundos) o algoritmo verifica se o resultado anterior foi considerado neutro (0) ou não-neutro (1). Se for neutro, o algoritmo procederá fazendo o mesmo cálculo. Porém, se valor da amostra anterior for considerada não-neutra, o algoritmo calcula o valor da amostra atual com o valor do desvio padrão das amostras anteriores que são neutros.

O algoritmo que realiza os cálculos para obtenção dos índices referentes as métricas de latência e *jitter*, pode ser visualizado no Algoritmo 1. Para cada amostra da métrica o método faz o cálculo e verifica se o valor da amostra se encontra entre os limites estabelecidos. Sendo verdadeira esta condição a função retorna 0, ou seja, valor neutro; se for falsa a condição, o retorno será 1, que representa valor não-neutro.

O cálculo da métrica de *jitter* é identico ao cálculo da métrica de latência, consequentemente o algoritmo do método segue os mesmos passos.

Algoritmo 1: Cálculo dos índices das métricas de latência e *jitter*

```
Entrada: vetor de amostras
Saída: Índice da comparação da amostra sendo 0 (neutro) ou 1 (não-neutro)
início
    μ <– calcula média das amostras anteriores ;
    σ <– calcula o desvio padrão das amostras anteriores ;
    repita
        leia índice atual;
        if (μ + 3*σ) <= valor da amostra atual >= (μ - 3*σ) then
            retorna 0;
        else
            retorna 1;
        end
    até fim do vetor;
fim
```

b) **Cálculo do IMJ do *throughput*.**

Para o cálculo da métrica de *throughput* também se utiliza o método CEP, entretanto, limita-se apenas à identificação de valores abaixo do limite inferior de controle (LIC). Sendo assim, quando a velocidade de transmissão de pacotes for maior que o LSC não será considerado um problema que afeta negativamente o tráfego de dados. Neste sentido, o índice da métrica de *throughput* tem a finalidade da verificar se a vazão dos dados sofreu uma diminuição de *bytes* por segundo.

O algoritmo 2 descreve como é realizado o cálculo para obtenção do índice da métrica de *throughput*. Para cada amostra da métrica o método faz o cálculo e verifica se o valor da amostra é maior que o limite estabelecido, sendo verdadeira esta condição a função retorna 0, ou seja, amostra neutra. Porém, se for falsa a condição, o valor de retorno será 1, que representa amostra não-neutra.

Algoritmo 2: Função do teste da métrica de *throughput*

```
Entrada: vetor de amostras
Saída: Índice da comparação da amostra sendo 0 (neutra) ou 1 (não-neutra)
início
    μ <– calcula média das amostras anteriores ;
    σ <– calcula o desvio padrão das amostras anteriores ;
    repita
        leia índice atual;
        if índice >= (μ - 3*σ) then
            retorna 0;
        else
            retorna 1;
        end
    até fim do vetor;
fim
```

c) **Cálculo do IMJ da perda de pacotes.**

Para obtenção do índice da métrica de perda de pacotes é utilizado um novo cálculo. Se a perda de pacotes for maior que 5% é considerado como fluxo não neutro dentro de janela de tempo correspondente. Entretando, se for menor ou igual a 5% será considerado como fluxo neutro. Adota-se tal parâmetro em relação à métrica de perda baseando-se no trabalho de Wille e Tenório (2014), onde a taxa de perda é considerada satisfatória pelo cliente quando menor que 3%, e considerada aceitável até 5%.

$$\chi <= 5\% \tag{4.2}$$

onde:

χ: valor da amostra atual

O Algoritmo 3 realiza o cálculo para obtenção dos índices da métrica de perda de pacotes. Para cada valor da amostra que satisfaz a equação 4.2, sendo verdadeira esta condição a

função retorna 0, ou seja, amostra neutra. Porém, se for falsa a condição, o valor de retorno será 1, que representa amostra não-neutra.

Algoritmo 3: Função do teste da métrica de perda de pacotes

```
Entrada: vetor de amostras
Saída: Índice da comparação da amostra sendo 0 (neutra) ou 1 (não-neutra)
início
    repita
        leia índice atual;
        if índice <= 5% then
            retorna 0;
        else
            retorna 1;
        end
    até fim do vetor;
fim
```

Após o cálculo dos índice de todas as métricas do fluxo, obtem-se o IMJ, conforme Tabela 2. Agora todos os valores das métricas passam a representar um índice que expressa a diferenciação de tráfego dentro de cada janela de tempo correspondente. A próxima etapa do método é a correlação entre as métricas dentro da mesma janela temporal.

Tabela 2 – Exemplo do IMJ calculado de todas as métricas por fluxo

Fluxo A											
	t 0	t 1	t 2	t 3	t 4	t 5	t 6	t 7	t 8	t 9	t 10
Latência	0	0	0	1	1	0	0	0	0	0	0
Jitter	0	0	1	1	1	0	0	1	0	0	0
Throughput	0	0	0	1	1	0	1	0	0	0	0
Perdas	0	0	0	1	1	0	1	1	0	0	0

Fonte. Autor

4.2.2 Obtenção de Índice por Janela de Tempo (IJT)

O IJT é relevante, pois é a etapa em que ocorre a correlação entre as métricas dentro do fluxo de dados. O valor é resultado da análise do mesmo instante de tempo de todas as métricas de desempenho. Esta fase do método representa a obtenção de um índice para cada janela temporal do fluxo. Assim, o índice expressa o quanto aquele instante de tempo do fluxo de dados é neutro ou não. O cálculo para obtenção do índice é a soma do IMJ de cada métrica divido por 4 (total de métricas de desempenho), conforme equação 4.3.

$$IJT(ti) = \frac{IMJ(lat) + IMJ(jit) + IMJ(thr) + IMJ(per)}{4} \tag{4.3}$$

O algoritmo 4, apresenta o cálculo do Índice por Janela de Tempo (IJT) que calcula a soma dos índices (0 ou 1) das métricas para cada janela de tempo das amostras.

Algoritmo 4: Cálculo do IJT

Entrada: Vetores Resultado das métricas (latência, *jitter*, *throughput* e perda)
Saída: Vetor Índice IJT
início
  repita
    vIJT[i] ← (vlatencia[i] + vjitter[i] + vthroughput[i] + vperda[i]) / 4 ;
  até *fim do vetor*;
fim

4.2.3 Obtenção de Índice Total por Fluxo (ITF)

Com os índices de cada janela de tempo, aplica-se um cálculo de média aritmética simples. O cálculo será o somatório de todos IJT que forem diferentes de 0 (zero) dividido pelo número de janelas do fluxo. Como exemplo, contabiliza-se apenas os valores que aparecem na Tabela 3. Com isso, o resultado do ITF será 0,325. O valor do ITF mais próximo de 1, representa a existência da quebra da neutralidade em mais janelas de tempo. E quando mais próximo de 0, significa que o fluxo de dados possui apenas poucas janelas com tráfego não-neutro.

Tabela 3 – Tabela de obtenção do ITF

Fluxo A											
	t 0	t 1	t 2	t 3	t 4	t 5	t 6	t 7	t 8	t 9	t 10
Latência	0	0	0	1	1	0	0	0	0	0	0
Jitter	0	0	1	1	1	0	0	1	0	0	0
Throughput	0	0	0	1	1	0	1	0	0	0	0
Perdas	0	0	0	1	1	0	1	1	0	0	0
IJT –>	0	0	0,25	1	1	0	0,5	0,5	0	0	0
ITF –>	0,325										

Fonte. Autor

4.2.4 Análise Total dos Fluxos do Tráfego

Nessa última etapa do método os resultados de cada fluxo são correlacionados entre os fluxos existentes para que possa-se fazer uma análise do tráfego como um todo, e não mais por fluxos isolados. O método informará em qual fluxo houve a diferenciação de tráfego, assim como quais métricas sofrem tal alteração, em qual a janela de tempo e a duração que aconteceu a quebra da neutralidade pelos fluxo de dados.

O Algoritmo 5, descreve o método de comparação entre a mesma janela de tempo de cada fluxo para aferir a quebra da neutralidade. Com esta comparação é possível identificar em quais fluxos, em qual o momento exato e por quanto tempo está ocorrendo a discriminação. O algorimto também é capaz de diferenciar a degradação da diferenciação entre fluxos, assim como informar se está acontecendo uma possível anomalia ou um congestionamento.

Algoritmo 5: Resultado final do método proposto

```
Entrada: IJT de cada fluxo
início
    aDeg <– armazena tempo da degradação
    aDif <– armazena tempo da diferenciação
    repita
        leia IJT atual;
        if IJT de todos os fluxos = 0 then
            retorna Tráfego Neutro;
        if IJT de todos os fluxos ≠ 0 then
            retorna Degradação no Tráfego;
            armazena posição da janela temporal;
        if IJT de todos os fluxos de mesmo protocolo ≠ 0 then
            retorna Diferenciação de Tráfego;
            armazena posição da janela temporal;
        else
            retorna tráfego com anomalia no fluxo ou congestionamento;
        end
    até fim do vetor;
    Resultado: Mostra o tempo que aconteceu a diferenciação e a degradação no
               tráfego
fim
```

Conforme a Tabela 4, onde apresenta-se dois fluxos (A e B), nota-se que o Fluxo A possui várias métricas de desempenho que apresentam tráfego não neutro. Porém, o Fluxo B, ao contrário do Fluxo A, apresenta quase na sua totalidade métricas de desempenho neutras. O ITF de cada fluxo expressa exatamente a diferenciação de tráfego que ocorre entre eles.

Enquanto o fluxo A possui ITF de 0,325 o fluxo B possui ITF de 0,025. Sendo assim, o fluxo A indica a existência de tráfego não-neutro.

Tabela 4 – Tabela de Análise Total dos Fluxos do Tráfego

Fluxo A											
	t 0	t 1	t 2	t 3	t 4	t 5	t 6	t 7	t 8	t 9	t 10
Latência	0	0	0	1	1	0	0	0	0	0	0
Jitter	0	0	1	1	1	0	0	1	0	0	0
Throughput	0	0	0	1	1	0	1	0	0	0	0
Perdas	0	0	0	1	1	0	1	1	0	0	0
IJT –>	0	0	0,25	1	1	0	0,5	0,5	0	0	0
ITF –> 0,325											

Fluxo B											
Latência	0	0	0	1	0	0	0	0	0	0	0
Jitter	0	0	0	0	0	0	0	0	0	0	0
Throughput	0	0	0	0	0	0	0	0	0	0	0
Perdas	0	0	0	0	0	0	0	0	0	0	0
IJT –>	0	0	0	0,25	0	0	0	0	0	0	0
ITF –> 0,025											

Fonte. Autor

Como exemplo, um tráfego de dados capturado contendo três fluxos diferentes FTP, HTTP/*facebook*, HTTP/*youtube*, sendo dois destes de uma mesmo protocolo, porém para serviços diferentes. Através do método é possível verificar a quebra da neutralidade por aplicação, comprovando a existência da política de diferenciação de tráfego. Assim, na hipótese de uma aplicação estar sofrendo a violação da neutralidade, tem-se a certeza de que a ocorrência estaria no fluxo da aplicação e não no protocolo em si.

Neste capítulo apresentou-se um novo método para diagnosticar a quebra da neutralidade através da relação entre as métricas de desempenho. Além disso, o método também demonstra ser agnóstico em relação aos fluxos de dados. No próximo capítulo é descrita toda a metodoliga utilizada para realização dos testes, após são apresentados os cenários no qual o método foi avaliado.

5 METODOLOGIA

Neste capítulo será apresentado o ambiente utilizado como cenário de testes nesta dissertação, além de todo o ferramental utilizados na criação e nos testes do ambiente.

5.1 AMBIENTE

O ambiente foi criado/montado em cima do *hypervisor* KVM (*Kernel Virtual Machine*), criando assim uma topologia utilizando máquinas virtuais. No KVM foram configurados quatro equipamentos que formam uma arquitetura simples de rede. A coleta do tráfego de dados ocorre através da medição passiva utilizando o *software* Wireshark. A proposta do trabalho simula 4 cenários diferentes para os testes, cada cenário possui uma configuração de roteador diferente. Em cada cenário é gerado tráfego de pacotes de aplicação FTP, HTTP e SMTP. Na aplicação HTTP, foram gerados dois fluxos, simulando assim uma mesma aplicação para serviços diferentes. A duração do tráfego gerado foi de 60 segundos. Tratando-se das amostras, foram realizados 30 (trinta) testes para cada tráfego gerado, e através das médias obteve-se o resultado final para cada cenário. Com as amostras de fluxos geradas obtiveram-se valores das métricas de desempenho de latência, de *jitter*, de *throughput* e da perda de pacotes.

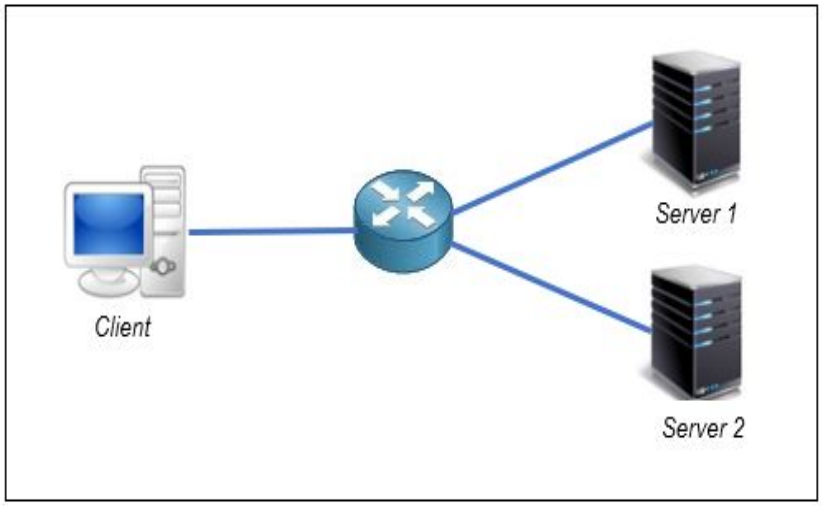
O ambiente proposto como cenário de testes para este trabalho foi desenvolvido utilizando a plataforma de virtualização KVM (*Kernel Virtual Machine*), sobre a qual foram configuradas máquinas e equipamentos virtuais que formam uma arquitetura de rede. As máquinas *Client* e *Servers* se conectam ao roteador através de *bridges*, simulando assim uma conexão de Internet (usuário – ISP – servidores), conforme Figura 7.

A arquitetura definida tem como base o modelo atual de conexão de Internet. Nesse modelo o usuário necessita estar conectado a um ISP, e a partir dessa conexão o usuário tem a acesso à Internet. Este modelo de conexão usuário-ISP-usuário é considerado como à borda da Internet, que representa os sistemas finais que executam as aplicações e onde se localiza o usuário.

5.2 DESCRIÇÃO DO AMBIENTE DE TESTES

O ambiente de testes é composto por quatro equipamentos virtualizados no *hypervisor* KVM: (i) um *host* que tem como característica de exercer a atividade de um usuário conectado

Figura 7 – Ambiente de testes



Fonte: acervo pessoal.

à Internet; (ii) dois *servers* que possuem como característica exercer o serviço de um servidor na Internet; (iii) e por último um roteador (*router*) que faz a interligação entre os equipamentos da rede e realiza a atividade de um ISP.

A configuração de cada equipamento:

a) *Client*: Sistema Operacional Ubuntu 16.04.1.LTS (*kernel* 4.4.0-3-*generic*), com 1GB de memória e 20GB de disco. Informações de rede com as seguintes configurações: IP – 192.168.121.10, Máscara – 255.255.255.0, *Gateway* – 192.168.121.1;

b) *Server1*: Sistema Operacional Ubuntu 16.04.1.LTS (*kernel* 4.4.0-3-*generic*), com 1GB de memória e 20GB de disco. Informações de rede com as seguintes configurações: IP – 192.168.122.20, Máscara – 255.255.255.0, *Gateway* – 192.168.122.1;

c) *Server2*: Sistema Operacional Ubuntu 16.04.1.LTS (*kernel* 4.4.0-3-*generic*), com 1GB de memória e 20GB de disco. Informações de rede com as seguintes configurações: IP – 192.168.122.22, Máscara – 255.255.255.0, *Gateway* – 192.168.122.1;

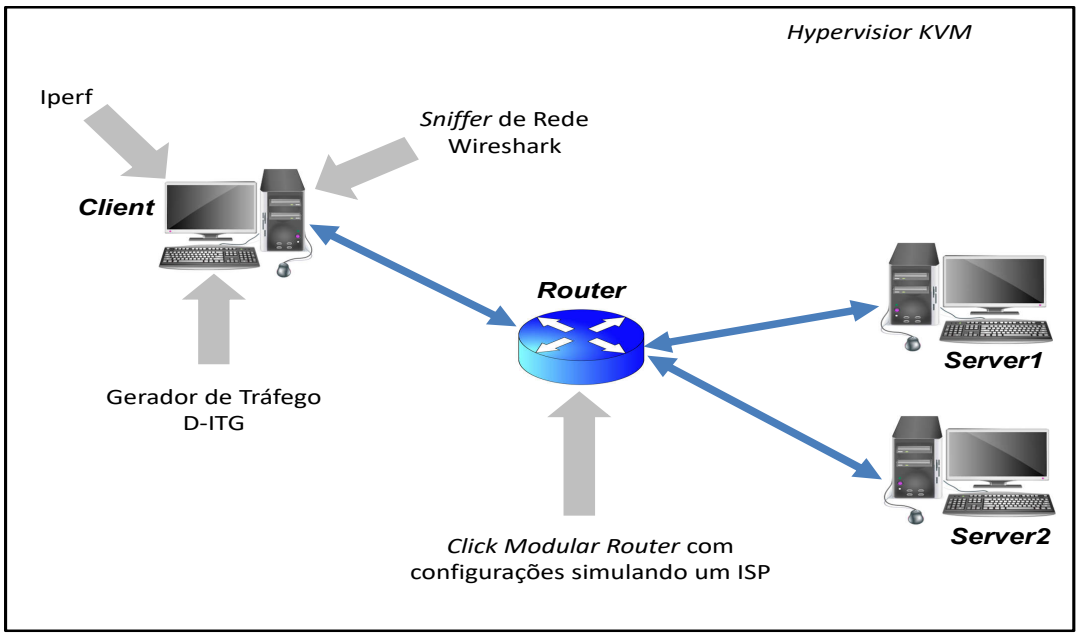
d) *Router*: Sistema Operacional Debian 7 (*kernel* 3.2.0-4-amd64), com 2GB de memória e 20GB de disco. O roteador por interligar duas redes distintas possui duas *bridges*;

e) *Bridge* 1: conecta o *host Client* ao *router*. IP – 192.168.121.1, Máscara – 255.255.255.0, *Gateway* – 192.168.121.1;

f) *Bridge* 2: conecta os *Servers* ao *router*. IP – 192.168.122.1, Máscara – 255.255.255.0, *Gateway* – 192.168.122.1.

Na configuração do ambiente, os fluxos são enviados da máquina *Client* para o *Server1* e *Server2*. O gerador de tráfego D-ITG está configurado no *Client* e nos *Servers* que simulam servidores de serviço. Entranto, o analisador de tráfego (Wireshark) é executado apenas na máquina *Client*. Todo o tráfego gerado tem como destino os *Servers*, passando pelo roteador que está configurado com uma das quatros configurações do *Click Modular Router*, definidas para este trabalho, como observa-se na Figura 8.

Figura 8 – Ambiente com as configurações



Fonte: acervo pessoal.

5.2.1 Configuração dos Roteadores

No ambiente proposto o roteador tem a função de simular um ISP, foram implementadas quatro configurações diferentes, sendo uma para cada cenário do ambiente. As configurações foram definidas conforme relatos na literatura sobre neutralidade de rede (DISCHINGER

et al., 2010), (TARIQ et al., 2009), (KANUPARTHY; DOVROLIS, 2010) e (ZHANG; MAO; ZHANG, 2009).

a) RtN (Roteador Neutro). Nesta configuração o roteador apenas encaminha os fluxos de dados sem fazer nenhum tratamento nos pacotes (*forward*).

b) RtNN1 (Roteador Não-Neutro 1). Configurado para descartar pacotes recebidos (*drop*). Usa função *RandomSample*, que descarta uma quantidade não determinística aproximada de 20% de pacotes conforme indicação na função. O descarte de pacotes acontece somente após o roteador receber mais de 24.000 pacotes dos fluxos HTTP1 e HTTP2. O roteador volta a ter configuração apenas de encaminhamento, após ultrapassar a taxa de 26.000 pacotes recebidos pelo roteador dos fluxos HTTP1 e HTTP2.

c) RtNN2 (Roteador Não-Neutro 2). Configurado para fazer uma redução no envio de pacotes conforme os fluxos recebidos. Usa função a *RatedSplitter*, que divide o fluxo de pacotes à taxa especificada de 400 pacotes por segundos. O controle da taxa de envio de pacotes acontece somente após o roteador receber mais de 26.000 pacotes dos fluxos HTTP1 e HTTP2. O roteador volta a ter configuração apenas de encaminhamento, após ultrapassar a taxa de 34.000 pacotes recebeidos pelo roteador dos fluxos HTTP1 e HTTP2.

d) RtNN3 (Roteador Não-Neutro 3). Configurado com a política de atraso de 0,5 segundos por pacote recebido, usa a função (*DelayUnqueue*). O atraso dos pacotes acontece somente após o roteador receber mais de 26.000 pacotes dos fluxos HTTP1 e HTTP2. O roteador volta a ter configuração apenas de encaminhamento, após ultrapassar a taxa de 34.000 pacotes recebeidos pelo roteador dos fluxos HTTP1 e HTTP2.

5.3 FERRAMENTAL

Nesta seção são apresentadas as ferramentas KVM, Click Modular Router, D-ITG, IPERF e Wireshark. Todas elas foram utilizadas na criação do ambiente e dos cenários, assim como para captura e análise dos fluxos de dados.

5.3.1 KVM

O KVM (*Kernel Virtual Machine*) é uma estrutura de virtualização residente no *kernel* para o *hardware* x86 e para sistemas Linux. Ele fornece suporte à virtualização completa e pode funcionar sem modificações no *kernel*, mas necessita que os processadores tenham as tecnologias de virtualização (Intel VT e AMD-V).

O virtualizador KVM tem um módulo principal e módulos específicos para cada tecnologia de virtualização de cada processador. O KVM não necessariamente emula o *hardware*, mas permite acesso do *qemu* a virtualização baseada em *hardware*. O KVM também possui suporte a uma série de ferramentas de gerenciamento, como por exemplo, o *Virsh*. Esta ferramenta é executada em linha de comando, permitindo fazer todo o gerenciamento das máquinas virtuais. Outra ferramenta é o *Virt Manager*, que oferece uma interface gráfica para o gerenciamento das máquinas virtuais. Ambas as ferramentas são fortemente dependentes da biblioteca *libvirt* (KIVITY et al., 2007).

Figura 9 – Ambiente do KVM simulando 4 máquinas virtuais

Fonte: Autor.

5.3.2 Click Modular Router

A tecnologia CLICK se baseia em uma arquitetura flexível e modular de *software* para criar e encaminhar pacotes. Com ela é possível criar roteadores, *switches*, *hubs*, entre outros elementos de uma rede IP utilizando como base um computador pessoal. Diversos aspectos da

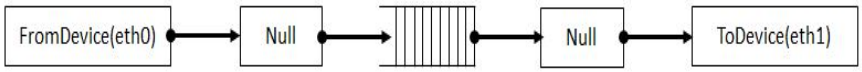
arquitetura CLICK são inspirados pelas propriedades de configurações dos roteadores (KOHLER et al., 2000).

A arquitetura CLICK é baseada em elemento, que são componentes de *softwares* que representam uma unidade de processamento. Elementos realizam operações conceitualmente simples, em vez de grandes cálculos complexos, como o próprio roteamento.

Os elementos CLICK suportam dois tipos de conexões, chamados de *push* e *pull*. Em uma conexão do tipo *push* o elemento origem inicia a transferência passando os pacotes a um elemento de destino. Isto corresponde à maneira com que os pacotes se movem na maioria dos roteadores por software. Já na conexão do tipo *pull*, o elemento de destino inicia a transferência pedindo por um pacote (KOHLER et al., 2000).

Na Figura 10 o elemento *FromDevice*, que captura pacotes que chega a interface *eth0*, possui um porta se saída do tipo *push*, portanto tal elemento só poderá se ligar à outro elemento que possua porta de entrada do mesmo tipo *push*, no caso o elemento *Null*.

Figura 10 – Conexão entre elementos no CLICK



Fonte: (KOHLER et al., 2000).

Os arquivos de configuração do CLICK são executados sobre um *driver* que implementa várias facilidades para todos os elementos. Existem dois *driver*: o *user-level*, que executa como uma aplicação a nível de usuário, e o Linux *kernel*, o *linux-module*, que executa como um módulo do *kernel* Linux. O *driver* a nível de *kernel*, *linuxmodule*, substitui completamente a pilha de rede do sistema operacional, transformando um computador pessoal convencional num roteador, alcançando alto desempenho (KOHLER et al., 2000).

Além disso, pode-se perceber que o ambiente CLICK, devido a sua modularidade, oferece acesso a pesquisa e desenvolvimento das configurações dos roteadores, que nos equipamentos comercias é praticamente impossível devido às licenças de uso do *hardware* e *softwares*, e ter como outro agravante o seu custo elevado.

5.3.3 D-ITG

A ferramenta utilizada para gerar o tráfego no cenário criado neste trabalho foi o *Distributed Internet Traffic Generator* (D-ITG) (AVALLONE et al., 2004). O D-ITG permite emular o tráfego com várias caracteristicas diferentes (*e.g.*, número da porta de envio, protocolo a ser enviado, velocidade de pacotes por segundo, duração do tráfego entre outra informações) (BOTTA et al., 2013).

O D-ITG consegue gerar vários tipos de tráfegos, suporta protocolos de transportes e replica ICMP. Além disso, realiza medições de variação do atraso (*jitter*), atraso (*delay*) onde é calculado o atraso mínimo, médio e máximo, também mede perda de pacotes e número de pacotes transferidos. Essas medições não foram utilizadas neste trabalho, utilizou-se apenas a geração de tráfego específico e com caracteristicas determinadas.

A ferramenta D-ITG é composta de cinco módulos básicos: ITGSend, ITGRecv, ITGLog, ITGDec e ITGManager. Entretanto, utilizou-se apenas a função de envio de tráfego (ITGSend) e a função para receber o tráfego (ITGRecv). Pois, o D-ITG segue o modelo cliente-servidor, onde o ITGSend é o emissor (cliente) que pode gerar um único fluxo de tráfego ou múltiplos fluxos, e O ITGRecv atua como servidor para receber os dados.

Os testes que foram realizados com quatro fluxos simultanêos de mesmo protocolo, onde cada fluxo possui uma característica específica e diferente dos demais fluxos. Para isso utilizou-se o *script* conforme demonstrado na Tabela 5.

Tabela 5 – Script para geração de tráfego de dados

-a 192.168.122.20 -rp 21 -T TCP -C 500 -t 6000 -c 512	(a)
-a 192.168.122.20 -rp 80 -T TCP -C 500 -t 6000 -c 512	(b)
-a 192.168.122.22 -rp 80 -T TCP -C 500 -t 6000 -c 512	(c)
-a 192.168.122.20 -rp 25 -T TCP -C 500 -t 6000 -c 512	(d)

Fonte. Autor

A funcionalidade do *script* basea-se num tráfego de dados compostos de 4 fluxos que são enviados ao mesmo tempo para dois destinos diferentes. Os fluxos (a), (b) e (d) possuem como destino o IP 192.168.122.20, já o fluxo (c) possue como destino o IP 192.168.122.22. Entretanto, para os fluxos chegarem ao destino, eles são processados, analisados e redirecionados pelo roteador.

5.3.4 Iperf

O Iperf (IPERF, 2017) é uma ferramenta *opensource*, desenvolvida pela *Distributed Applications Support Team* (DAST) no laboratório nacional de investigação de rede aplicada (NLANR) da universidade de Illinois e está disponível para os sistemas operacionais Windows (Gates Allen, 1985), Linux (Torvalds, 1991), Mac OSX (Apple, 2001), FreeBSD (FreeBSD-Project, 1993), Android (Google, 2008) e iOS (Apple, 2007).

A ferramenta Iperf tem como funcionalidade principal o estresse da rede, realizando testes de desempenho entre hosts. O cliente gera um determinado fluxo de dados para o servidor Iperf em escuta, então é enviado uma certa quantidade de pacotes por segundo, assim, a ferramenta testa a capacidade máxima do meio de transmissão (IPERF, 2017).

A ferramenta Iperf foi compilada tanto no *Client* quanto no *Server2*. O servidor fica em modo escuta, e o cliente gera tráfego no sentido ao servidor. Os testes foram realizados nos quatro cenários. No tráfego de rede gerado pelo cliente, utilizou-se o tamanho fixo do fluxo de dados, endereço IP do servidor e o protocolo da camada de transporte. Nos testes foram efetuados com o protocolo TCP, e também foi necessário configurar a largura de banda utilizada no cenário para velocidade de 200 Mbps, conforme Tabela 6.

Tabela 6 – Script do Iperf para geração de tráfego

iperf3 -c 192.168.122.22 -b 200.00M -t 65

Fonte. Autor

5.3.5 Wireshark

A ferramenta Wireshark (WIRESHARK, 2017) permite capturar e analisar as mais complexas e variadas formas que os pacotes passam por uma determinada interface. Esta ferramenta consegue analisar detalhes dos pacotes, como endereçamento (IPs de origem e destino), tamanho, tempo de transmissão, reenvio, perda de pacotes, porta de origem e destino entre outros detalhes (OREBAUGH; RAMIREZ; BEALE, 2006).

O Wireshark é utilizado para capturar o tráfego gerado no ambiente, assim como na

filtragem dos fluxos capturados. Através de filtros definidos foram selecionados os diferentes tipos de fluxos para posterior analise.

Foram utilizados alguns filtros para selecionar os fluxos capturados para este trabalho:

- **ip.dst:** mostra somente ocorrência para o IP de destino estabelecido;

- **ip.scr:** mostra somente ocorrência para o IP de origem estabelecido;

- **tcp.port:** mostra somente ocorrência para a porta estabelecida;

- **tcp.analysis.ack_rtt:** mostra o tempo de ida e volta (*Round Trip Time*-RTT) dos segmentos TCP;

- **tcp.analysis.retransmission:** mostra as retransmissões de segmentos TCP;

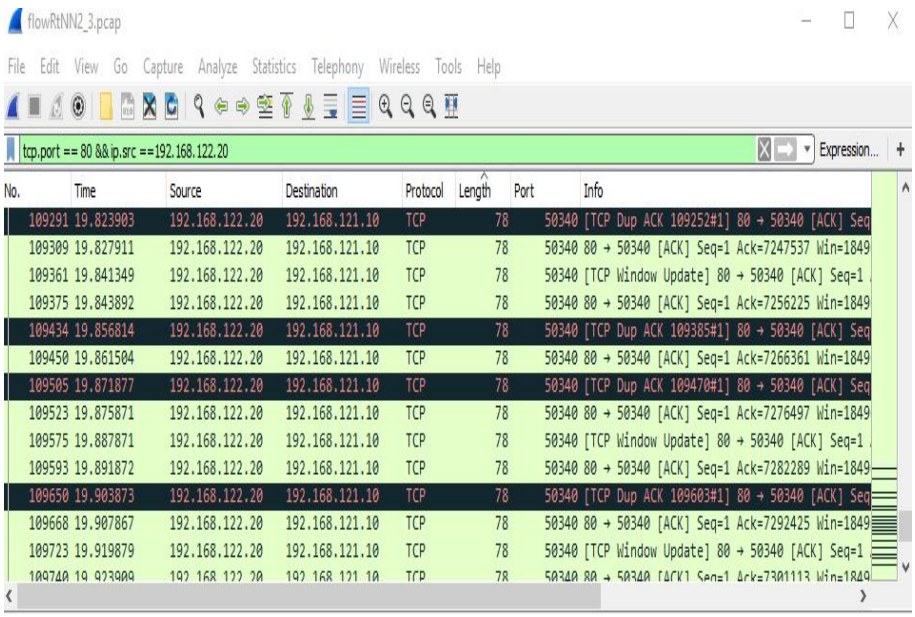
- **tcp.analysis.fast_retransmission:** mostra as retransmissões rápidas dos segmentos TCP;

- **tcp.analysis.out_of_order:** mostra a ocorrência de segmentos fora de ordem;

- **tcp.analysis.duplicate_ack:** mostra a ocorrência para ACKs duplicados.

Com o Wireshark, é possível classificar e separar os fluxos, identificando inclusive as perdas de pacotes. A Figura 11, mostra pacotes capturados com seus endereços e origem, destino, protocolos, tamanhos, portas e outras informações. As linhas pretas representam pacotes duplicados devido a perda de pacotes.

Figura 11 – Janela do Wireshark com filtro por porta TCP e IP de origem



Fonte: acervo pessoal.

6 AVALIAÇÃO

Este capítulo apresenta as avaliações dos resultados obtidos com a utilização do método proposto nesta dissertação. Na Seção 6.1 descreve-se a avaliação dos resultados na qual a configuração do roteador é de encaminhamento de pacotes (*forward*). Na Seção 6.2 é descrito os resultados das avaliações com o roteador configurado para realizar descarte de pacotes de alguns fluxos (*drop*). Assim, na Seção 6.3 avaliou-se os resultados aplicando configuração na qual o roteador realiza redução na velocidade dos pacotes (*traffic shaping*) para fluxos específicos. Na última avaliação, Seção 6.4, a configuração do roteador é para realizar atraso (*delay*) em determinados fluxos.

Os cenários foram configurados para simular políticas de um ISP. Nos testes gerou-se quatro fluxos do mesmo protocolo (TCP), porém de três aplicações diferentes (FTP, HTTP e SMTP). Os fluxos iniciam-se a trafegar pelo cenário e após uma certa quantidade de pacotes enviados o roteador inicia o processo de diferenciação de tráfego em dois fluxos

Os fluxos de dados são gerados pela máquina *client* e enviados para os *Servers*, durante 60 segundos. Esta operação realizou-se 30 vezes, nas quais calculou-se as médias no intervalo de tempo de 1 segundo, para cada métrica de desempenho. Assim, obteve-se um arquivo contendo valores de cada fluxo com as respectivas métricas.

Para avaliar o método agnóstico de detecção da quebra da neutralidade, é comparado o resultado final do algoritmo desenvolvido neste trabalho, com gráficos gerados em um *software* de planilhas eletrônicas.

6.1 ROTEADOR NEUTRO (RTN)

Neste cenário o roteador é configurado para execer a função de encaminhamento direto de todos os pacotes que trafegam por ele, ou seja, encaminha os fluxos sem fazer nenhum tratamento nos pacotes. A configuração do roteador ocorre através do CLICK MODULAR ROUTER, conforme Apêndice A.

Após testes com os fluxos de dados já mencionados, obteve-se através do método proposto nesta dissertação o resultado conforme Figuras 12 e 13. Nota-se que o resultado demonstra que o método identificou corretamente 99,69% dos resultados, consequentemente o método indicou 0,31% de Falsos-Negativos. Os valores Falsos-Negativos aconteceram no Fluxo FTP

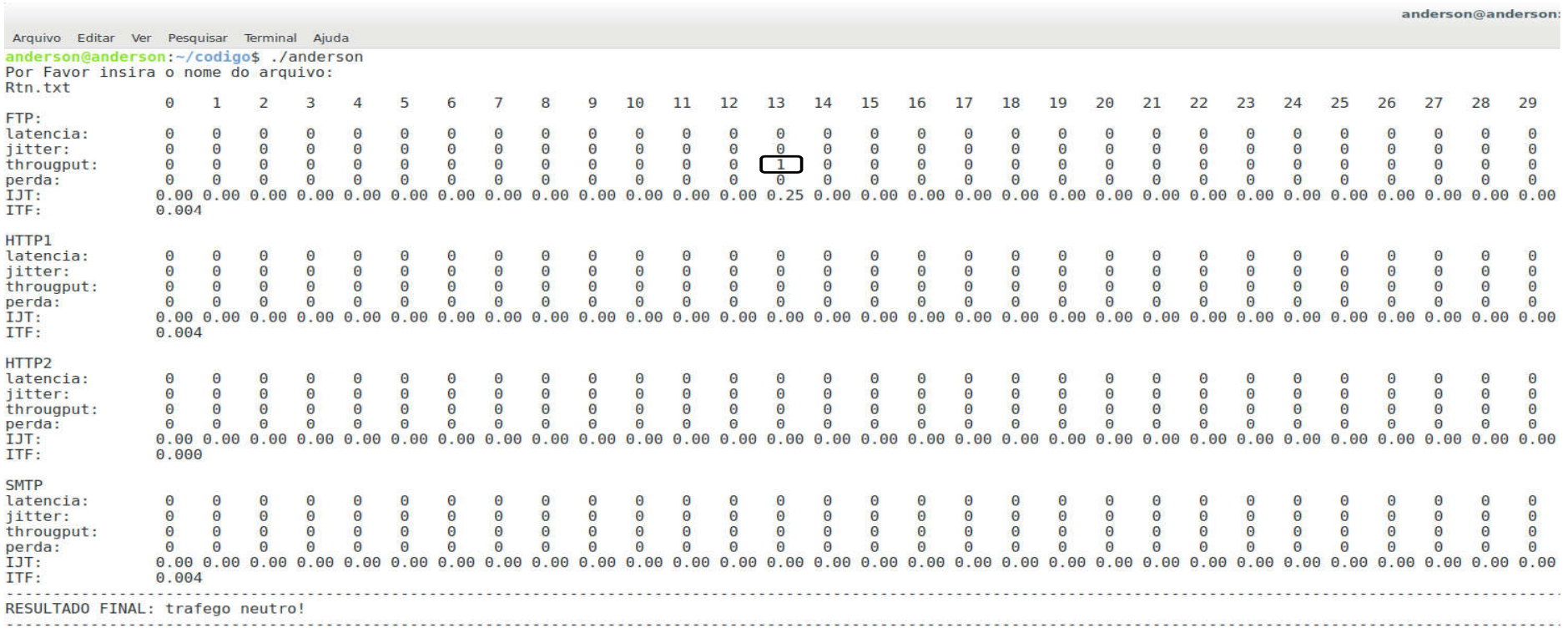
na métrica de *throughput* janela de tempo de 13 segundos; outro ocorreu no Fluxo HTTP1 na métrica de *jitter* no tempo de 33 segundos; e por último no Fluxo SMTP na métrica de *jitter* no tempo de 50 segundos. Entretanto, no fluxo HTTP2 só ocorreram Verdadeiros-Negativos.

Visualizando-se os gráficos gerados neste cenário proposto e comparando-os com o resultado do método, nota-se que métrica de latência dos fluxos se comportaram de maneira estável e estão entre as medidas de 0,300ms até 0,400 ms, conforme Figura 14.

A métrica de *jitter* dos fluxos apresentou-se de maneira estável, porém com pequena oscilação no fluxo HTTP1 no tempo de 33 segundos, assim como o fluxo SMTP no tempo de 50 segundos, conforme Figura 15. Esses pontos podem ser considerados como um simples congestionamento nesse instante de tempo.

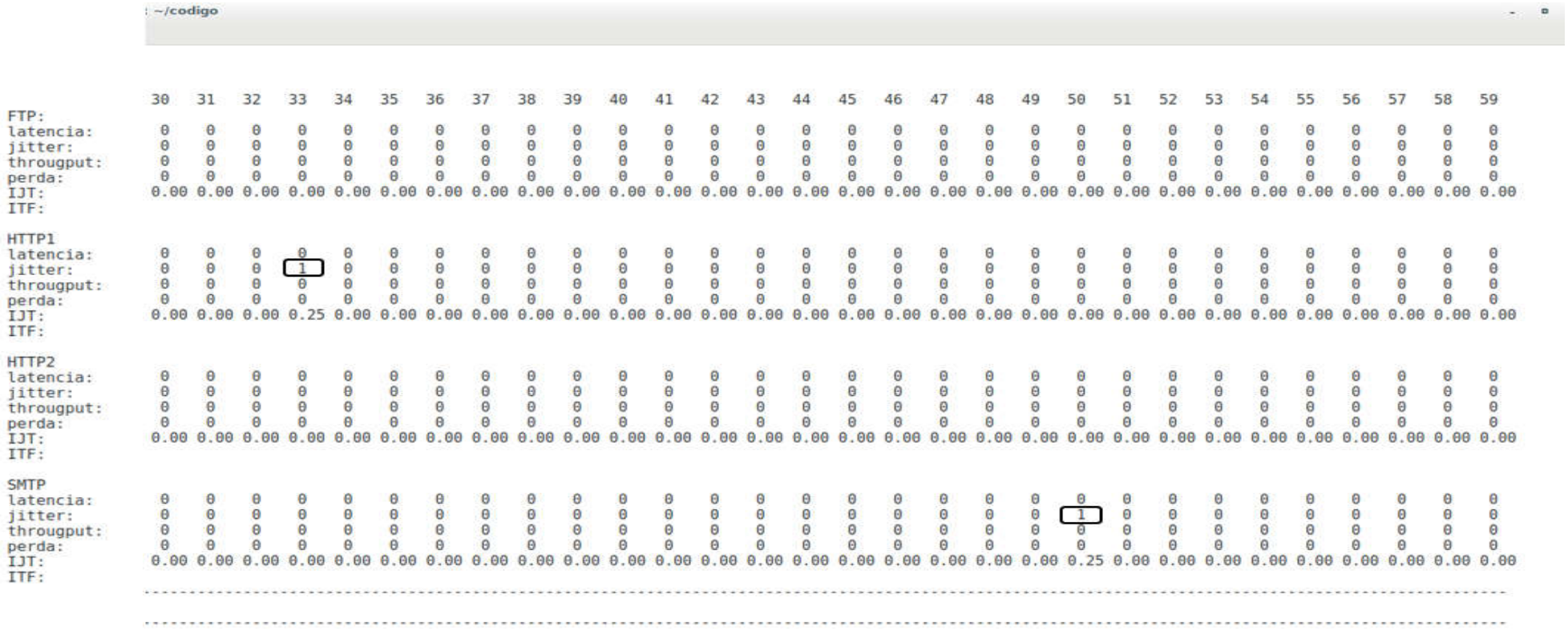
Outro falso-negativo ocorreu no fluxo FTP na métrica de *throughput* na janela de tempo de 13 segundos, no restante o fluxo comportou-se de maneira estável, conforme Figura 16. Por último, a métrica de perda de pacotes dos fluxos não apresenta nenhum valor diferente de zero, consequentemente as informações são de fluxos neutros.

Figura 12 – Tela de visualização do método agnóstico para a configuração RtN - Primeiros 30 segundos



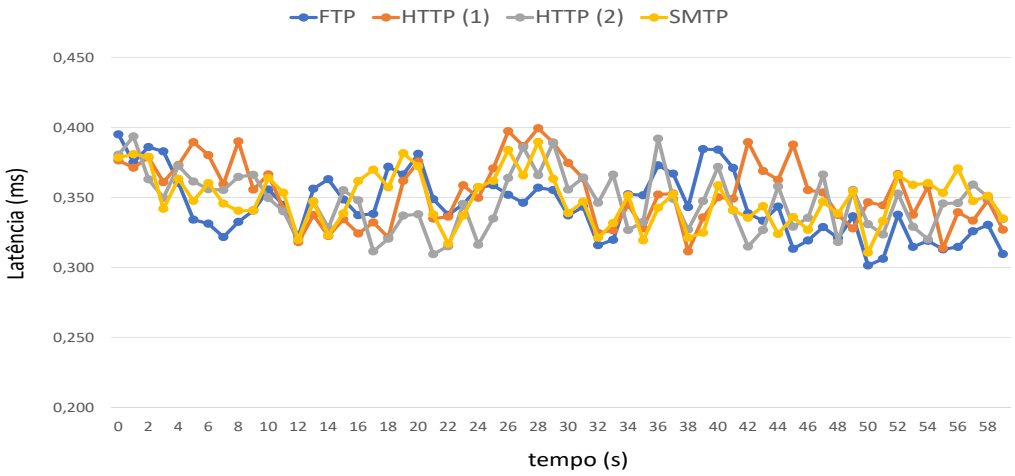
Fonte: acervo pessoal.

Figura 13 – Tela de visualização do método agnóstico para a configuração RtN - Últimos 30 segundos



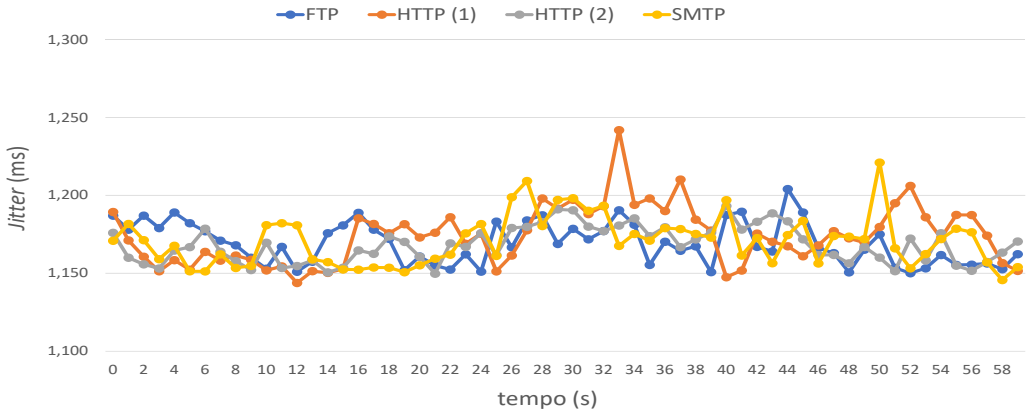
Fonte: acervo pessoal.

Figura 14 – Latência do tráfego no roteador neutro



Fonte: acervo pessoal.

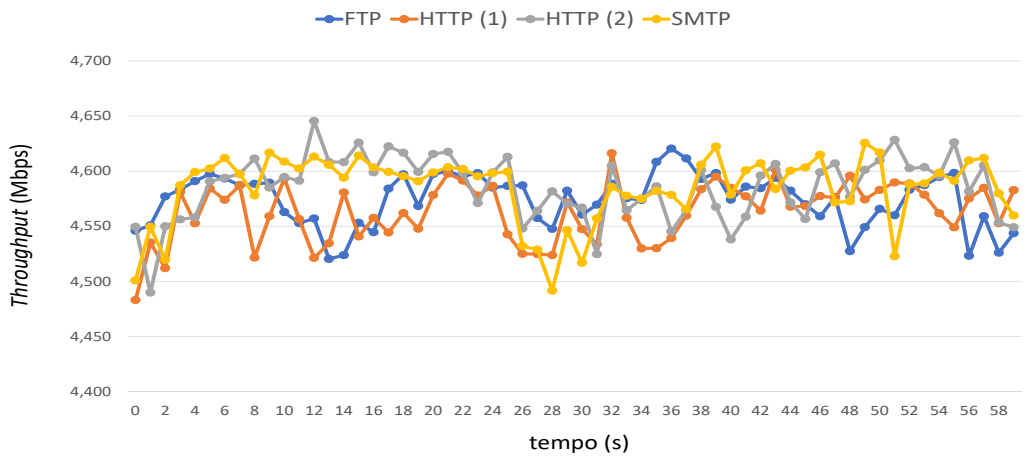
Figura 15 – *Jitter* do tráfego no roteador neutro



Fonte: acervo pessoal.

O ITF de cada fluxo (FTP = 0,004 - HTTP1 = 0,004 - HTTP2 = 0 - SMTP = 0,004) demonstra que o tráfego é considerado neutro, apesar de uma pequena variação nos resultados do ITF dos fluxos FTP, HTTP1 e SMTP. Esta variação é identificada através do método,

Figura 16 – *Throughput* do tráfego no roteador neutro



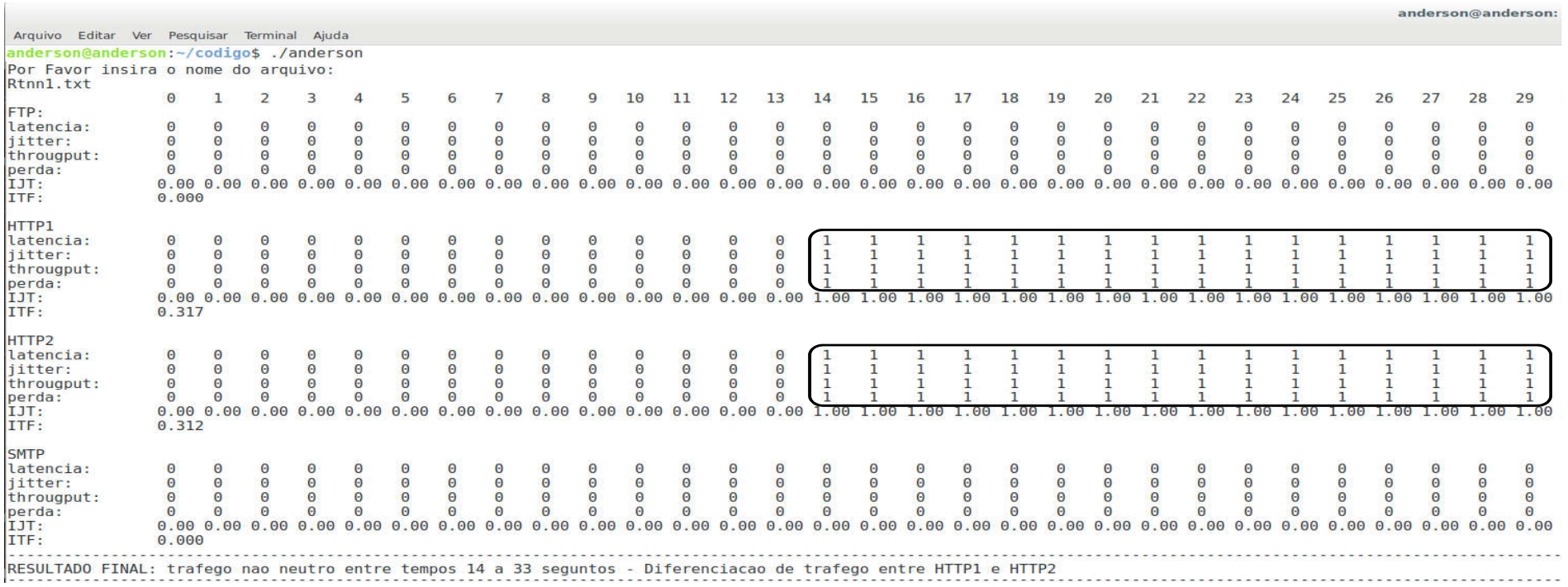
Fonte: acervo pessoal.

onde diagnosticou-se 3 falsos-negativos (0,31%) e 957 verdadeiros-negativos (99,69%). Representando assim um índice de confiabilidade muito alta. Pode-se afirmar que a avaliação para o tráfego capturado no cénario de configuração neutra foi eficaz, pois o resultado do método condiz com as informações resultantes dos gráficos.

6.2 ROTEADOR NÃO-NEUTRO 1 (RTNN1)

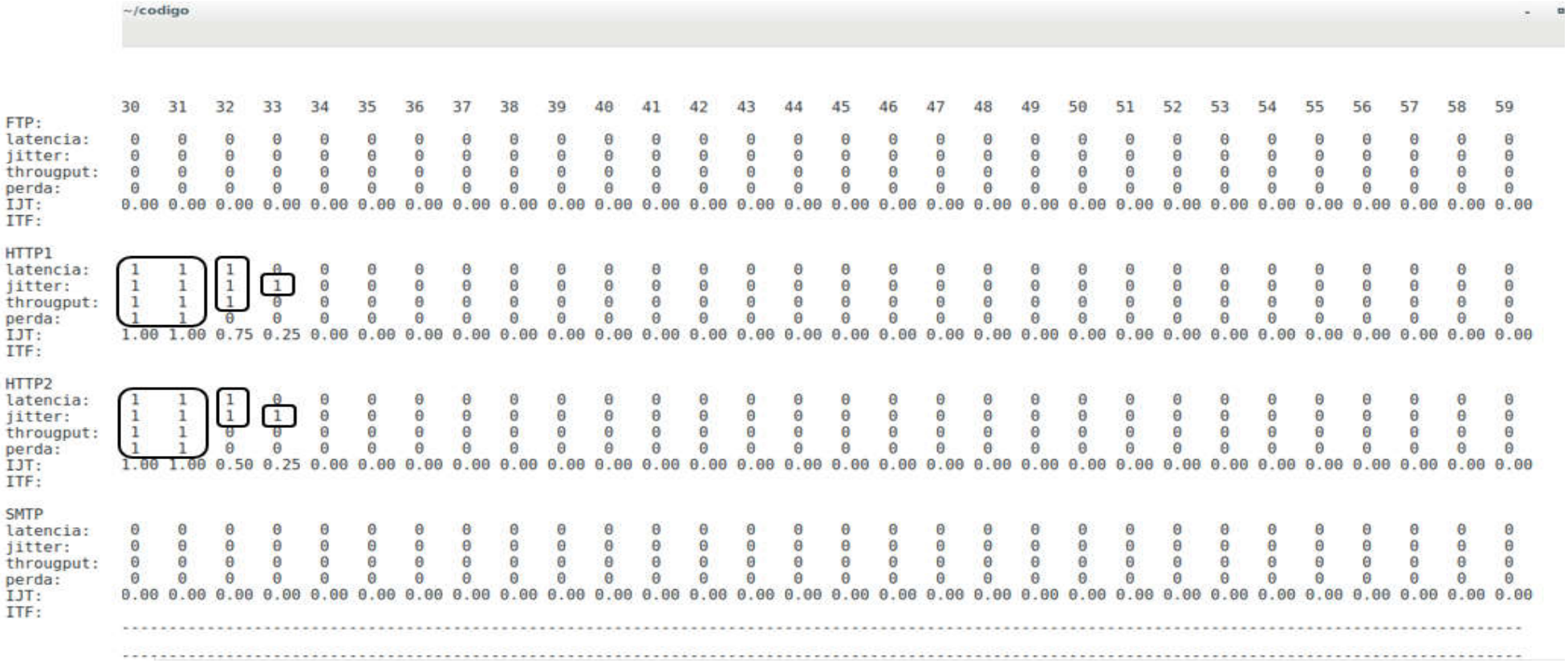
Neste teste de avaliação o roteador está configurado para descartar uma quantidade não determinística, porém esta quantidade tende a ser próxima de 20%. O descarte de pacotes acontece somente após o roteador receber mais de 24.000 pacotes dos fluxos HTTP1 e HTTP2, a partir desse momento os fluxos sofrem uma discrimição em seu tráfego. O roteador volta a ter configuração apenas de encaminhamento, após ultrapassar a taxa de 26.000 pacotes recebidos pelo roteador dos fluxos HTTP1 e HTTP2. Os demais fluxos do tráfego não são afetados pela configuração. O resultado do método é demonstrado nas Figuras 17 e 18. A configuração do roteador ocorre através do CLICK MODULAR ROUTER, conforme Apêndice B.

Figura 17 – Tela de visualização do método agnóstico para a configuração RtNN1 - Primeiros 30 segundos



Fonte: acervo pessoal.

Figura 18 – Tela de visualização do método agnóstico para a configuração RtNN1 - Últimos 30 segundos

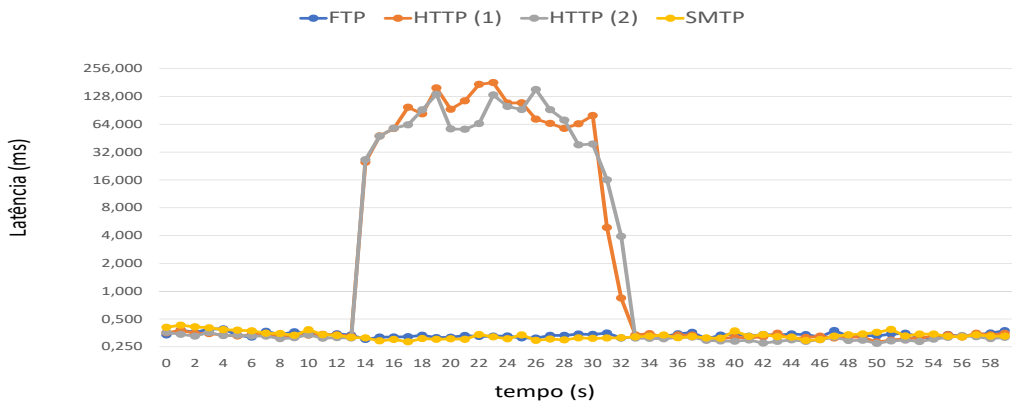


Fonte: acervo pessoal.

Percebe-se valores iguais a 1 em todas as métricas de desempenho nos fluxos HTTP1 e HTTP2 entre as janelas de tempo de 14 à 31 segundos. Consequemente, representa que fluxos são não-neutros entre esses tempos, e nota-se que a diferenciação só acontece nas aplicações HTTP, sendo que as demais permanecem neutras. Nesse sentido, o método consegue identificar corretamente o momento exato que tem início a diferenciação de fluxos e também por quanto tempo acontece.

Em comparação aos gráficos gerados, a métrica de latência dos fluxos comportou-se de maneira estável até o ponto 13. A partir do próximo ponto os fluxos HTTP1 e HTTP2 sofrem uma alteração brusca em seus valores de latência. Esta diferença entre os fluxos ocorre até a janela de tempo de 31 segundos, depois os valores começam a diminuir e estabilizam-se na janela de 32 segundos, após os quatros fluxos pernamencem iguais, conforme Figura 19.

Figura 19 – Latência do tráfego no roteador que realiza descarte de pacotes

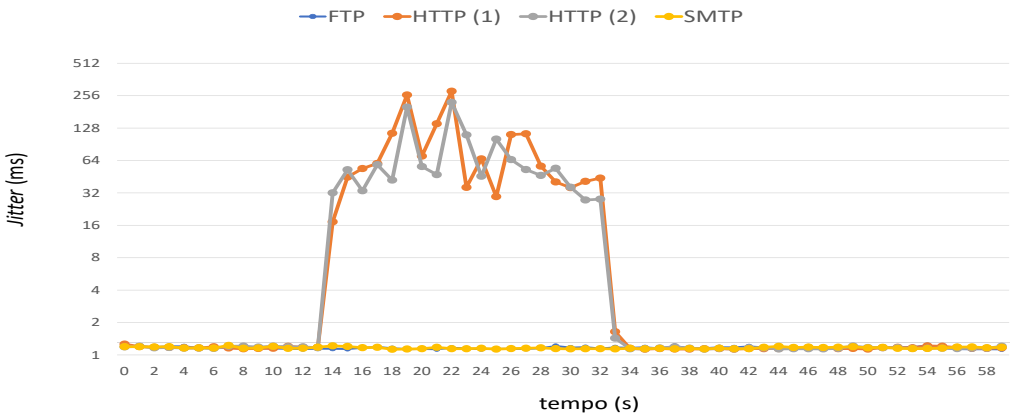


Fonte: acervo pessoal.

A métrica de *jitter* dos fluxos HTTP1 e HTTP2 também manteve-se constante no inicio, conforme Figura 20. Uma alteração brusca nos valores é notada a partir da janela de 14 segundos, porém ela se mantem até o tempo de 33 segundos, diferentemente das outras métricas. Esse fator deve-se a caracterítica do *jitter* que está diretamente relacionada a latência. Os valores tornam a ficar neutros a partir da janela de 34 segundos dos dois fluxos que quase voltam a normalidade.

A métrica de *throughput* dos fluxos também sofre diferenciação na janela de tempo de 14 segundos, conforme Figura 21. Entretanto, o fluxo HTTP1 volta a normalidade 1 segundo

Figura 20 – *Jitter* do tráfego no roteador que realiza descarte de pacotes

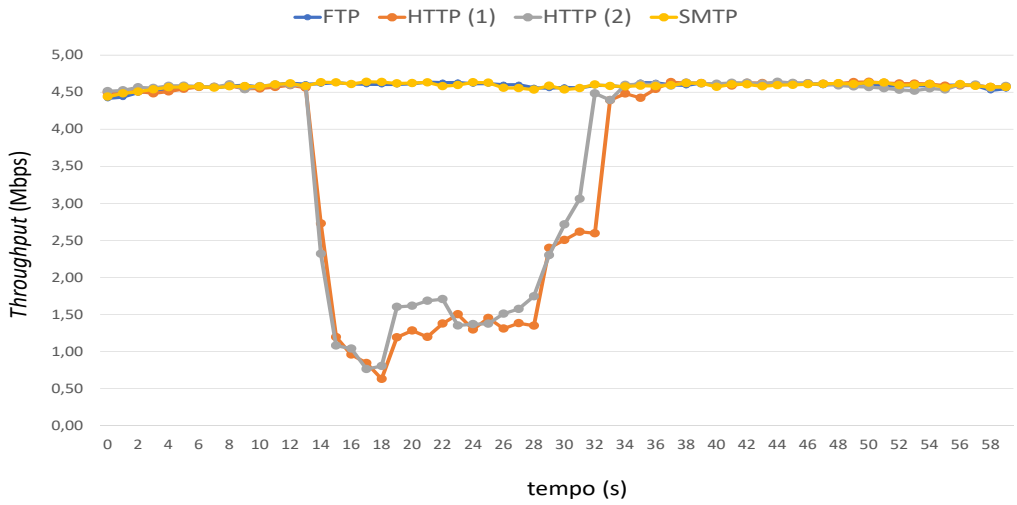


Fonte: acervo pessoal.

após o fluxo HTTP2.

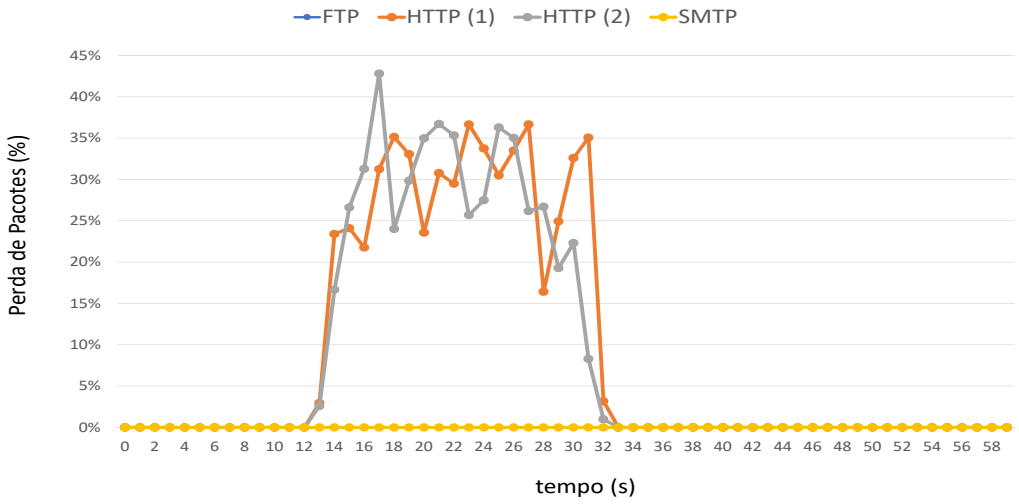
Por último, a métrica de perda de pacotes dos fluxos que apresenta degradação tem suas alterações perceptíveis entre a janela de 14 segundos e a de 32 segundos, conforme Figura 22.

Figura 21 – *Throughput* do tráfego no roteador que realiza descarte de pacotes



Fonte: acervo pessoal.

Figura 22 – Perda de pacotes do tráfego no roteador que realiza descarte de pacotes



Fonte: acervo pessoal.

Com a análise dos resultados através do ITF e da relação dos IJTs de cada fluxo, nota-se a existência de diferenciação de tráfego entre HTTP1 e HTTP2. O ITF de valor 0 (zero) dos fluxos FTP e SMTP representam que ambos são neutros.

As métricas de desempenho (latência, *jitter*, *throughput* e perda de pacotes) apresentam semelhanças nas janelas de tempo quando ocorre a degração do tráfego. O comportamento dos valores das métricas de latência, *jitter* e perda de pacotes são semelhantes na variação entre as janelas. O comportamento da métrica de *throughput* é inversa aos demais, apresentando um diminuição do volume de dados nos fluxos. Outra característica da métrica de *throughput* é que com o avanço na janela de tempo o fluxo de dados aumenta gradativamente, até voltar a normalidade.

A avaliação para o tráfego capturado no cénario de configuração de descarte de pacotes foi considerada positiva, pois, o resultado do método é confirmado pelos gráficos gerados. Através do método não foi observado nenhum falso-negativo, consequentemente tem-se 100% de verdadeiros-negativos. Representando assim um método eficaz.

6.3 ROTEADOR NÃO-NEUTOR 2 (RTNN2)

Nesta avaliação o roteador é configurado para realizar *traffic shaping* em dois fluxos determinados. A redução da taxa de envio de pacotes acontece somente após o roteador receber mais de 26.000 pacotes dos fluxos HTTP1 e HTTP2, a partir desse momento os fluxos sofrem uma discrimição em seu tráfego. O roteador volta a ter configuração apenas de encaminhamento, após ultrapassar a taxa de 34.000 pacotes recebidos pelo roteador dos fluxos HTTP1 e HTTP2. Os demais fluxos do tráfego não são afetados pela configuração. O resultado do método é demonstrado nas Figuras 23 e 24. A configuração do roteador ocorre através do CLICK MODULAR ROUTER, conforme Apêndice C.

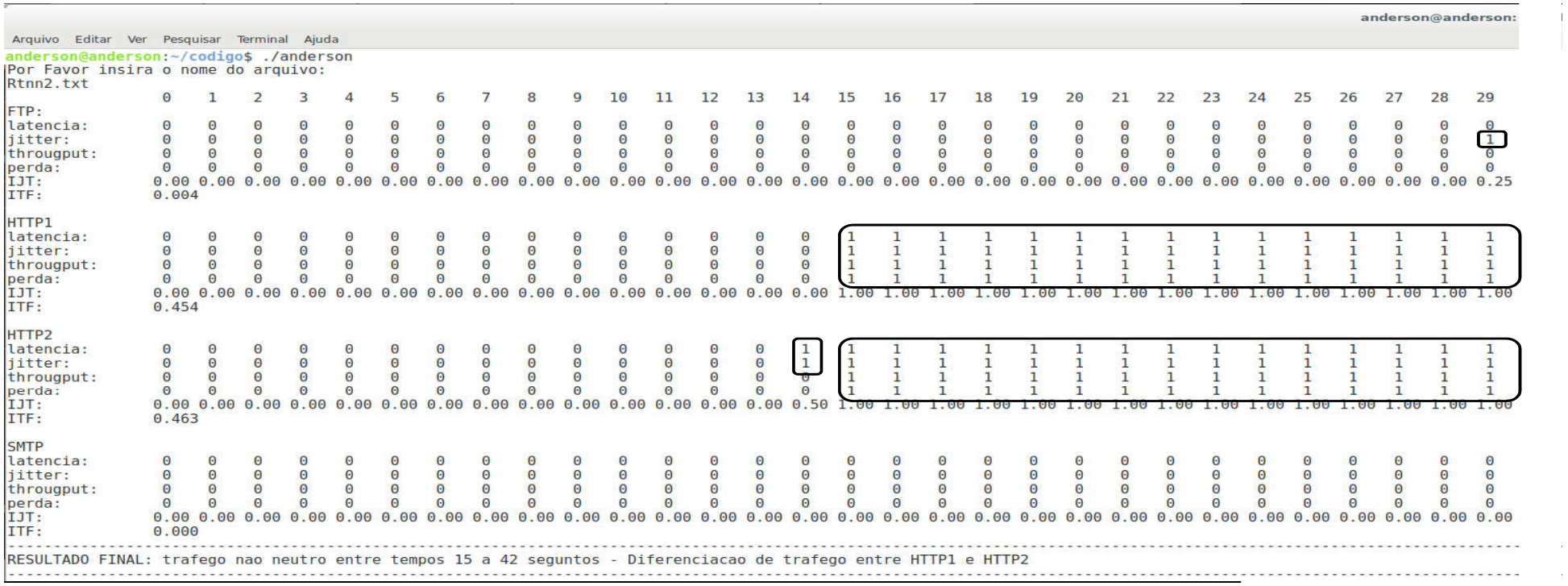
Constata-se que ocorreu apenas um Falso-Negativo no fluxo FTP na métrica de *jitter* na janela de tempo 29 segundos. O método identificou corretamente o momento exato que se inicia a degração de fluxos e também por quanto tempo a degradação ocorre.

Percebe-se valores iguais a 1 em todas as métricas de desempenho nos fluxos HTTP1 e HTTP2 entre as janelas de tempo de 14 à 39 segundos. Consequemente, representa que fluxos são não-neutros entre esses tempos, e nota-se que a diferenciação só acontece nas aplicações HTTP, sendo que as demais permanecem neutras. Nesse sentido, o método novamente consegue identificar a janela de tempo de início e término de cada ocorrência não-neutra. Entretanto, ressalta-se que a janela de tempo onde encerra-se as anomalias de fluxos são diferentes.

Em comparação aos gráficos gerados, a métrica de latência dos fluxos comportou-se de maneira estável até o ponto 13 no fluxo HTTP2, pois no fluxo HTTP2 a métrica manteve-se estável até 14 segundos, a partir desses pontos os fluxos HTTP1 e HTTP2 sofrem uma alteração brusca em seus valores de latência. Esta diferença entre os fluxos ocorre até a janela de tempo de 43 segundos para HTTP1 e 41 segundos para HTTP2, após os valores estabilizam-se permanecendo próximos aos valores dos outros fluxos, conforme Figura **??**. Entretando, nota-se a existência de 1 Falso-Negativo que acontece no Fluxo FTP na janela de tempo de 29 segundos.

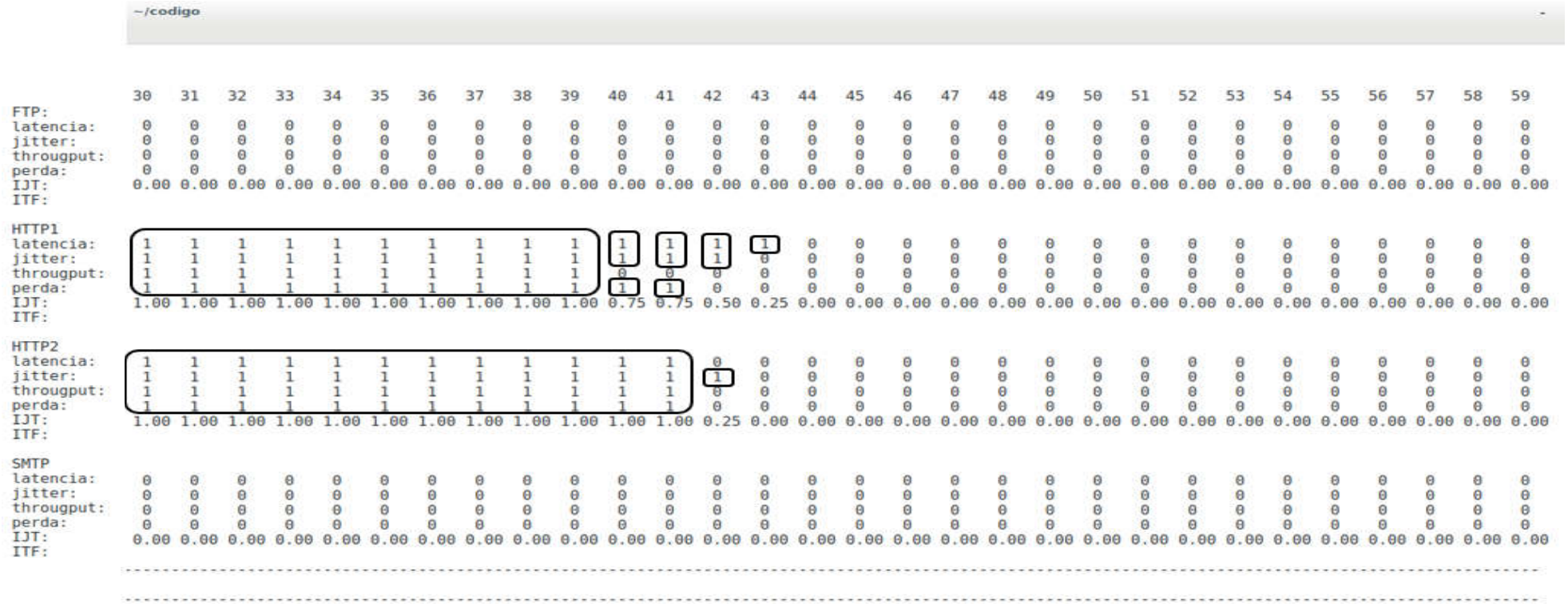
A métrica de *jitter* dos fluxos HTTP1 e HTTP2 também manteve-se constante no início. Uma alteração brusca nos valores é notada a partir da janela de 15 segundos no fluxo HTTP1 e de 14 segundos no fluxo de HTTP2, e mantiveram-se assim até o tempo de 41 segundos, diferentemente das outras métricas. Após os dois fluxos voltam a normalidade, conforme 26

Figura 23 – Tela de visualização do método agnóstico para a configuração RtNN2 - Primeiros 30 segundos



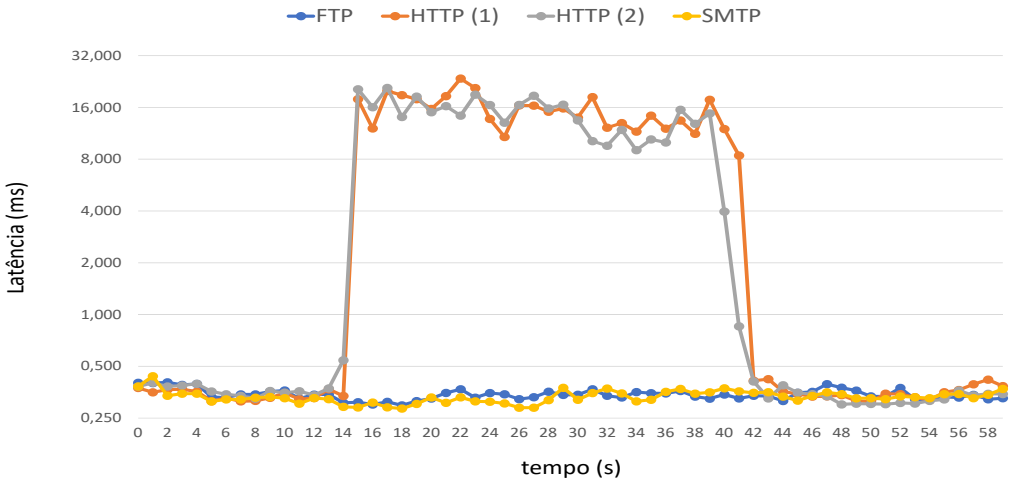
Fonte: acervo pessoal.

Figura 24 – Tela de visualização do método agnóstico para a configuração RtNN2 - Últimos 30 segundos



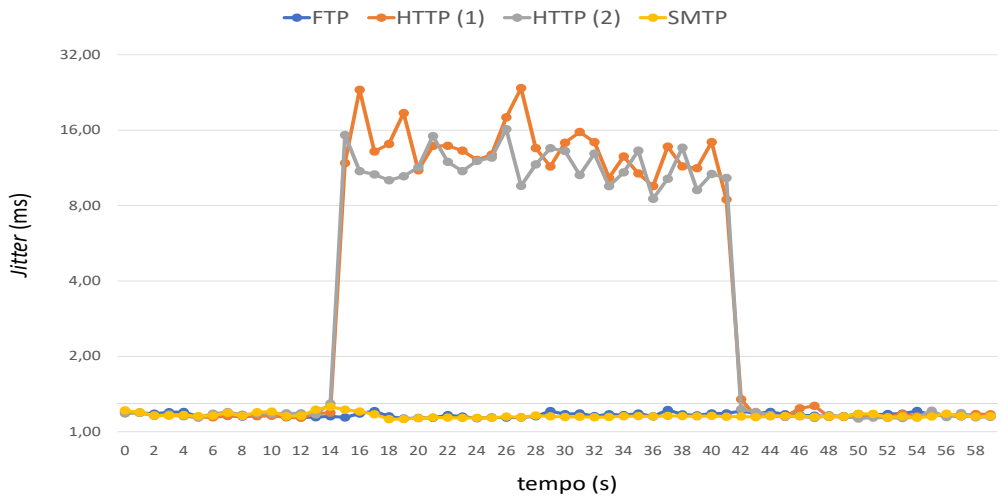
Fonte: acervo pessoal.

Figura 25 – Latência do tráfego no roteador que realiza *traffic shaping*



Fonte: acervo pessoal.

Figura 26 – *Jitter* do tráfego no roteador que realiza *traffic shaping*



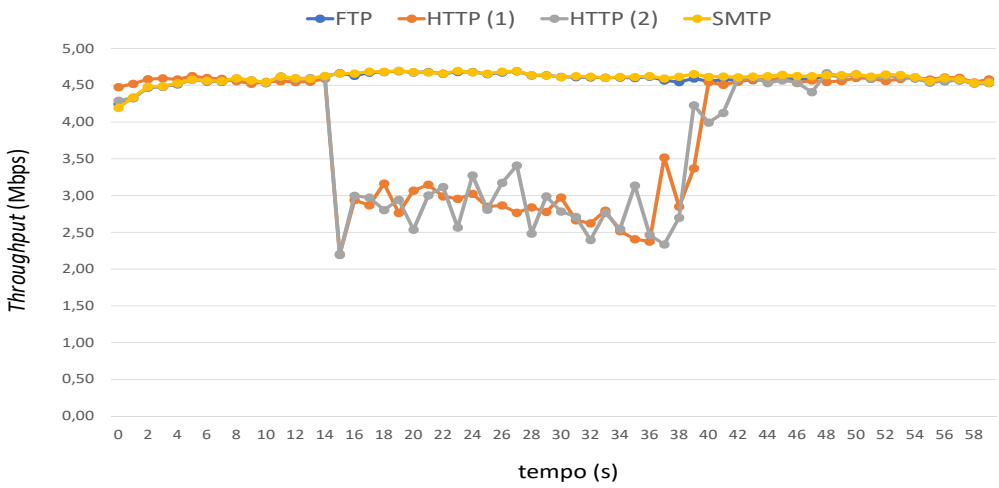
Fonte: acervo pessoal.

A métrica de *throughput* dos fluxos também sofre diferenciação na janela de tempo de 15 segundos, conforme Figura 21. Entretanto, o fluxo HTTP2 volta a normalidade 2 segundo

após o fluxo HTTP1.

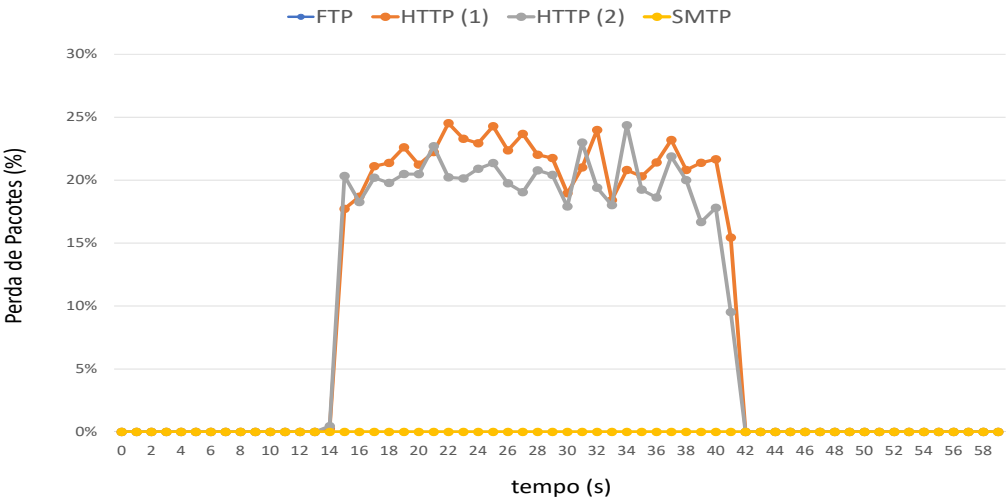
Por último, a métrica de perda de pacotes dos fluxos que apresenta degradação tem suas alterações perceptíveis entre a janela de 15 segundos e a de 41 segundos, conforme Figura 28.

Figura 27 – *Througput* do tráfego no roteador que realiza *traffic shaping*



Fonte: acervo pessoal.

Figura 28 – Perda de pacotes do tráfego no roteador que realiza *traffic shaping*



Fonte: acervo pessoal.

Com a análise dos resultados, nota-se a existência de diferenciação de tráfego entre os fluxos de mesmo protocolo HTTP através do ITF e da relação entre os IJTs. Durante a janela de tempo de 15 segundos à 39 segundos todas as métricas de desempenho (latência, *jitter*, *throughput* e perda de pacotes) apresentam tráfego não-neutro. Nesse mesmo período o fluxo FTP apresenta um anomalia apenas no tempo 29 segundos na métrica *jitter*. Enquanto o fluxo SMTP apresenta fluxo neutro durante toda sua janela de tempo.

O comportamento dos valores das métricas de latência, *jitter* e perda de pacotes são semelhantes na variação entre as janelas. O comportamento da métrica de *throughput* é inversa aos demais, apresentando um diminuição do volume de dados nos fluxos quando acontece a diferenciação de tráfego.

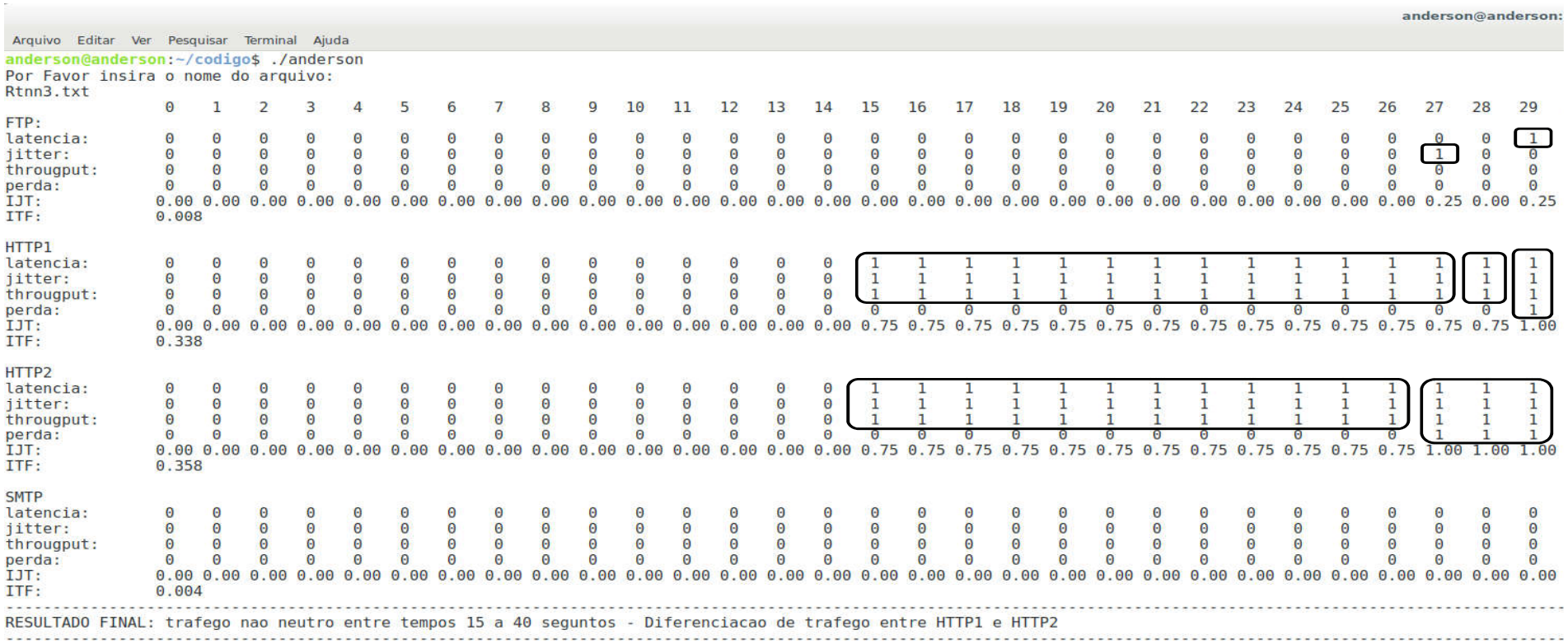
A avaliação para o tráfego capturado no cénario de configuração de *traffic shaping* é positiva, pois o resultado do método foi semelhante aos resultados dos gráficos. Entretanto, através do método foi diagnosticado 1 falso-negativo (0,10%) e 959 verdadeiros-negativos (99,90%). Representando assim mais uma vez a eficácia do método proposto.

6.4 ROTEADOR NÃO-NEUTRO 3 (RTNN3)

Neste último cenário avaliado, no qual o roteador está configurado para realizar *delay* em dois fluxos apenas. O atraso na taxa de envio de pacotes acontece somente após o roteador receber mais de 26.000 pacotes dos fluxos HTTP1 e HTTP2, a partir desse momento os fluxos sofrem uma discrimição em seu tráfego. O roteador volta a ter configuração apenas de encaminhamento, após ultrapassar a taxa de 34.000 pacotes recebidos pelo roteador dos fluxos HTTP1 e HTTP2. Os demais fluxos do tráfego não são afetados pela configuração. O resultado do método é demonstrado nas Figuras 29 e 30. A implementação do roteador no CLICK MODULAR ROUTER está descrita no Apêndice D.

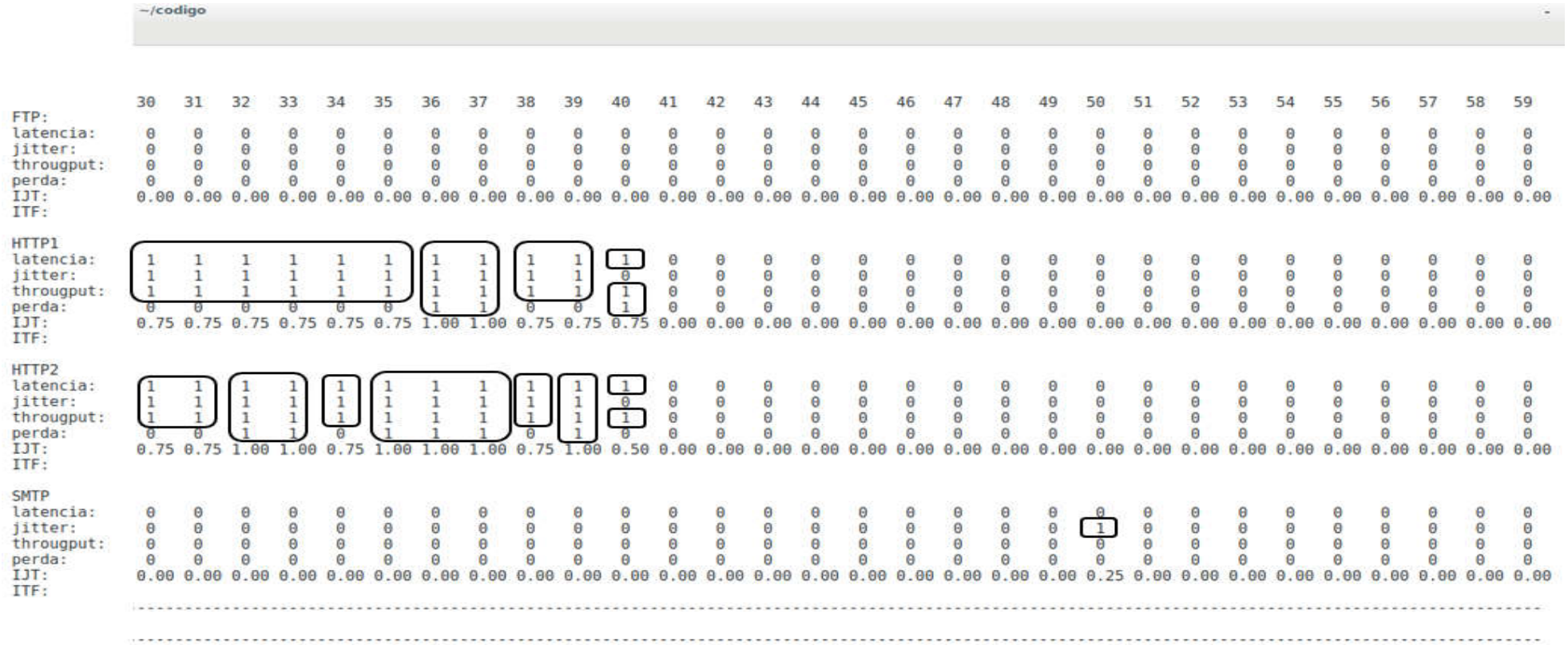
Constata-se que ocorreu três Falsos-Negativos (0,31%), dois no fluxo FTP na métrica de latência na janela de tempo de 29 segundos e outro na métrica do *jitter* na janela de tempo de 27 segundos. E no fluxo SMTP na métrica *jitter* na janela de tempo de 50 segundos. Consequentemente, ocorreu 99,69% de verdadeiros-negativos.

Figura 29 – Tela de visualização do método agnóstico para a configuração RtNN3 - Primeiros 30 segundos



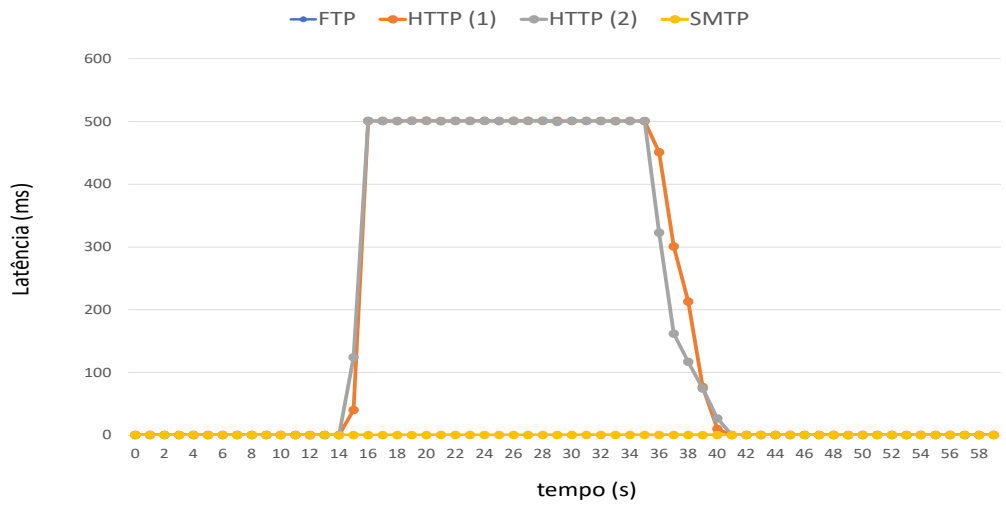
Fonte: acervo pessoal.

Figura 30 – Tela de visualização do método agnóstico para a configuração RtNN3 - Últimos 30 segundos



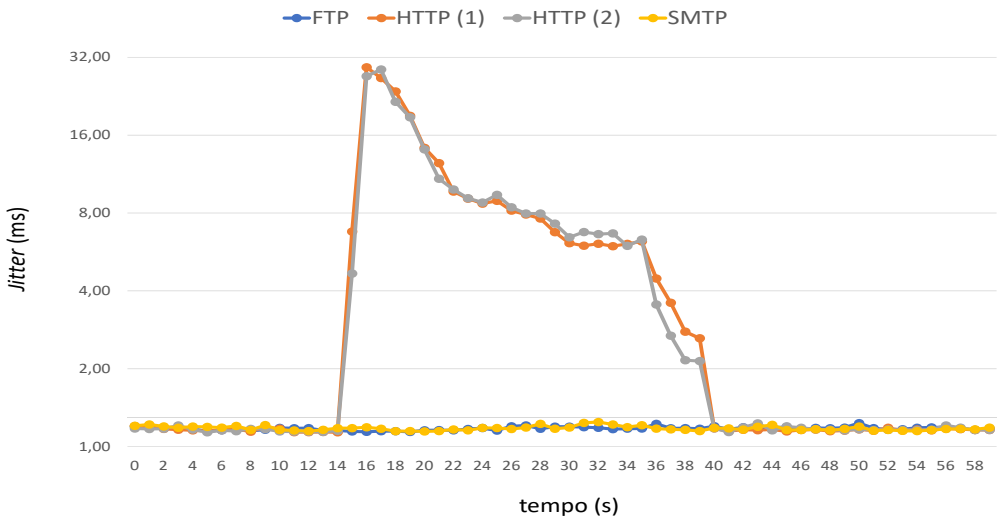
Fonte: acervo pessoal.

Figura 31 – Latência do tráfego no roteador que realiza *delay*



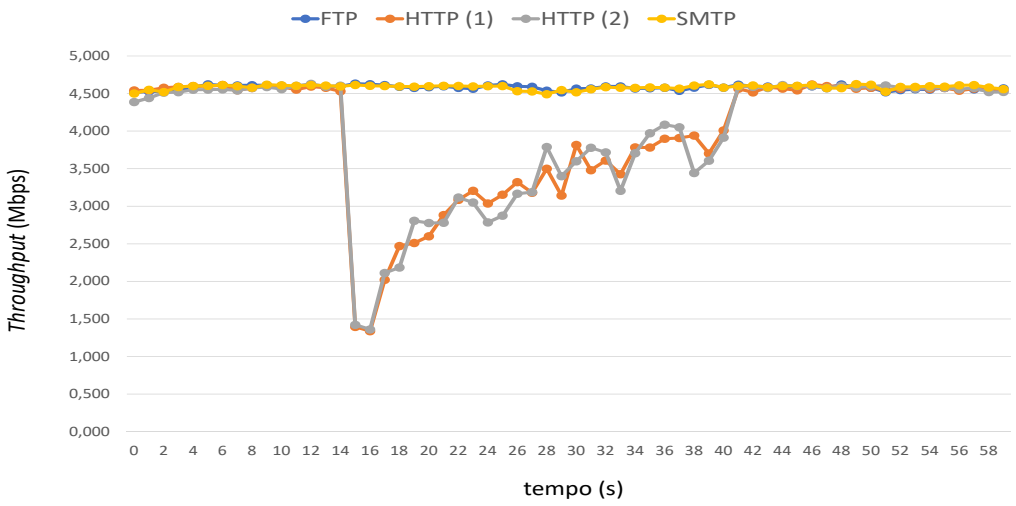
Fonte: acervo pessoal.

Figura 32 – *Jitter* do tráfego no roteador que realiza *delay*



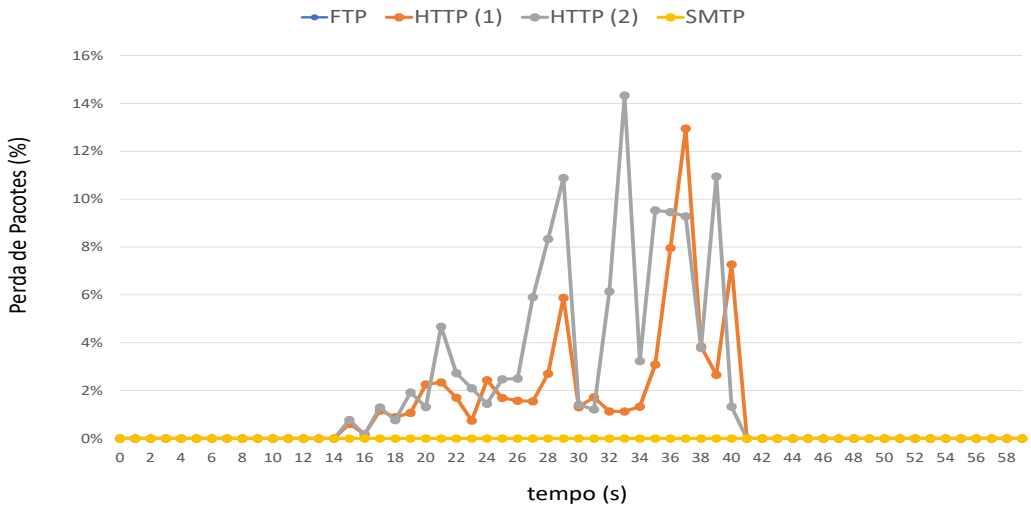
Fonte: acervo pessoal.

Figura 33 – *Throughput* do tráfego no roteador que realiza *delay*



Fonte: acervo pessoal.

Figura 34 – Perda de pacotes do tráfego no roteador que realiza *delay*



Fonte: acervo pessoal.

Analisando os resultados deste cenário, verificou-se a existência de relação entre as métricas de desempenho quando um roteador está configurado para *delay*. As métricas de desempenho *jitter* e *throughput* possuem seus valores relacionados de maneira oposta, conforme aumenta o *throughput* diminui os valores do *jitter*. Em relação a métrica de latência, ocorre

uma constante em seus valores, diminuindo apenas quando a diferenciação por *delay* está por finalizar. E a métrica de perda de pacotes tem sua característica de alternância no início da diferenciação e no final uma alta taxa de perda.

Concluindo, observou-se a eficácia do método prosposto nesta dissertação na classificação de tráfego neutro ou não-neutro, verificando-se que no conjuto de todos os cenários obtiveram-se uma acurácia de 99,84%. Deste modo, considera-se o método altamente satisfatório para aferir a existência de diferenciação de tráfego dentro do mesmo fluxo.

Uma contribuição secundária deste trabalho é a observação da relação entre métricas de desempenho quando ocorre a diferenciação de tráfego na rede, conforme Tabela 7. Assim, sabendo-se como é o comportamento das métricas de desempenho tem-se um entendimento em saber qual tipo de política que o ISP pode estar utilizando no momento da captura do tráfego.

Tabela 7 – Tabela de caraterísticas das métricas

	Latência	***Jitter***	***Throughput***	**Perda**
Neutro	valores próximos e com pouca variação entre eles	valores próximos e com pouca variação entre eles	valores próximos e com pouca variação entre eles	valores próximos e com pouca variação entre eles
Drop	valores com alta taxa de variação	valores com alta taxa de variação	valores com alta taxa de variação na redução da velocidade dos pacotes	valores com alta taxa de variação
Traffic Shaping	valores elevados com pouca variação entre os valores	valores elevados com pouca variação entre os valores	valores elevados com pouca variação entre os valores	valores elevados com pouca variação entre os valores
Delay	mantem-se constante durante a diferenciação	inicia-se elevado e gradativamente reduz	inicia-se com uma baixa velocidade e gradativamente aumenta	inicia-se sem perda considerável depois alternadamente eleva sua taxa de perda

Fonte. Autor

6.5 DISCUSSÃO

Com aplicação do método nos 4 cenários implementados nesta pesquisa, é possível fazer uma discussão sobre comportamento das métricas quando submetidas a políticas de configuração de um ISP.

Na comunicação neutra o roteador não discrimina nenhum tipo de pacote, assim os valores das métricas são próximos com uma variação menos acentuada entre os valores da amostra por janela de tempo. Pode-se comprovar tal afirmação observando as Figuras 14, 15 e 16. Caso ocorra algum tipo de discriminação, será aleatória e não específica, sendo assim todos os fluxos são afetados com alterações nos valores das métricas, independente do tipo de fluxo, da porta de origem ou destino ou qualquer outra características. Porque nesta situação o roteador não tem prioridade em agir de alguma maneira tendenciosa contra pacotes específicos.

No cenário de descartes de pacotes, fica nítido que as diferenças são elevadas entre o menor valor e o maior valor de cada métrica dentro da diferenciação do tráfego. Os valores possuem alta taxa de variação entre as janelas de tempo. Quanto maior for a porcentagem da perda de pacotes, mais comprometido está as outras métricas.

Diferentemente do cenário anterior, o cenário avaliado na seção 6.3, que realiza *traffic shaping* possui os valores das métricas mais próximos quando está ocorrendo diferenciação de tráfego. Consequentemente existe pouca variação entre os valores das amostras por janela de tempo.

O último cenário avaliado, seção 6.4, trás a simulação de *delay* pelo roteador. Nesta situação a métrica de latência permanece constante durante todo o tempo o roteador implementa a diferenciação de tráfego. Outra observação importante acontece na métrica de perda de pacotes, onde ocorre no inicio da diferenciação a perda com porcentagens baixas até um ponto que a porcentagem aumenta significativamente durante um instante de tempo apenas.

Em relação a métrica de perda de pacotes também é possível notar que quando há uma discriminação nessa métrica sempre haverá outra métrica sofrendo discriminação também. Tal característica faz com que a métrica seja importante na identificação da quebra da neutralidade.

7 CONCLUSÃO

Nesta dissertação foi proposto um método agnóstico de identificação da quebra da neutralidade pelos ISPs. Este método permitiu, através da análise de várias métricas de desempenho (latência, *jitter*, *throughput* e perda de pacotes) que seja aferida a diferenciação de tráfego. Além disso, o método desenvolvido consegue distinguir quando ocorre uma diferenciação entre os fluxos de quando acontece uma degradação dos fluxos. Apresentou-se também uma contribuição sobre a relação entre as métricas de desempenho quando ocorre a diferenciação de tráfego na rede.

O ambiente desenvolvido nesta pesquisa utilizou-se do D-ITG para gerar um tráfego realístico a ser processado pelo roteador. Também utilizou-se o Wireshark para capturas de tráfego para análise das métricas. O roteador nos cenários avaliados tinha como função simular um ISP. As configurações executadas no roteador foram: implementar um roteador neutro para todo o tráfego de dados; e configurar roteadores neutros para um fluxo de dados e após uma certa quantidade de pacotes o roteador começa a discriminar por alguns fatores outros fluxos. Para simular a discriminação de fluxos foram configurados três tipos de roteadores que fizeram a diferenciação de um fluxo específico para descarte de pacotes, *traffic shaping* e *delay*. Utilizou-se o CLICK MODULAR ROUTER para configurar o roteador nos cenários.

Com a aplicação do método agnóstico foi viável identificar claramente as discriminações por aplicação e correlacionar com os demais fluxos capturados no tráfego de dados analisado. Assim como, foi possível a identificação em relação ao tempo que ocorre tais discriminações e quais métricas que foram afetadas por este comportamento.

Finalmente, com base nas conclusões obtidas com a utilização do método nos 4 cenários propostos, pode-se concluir que de fato o método se mostrou eficaz na identificação da quebra da neutralidade. O diferencial desta pesquisa, e principal contribuição em relação aos outros trabalhos, foi a utilização de quatro métricas de desempenho para analisar os fluxos e identificar diferenciação do tráfego, e também a correlação existente entre as métricas de desempenho quando submetidas à políticas de configuração de um ISP.

Como trabalhos futuros, fica a proposição de implementar o método proposto em um ambiente real. Também, a de configurar novos cenários com novas implementações de diferenciação de tráfego, além da geração de novos resultados nesses mesmos cenários, porém modificando os valores para descarte de pacotes, *traffic shaping* e *delay*.

REFERÊNCIAS

AHMED, M.; MAHMOOD, A. N.; HU, J. A survey of network anomaly detection techniques. **Journal of Network and Computer Applications**, [S.l.], v.60, p.19–31, 2016.

AUSTEN, I. **A Canadian Telecom's Labor Dispute Leads to Blocked Web Sites and Questions of Censorship**. Acesso: setembro de 2017, Disponível em: <http://www.nytimes.com/2005/08/01/business/worldbusiness/a-canadian-telecoms-labor-dispute-leads-to-blocked.html>.

AVALLONE, S. et al. D-ITG distributed internet traffic generator. In: QUANTITATIVE EVALUATION OF SYSTEMS, 2004. QEST 2004. PROCEEDINGS. FIRST INTERNATIONAL CONFERENCE ON THE. **Anais...** [S.l.: s.n.], 2004. p.316–317.

AXELSSON, S. **Intrusion detection systems**: a survey and taxonomy. [S.l.]: Technical report, 2000.

BASSO, S.; SERVETTI, A.; DE MARTIN, J. C. The network neutrality bot architecture: a preliminary approach for self-monitoring of internet access qos. In: COMPUTERS AND COMMUNICATIONS (ISCC), 2011 IEEE SYMPOSIUM ON. **Anais...** [S.l.: s.n.], 2011. p.1131–1136.

BEREC. A framework for Quality of Service in the scope of Net Neutrality. In: **Anais...** Body of European Regulators for Electronic Communications, 2011.

BOTTA, A. et al. **D-ITG 2.8.1 Manual**. Acesso: abril de 2017, Disponível em: <http://traffic.comics.unina.it/software/ITG/manual/>.

BRADNER, S.; MCQUAID, J. **Benchmarking Methodology for Network Interconnect Devices**. [S.l.]: RFC Editor, 1999. RFC. (2544).

BRODKIN, J. **Netflix performance on Verizon and Comcast has been dropping for months**. Acesso: setembro de 2017, Disponível em: <https://arstechnica.com/information-technology/2014/02/netflix-performance-on-verizon-and-comcast-has-been-dropping-for-months/>.

BUSTOS-JIMÉNEZ, J.; FUENZALIDA, C. All Packets Are Equal, but Some Are More Equal Than Others. In: LATIN AMERICA NETWORKING CONFERENCE ON LANC 2014, New York, NY, USA. **Proceedings...** ACM, 2014. p.5:1–5:8. (LANC '14).

CALLADO, A. et al. A survey on internet traffic identification. **IEEE communications surveys & tutorials**, [S.l.], v.11, n.3, 2009.

CISCO. **Cisco prevê triplicação do tráfego IP entre 2014 e 2019.** Acesso: dezembro de 2016, Disponível em: <https://www.cisco.com/c/pt_pt/about/press/news-archive-2015/20150527.html>.

CLAFFY, K. C.; BRAUN, H.-W.; POLYZOS, G. C. A parameterizable methodology for Internet traffic flow profiling. **IEEE Journal on selected areas in communications**, [S.l.], v.13, n.8, p.1481–1494, 1995.

COMMISSION, F. C.; COMMISSION, F. C. et al. Open internet order. **http://www.fcc.gov/openinternet**, [S.l.], 2015.

COMMISSION, F. C. et al. 47 CFR Parts 0 and 8. GN Docket No. 09–191; WC Docket No. 07–52; FCC 10–201. Preserving the Open Internet. Final Rule. **Federal Register**, [S.l.], v.76, p.59192–59235, 2011.

COSTA, A. F. B.; EPPRECHT, E. K.; CARPINETTI, L. C. R. **Controle estatístico de qualidade**. [S.l.]: Atlas São Paulo, 2005.

CROWCROFT, J. Net neutrality: the technical side of the debate: a white paper. **ACM SIGCOMM Computer Communication Review**, [S.l.], v.37, n.1, p.49–56, 2007.

DECKER, C.; EIDENBENZ, R.; WATTENHOFER, R. Exploring and improving bittorrent topologies. In: PEER-TO-PEER COMPUTING (P2P), 2013 IEEE THIRTEENTH INTERNATIONAL CONFERENCE ON. **Anais...** [S.l.: s.n.], 2013. p.1–10.

DEWAELE, G. et al. Extracting hidden anomalies using sketch and non gaussian multiresolution statistical detection procedures. In: LARGE SCALE ATTACK DEFENSE, 2007. **Proceedings...** [S.l.: s.n.], 2007. p.145–152.

DISCHINGER, M. et al. Detecting bittorrent blocking. In: ACM SIGCOMM CONFERENCE ON INTERNET MEASUREMENT, 8. **Proceedings...** [S.l.: s.n.], 2008. p.3–8.

DISCHINGER, M. et al. Glasnost: enabling end users to detect traffic differentiation. In: NSDI. **Anais...** [S.l.: s.n.], 2010. p.405–418.

ESNAASHARI, S. nvisible Barriers: identifying restrictions affecting new zealanders access to the internet. **Victoria University of Wellington**, [S.l.], 2014.

FLACH, T. et al. An Internet-wide analysis of traffic policing. In: ACM SIGCOMM 2016 CONFERENCE, 2016. **Proceedings...** [S.l.: s.n.], 2016. p.468–482.

GROVE, N.; AGIC, D.; SEDLMEIR, J. Network neutrality and consumer discrimination: comparing isp's gtcs and dpi application. In: OF THE 2016 CONFERENCE ON , Wien. **Anais...** ITS, 2012.

HOSEIN, P. et al. Detecting network neutrality violations through packet loss statistics. In: NETWORK OPERATIONS AND MANAGEMENT SYMPOSIUM (APNOMS), 2015 17TH ASIA-PACIFIC. **Anais...** [S.l.: s.n.], 2015. p.404–407.

IDEC. **dec notifica Claro sobre possível discriminação de tráfego de dados do Pokémon GO**. Acesso: setembro de 2017, Disponível em: <https://idec.org.br/em-acao/em-foco/idec-notifica-claro-sobre-possivel-discriminaco-de-trafego-de-dados-do-pokemon-go>.

IPERF. **iPerf - The ultimate speed test tool for TCP, UDP and SCTP**. Acesso: outubro de 2017, Disponível em: <https://iperf.fr/>.

JAMHOUR, E. **Qualidade de Serviços em Redes IP**. Acesso: agosto de 2016, Disponível em: <https://www.ppgia.pucpr.br/ jamhour/Pessoal/Mestrado/TARC/QoSIP.pdf>.

JORDAN, S. [Special Section on Net Neutrality] A Layered Network Approach to Net Neutrality. **International Journal of Communication**, [S.l.], v.1, n.1, p.34, 2007.

KANUPARTHY, P.; DOVROLIS, C. Diffprobe: detecting isp service discrimination. In: INFOCOM, 2010 PROCEEDINGS IEEE. **Anais...** [S.l.: s.n.], 2010. p.1–9.

KENDRIC, J. **T-Mobile Germany Blocks iPhone Skype Over 3G and WiFi**. Acesso: outubro de 2017, Disponível em: <https://gigaom.com/2009/04/06/t-mobile-germany-blocks-iphone-skype-over-3g-too/>.

KIVITY, A. et al. kvm: the linux virtual machine monitor. In: LINUX SYMPOSIUM. **Proceedings...** [S.l.: s.n.], 2007. v.1, p.225–230.

KOHLER, E. et al. The Click modular router. **ACM Transactions on Computer Systems (TOCS)**, [S.l.], v.18, n.3, p.263–297, 2000.

KREIBICH, C. et al. Netalyzr: illuminating the edge network. In: ACM SIGCOMM CONFERENCE ON INTERNET MEASUREMENT, 10. **Proceedings...** [S.l.: s.n.], 2010. p.246–259.

LAKHINA, A.; CROVELLA, M.; DIOT, C. Diagnosing network-wide traffic anomalies. In: ACM SIGCOMM COMPUTER COMMUNICATION REVIEW. **Anais...** [S.l.: s.n.], 2004. v.34, n.4, p.219–230.

MAHAJAN, P. S.; INGALLS, R. G. Evaluation of methods used to detect warm-up period in steady state simulation. In: WINTER SIMULATION, 36. **Proceedings...** [S.l.: s.n.], 2004. p.663–671.

MARSDEN, C. Network neutrality: a research guide. , [S.l.], 2011.

MONTGOMERY, D. C. **Introdução Ao Controle Estatístico Da Qualidade .** [S.l.]: Grupo Gen-LTC, 2000.

MUELLER, M. L.; ASGHARI, H. Deep packet inspection and bandwidth management: battles over bittorrent in canada and the united states. **Telecommunications Policy**, [S.l.], v.36, n.6, p.462–475, 2012.

OREBAUGH, A.; RAMIREZ, G.; BEALE, J. **Wireshark & Ethereal network protocol analyzer toolkit**. [S.l.]: Syngress, 2006.

PATHANIA, A.; KALRA, P. Network Neutrality Survey. **Indian Journal of Computer Science and Engineering (IJCSE)**, [S.l.], v.3, n.1, 2012.

PEITZ, M.; SCHUETT, F. Net neutrality and inflation of traffic. **International Journal of Industrial Organization**, [S.l.], v.46, p.16–62, 2016.

RAVAIOLI, R.; BARAKAT, C.; URVOY-KELLER, G. Chkdiff: checking traffic differentiation at internet access. In: ACM CONFERENCE ON CONEXT STUDENT WORKSHOP, 2012. **Proceedings...** [S.l.: s.n.], 2012. p.57–58.

RESPECTMYNET. **Respect My Net - Report cases of Net Neutrality violations**. Acesso: outubro de 2017, Disponível em: <https://respectmynet.eu/about/>.

ROBINSON, S. A statistical process control approach for estimating the warm-up period. In: SIMULATION CONFERENCE, 2002. PROCEEDINGS OF THE WINTER. **Anais...** [S.l.: s.n.], 2002. v.1, p.439–446.

SANTOS, V. W. O. et al. Neutralidade da rede e o Marco Civil da Internet no Brasil: atores, políticas e controvérsias. , [S.l.], 2016.

TARIQ, M. B. et al. Detecting network neutrality violations with causal inference. In: EMERGING NETWORKING EXPERIMENTS AND TECHNOLOGIES, 5. **Proceedings...** [S.l.: s.n.], 2009. p.289–300.

TOPOLSKI, R. **Comcast is using Sandvine to manage P2P Connections**. Acesso: outubro de 2017, Disponível em: <http://www.dslreports.com/forum/r18323368-Comcast-is-using-Sandvine-to-manage-P2P-Connections>.

VAN SCHEWICK, B. **Internet architecture and innovation**. [S.l.]: MIT Press, 2012.

VAN SCHEWICK, B. Network neutrality and quality of service: what a nondiscrimination rule should look like. **Stan. L. Rev.**, [S.l.], v.67, p.1, 2015.

WILLE, E. C. G.; TENORIO, M. M. Considering packet loss probability in fault-tolerant OSPF routing. **IEEE Latin America Transactions**, [S.l.], v.12, n.2, p.248–255, 2014.

WIRESHARK. **Wireshark**. Acesso: março de 2017, Disponível em: <https://www.wireshark.org//>.

WU, T. Network neutrality, broadband discrimination. **J. on Telecomm. & High Tech. L.**, [S.l.], v.2, p.141, 2003.

YOO, C. S. Wickard for the internet? network neutrality after Verizon v. FCC. In: OF THE 5TH INTERNATIONAL CONFERENCE ON . **Anais...** Federal Communications Law Journal, 2014.

ZHANG, J. et al. Detecting anomalies from big network traffic data using an adaptive detection approach. **Information Sciences**, [S.l.], v.318, p.91–110, 2015.

ZHANG, Y.; MAO, Z. M.; ZHANG, M. Detecting traffic differentiation in backbone ISPs with NetPolice. In: ACM SIGCOMM CONFERENCE ON INTERNET MEASUREMENT CONFERENCE, 9. **Proceedings...** [S.l.: s.n.], 2009. p.103–115.

ZHANG, Z.; MARA, O.; ARGYRAKI, K. Network neutrality inference. In: ACM SIGCOMM COMPUTER COMMUNICATION REVIEW. **Anais...** [S.l.: s.n.], 2014. v.44, n.4, p.63–74.

APÊNDICES

APÊNDICE A – Configuração de Roteador Neutro

Neste apêndice é apresentado a configuração realizada na máquina virtual que possui o papel de roteador no cenário neutro. A implementação foi realizada através da linguaguem CLICK MODULAR ROUTER.

```
1  //BRIDGEs
2  // br1 - eth0 192.168.121.1 52:54:00:76:4c:7c
3  // br2 - eth1 192.168.122.1 52:54:00:bc:7a:22
4
5  net0i :: FromDevice(eth0);
6  net0o :: ToDevice(eth0);
7
8  net1i :: FromDevice(eth1);
9  net1o :: ToDevice(eth1);
10
11 net0q :: Queue();
12 net1q :: Queue();
13
14 //ARP Queries - ARP Replices - IP - outros
15 c0 :: Classifier(12/0806 20/0001,
16                  12/0806 20/0002,
17                  12/0800,
18                  -);
19 c1 :: Classifier(12/0806 20/0001,
20                  12/0806 20/0002,
21                  12/0800,
22                  -);
23
24 c0[3] -> Discard;
25 c1[3] -> Discard;
26
27 net0i -> [0]c0;
28 net1i -> [0]c1;
29
30 //Tratar os pedidos de ARP nas interfaces
31
32 arpq0 :: ARPQuerier(192.168.121.1, 52:54:00:76:4c:7c);
33 arpq1 :: ARPQuerier(192.168.122.1, 52:54:00:bc:7a:22);
34
35 t :: Tee(2);
36 c0[1] -> t;
37 c1[1] -> t;
38 t[0] -> [1]arpq0;
39 t[1] -> [1]arpq1;
40
41 arpq0 -> net0q -> net0o;
42 arpq1 -> net1q -> net1o;
```

```
43
44 arpr0 :: ARPResponder(192.168.121.1 52:54:00:76:4c:7c);
45 arpr1 :: ARPResponder(192.168.122.1 52:54:00:bc:7a:22);
46
47 c0[0] -> arpr0 -> net0q -> net0o;
48 c1[0] -> arpr1 -> net1q -> net1o;
49
50 rt :: StaticIPLookup(192.168.121.0/24 0, 192.168.122.0/24 1);
51
52 ip :: Strip(14) -> CheckIPHeader(INTERFACES 192.168.121.1/24
      192.168.122.1/24) -> [0]rt;
53 c0[2]-> ip;
54 c1[2]-> ip;
55
56 rt[0] -> DropBroadcasts -> FixIPSrc(192.168.121.1) -> [0]arpq0;
57 rt[1] -> DropBroadcasts -> FixIPSrc(192.168.122.1) -> [0]arpq1;
```

APÊNDICE B – Configuração de Roteador de realiza descarte de pacotes

Neste apêndice é apresentado a configuração realizada na máquina virtual que possui o papel de roteador no cenário não-neutro1 (RtNN1). A implementação foi realizada através da linguaguem CLICK MODULAR ROUTER. O roteador está configurado para ser neutro para todos os fluxos até passagem de 24.000 pacotes do fluxo HTTP1 e HTTP2. Quando ultrapassar tal limite definido, o roteador começa a descartar pacotes somente desses fluxos em uma taxa não determinística próxima de 20%. O roteador volta a ser neutro para os fluxos HTTP1 e HTTP2 quando ultrapassar a taxa de 26.000 pacotes dos fluxos em questão.

```
1  //BRIDGEs
2  // br1 - eth0 192.168.121.1 52:54:00:76:4c:7c
3  // br2 - eth1 192.168.122.1 52:54:00:bc:7a:22
4
5  net0i :: FromDevice(eth0);
6  net0o :: ToDevice(eth0);
7
8  net1i :: FromDevice(eth1);
9  net1o :: ToDevice(eth1);
10
11 net0q :: Queue();
12 net1q :: Queue();
13
14 //ARP Queries - ARP Replices - IP - outros
15 c0 :: Classifier(12/0806 20/0001,
16                  12/0806 20/0002,
17                  12/0800,
18                  -);
19
20 c1 :: Classifier(12/0806 20/0001,
21                   12/0806 20/0002,
22                   12/0800,
23                   -);
24
25 c0[3] -> Discard;
26 c1[3] -> Discard;
27
28 net0i -> [0]c0;
29 net1i -> [0]c1;
30
31 //Tratar os pedidos de ARP nas interfaces
32 arpq0 :: ARPQuerier(192.168.121.1, 52:54:00:76:4c:7c);
33 arpq1 :: ARPQuerier(192.168.122.1, 52:54:00:bc:7a:22);
34
35 t :: Tee(2);
36 c0[1] -> t;
37 c1[1] -> t;
```

```
38 t[0] -> [1]arpq0;
39 t[1] -> [1]arpq1;
40
41 arpq0 -> net0q -> net0o;
42 arpq1 -> net1q -> net1o;
43
44 arpr0 :: ARPResponder(192.168.121.1 52:54:00:76:4c:7c);
45 arpr1 :: ARPResponder(192.168.122.1 52:54:00:bc:7a:22);
46
47 c0[0] -> arpr0 -> net0q -> net0o;
48 c1[0] -> arpr1 -> net1q -> net1o;
49
50 rt :: StaticIPLookup(
51           192.168.121.0/24 0,
52           192.168.122.0/24 1);
53
54 ipclass :: IPClassifier (dst tcp port 21,
55                          dst tcp port 25,
56                          dst tcp port 80,
57                          dst 192.168.122.22 tcp port 80, -);
58
59 ip :: Strip(14) -> CheckIPHeader(INTERFACES 192.168.121.1/24
      192.168.122.1/24) -> ipclass;
60
61 //Porta 21
62 ipclass[0] -> [0]rt;
63
64 //Porta 25
65 ipclass[1] -> [0]rt;
66
67 swt :: Switch();
68
69 //Porta 80 192.168.122.20
70 ipclass[2]-> Counter(COUNT_CALL 24000 swt.switch 1) -> Counter(
      COUNT_CALL 26000 swt.switch 2) -> [0]swt;
71 net1i -> [0]c1;
72
73 //Roteador Neutro - Pkt 0 ate 24000
74 swt[0] -> [0]rt;
75
76 //RtNN1 - Pkt 24000 ate 26000
77 swt[1] -> RandomSample(0.8) -> [0]rt;
78
79 //Roteador Neutro - 26000 >
80 swt[2] -> [0]rt;
81
82 swtc :: Switch();
83
84 //Porta 80 192.168.122.22
85 ipclass[3]-> Counter(COUNT_CALL 24000 swtc.switch 1) -> Counter(
      COUNT_CALL 24000 swtc.switch 2) -> [0]swtc;
```

```
86  net1i -> [0]c1;
87
88  //Roteador Neutro - Pkt 0 ate 24000
89  swtc[0] -> [0]rt;
90
91  //RtNN1 - Pkt 24000 ate 26000
92  swtc[1] -> RandomSample(0.8) -> [0]rt;
93
94  //Roteador Neutro - 26000 >
95  swtc[2] -> [0]rt;
96
97  //Restante
98  ipclass[4] -> [0]rt;
99
100 c0[2]-> ip;
101 c1[2]-> ip;
102
103 rt[0] -> DropBroadcasts -> FixIPSrc(192.168.121.1) -> [0]arpq0;
104 rt[1] -> DropBroadcasts -> FixIPSrc(192.168.122.1) -> [0]arpq1;
```

APÊNDICE C – Configuração de Roteador de realiza *traffic shaping* de pacotes

Neste apêndice é apresentado a configuração realizada na máquina virtual que possui o papel de roteador no cenário não-neutro2 (RtNN2). A implementação foi realizada através da linguaguem CLICK MODULAR ROUTER. O roteador está configurado para ser neutro para todos os fluxos até passagem de 26.000 pacotes do fluxo HTTP1 e HTTP2. Quando ultrapassar tal limite definido, o roteador começa a reduzir a taxa de envio de pacotes somente desses fluxos em uma taxa 400 pacotes por segundo. O roteador volta a ser neutro para os fluxos HTTP1 e HTTP2 quando ultrapassar a taxa de 34.000 pacotes dos fluxos em questão.

```
1  //BRIDGEs
2  // br1 - eth0 192.168.121.1 52:54:00:76:4c:7c
3  // br2 - eth1 192.168.122.1 52:54:00:bc:7a:22
4
5  net0i :: FromDevice(eth0);
6  net0o :: ToDevice(eth0);
7
8  net1i :: FromDevice(eth1);
9  net1o :: ToDevice(eth1);
10
11 net0q :: Queue();
12 net1q :: Queue();
13
14 //ARP Queries - ARP Replices - IP - outros
15 c0 :: Classifier(12/0806 20/0001,
16                  12/0806 20/0002,
17                  12/0800,
18                  -);
19
20 c1 :: Classifier(12/0806 20/0001,
21                   12/0806 20/0002,
22                   12/0800,
23                   -);
24
25 c0[3] -> Discard;
26 c1[3] -> Discard;
27
28 net0i -> [0]c0;
29 net1i -> [0]c1;
30
31 //Tratar os pedidos de ARP nas interfaces
32 arpq0 :: ARPQuerier(192.168.121.1, 52:54:00:76:4c:7c);
33 arpq1 :: ARPQuerier(192.168.122.1, 52:54:00:bc:7a:22);
34
35 t :: Tee(2);
```

```
36 c0[1] -> t;
37 c1[1] -> t;
38 t[0] -> [1]arpq0;
39 t[1] -> [1]arpq1;
40
41 arpq0 -> net0q -> net0o;
42 arpq1 -> net1q -> net1o;
43
44 arpr0 :: ARPResponder(192.168.121.1 52:54:00:76:4c:7c);
45 arpr1 :: ARPResponder(192.168.122.1 52:54:00:bc:7a:22);
46
47 c0[0] -> arpr0 -> net0q -> net0o;
48 c1[0] -> arpr1 -> net1q -> net1o;
49
50 rt :: StaticIPLookup(
51           192.168.121.0/24 0,
52           192.168.122.0/24 1);
53
54 ipclass :: IPClassifier (dst tcp port 21,
55                          dst tcp port 25,
56                          dst tcp port 80,
57                          dst 192.168.122.22 tcp port 80, -);
58
59 ip :: Strip(14) -> CheckIPHeader(INTERFACES 192.168.121.1/24
      192.168.122.1/24) -> ipclass;
60
61 //Porta 21
62 ipclass[0] -> [0]rt;
63
64 //Porta 25
65 ipclass[1] -> [0]rt;
66
67 swt :: Switch();
68
69 //Porta 80 192.168.122.20
70 ipclass[2]-> Counter(COUNT_CALL 26000 swt.switch 1) -> Counter(
      COUNT_CALL 34000 swt.switch 2) -> [0]swt;
71 net1i -> [0]c1;
72
73 //Roteador Neutro - Pkt 0 ate 26000
74 swt[0] -> [0]rt;
75
76 //RtNN1 - Pkt 24000 ate 34000
77 swt[1] -> RatedSplitter(400) -> [0]rt;
78
79 //Roteador Neutro - 34000 >
80 swt[2] -> [0]rt;
81
82 swtc :: Switch();
83
84 //Porta 80 192.168.122.22
```

```
85  ipclass[3]-> Counter(COUNT_CALL 26000 swtc.switch 1) -> Counter(
        COUNT_CALL 34000 swtc.switch 2) -> [0]swtc;
86  net1i -> [0]c1;
87
88  //Roteador Neutro - Pkt 0 ate 26000
89  swtc[0] -> [0]rt;
90
91  //RtNN1 - Pkt 26000 ate 34000
92  swtc[1] -> RatedSplitter(400) -> [0]rt;
93
94  //Roteador Neutro - 34000 >
95  swtc[2] -> [0]rt;
96
97  //Restante
98  ipclass[4] -> [0]rt;
99
100 c0[2]-> ip;
101 c1[2]-> ip;
102
103 rt[0] -> DropBroadcasts -> FixIPSrc(192.168.121.1) -> [0]arpq0;
104 rt[1] -> DropBroadcasts -> FixIPSrc(192.168.122.1) -> [0]arpq1;
```

APÊNDICE D – Configuração de Roteador de realiza *delay* de pacotes

Neste apêndice é apresentado a configuração realizada na máquina virtual que possui o papel de roteador no cenário não-neutro3 (RtNN3). A implementação foi realizada através da linguaguem CLICK MODULAR ROUTER. O roteador está configurado para ser neutro para todos os fluxos até passagem de 26.000 pacotes do fluxo HTTP1 e HTTP2. Quando ultrapassar tal limite definido, o roteador começa a atrasar o envio dos pacotes somente desses fluxos à uma taxa de 0,5 segundos de *delay* por pacotes. O roteador volta a ser neutro para os fluxos HTTP1 e HTTP2 quando ultrapassar a taxa de 34.000 pacotes dos fluxos em questão.

```
1  //BRIDGEs
2  // br1 - eth0 192.168.121.1 52:54:00:76:4c:7c
3  // br2 - eth1 192.168.122.1 52:54:00:bc:7a:22
4
5  net0i :: FromDevice(eth0);
6  net0o :: ToDevice(eth0);
7
8  net1i :: FromDevice(eth1);
9  net1o :: ToDevice(eth1);
10
11 net0q :: Queue();
12 net1q :: Queue();
13
14 //ARP Queries - ARP Replices - IP - outros
15 c0 :: Classifier(12/0806 20/0001,
16                  12/0806 20/0002,
17                  12/0800,
18                  -);
19
20 c1 :: Classifier(12/0806 20/0001,
21                   12/0806 20/0002,
22                   12/0800,
23                   -);
24
25 c0[3] -> Discard;
26 c1[3] -> Discard;
27
28 net0i -> [0]c0;
29 net1i -> [0]c1;
30
31 //Tratar os pedidos de ARP nas interfaces
32 arpq0 :: ARPQuerier(192.168.121.1, 52:54:00:76:4c:7c);
33 arpq1 :: ARPQuerier(192.168.122.1, 52:54:00:bc:7a:22);
34
35 t :: Tee(2);
36 c0[1] -> t;
37 c1[1] -> t;
```

```
38 t[0] -> [1]arpq0;
39 t[1] -> [1]arpq1;
40
41 arpq0 -> net0q -> net0o;
42 arpq1 -> net1q -> net1o;
43
44 arpr0 :: ARPResponder(192.168.121.1 52:54:00:76:4c:7c);
45 arpr1 :: ARPResponder(192.168.122.1 52:54:00:bc:7a:22);
46
47 c0[0] -> arpr0 -> net0q -> net0o;
48 c1[0] -> arpr1 -> net1q -> net1o;
49
50 rt :: StaticIPLookup(
51           192.168.121.0/24 0,
52           192.168.122.0/24 1);
53
54 ipclass :: IPClassifier (dst tcp port 21,
55                          dst tcp port 25,
56                          dst tcp port 80,
57                          dst 192.168.122.22 tcp port 80, -);
58
59 ip :: Strip(14) -> CheckIPHeader(INTERFACES 192.168.121.1/24
      192.168.122.1/24) -> ipclass;
60
61 //Porta 21
62 ipclass[0] -> [0]rt;
63
64 //Porta 25
65 ipclass[1] -> [0]rt;
66
67 swt :: Switch();
68
69 //Porta 80 192.168.122.20
70 ipclass[2]-> Counter(COUNT_CALL 26000 swt.switch 1) -> Counter(
      COUNT_CALL 34000 swt.switch 2) -> [0]swt;
71 net1i -> [0]c1;
72
73 //Roteador Neutro - Pkt 0 ate 26000
74 swt[0] -> [0]rt;
75
76 //RtNN1 - Pkt 26000 ate 34000
77 swt[1] -> Queue() -> DelayUnqueue(0.5) -> [0]rt;
78
79 //Roteador Neutro - 34000 >
80 swt[2] -> [0]rt;
81
82 swtc :: Switch();
83
84 //Porta 80 192.168.122.22
85 ipclass[3]-> Counter(COUNT_CALL 26000 swtc.switch 1) -> Counter(
      COUNT_CALL 34000 swtc.switch 2) -> [0]swtc;
```

```
86  net1i -> [0]c1;
87
88  //Roteador Neutro - Pkt 0 ate 26000
89  swtc[0] -> [0]rt;
90
91  //RtNN1 - Pkt 26000 ate 34000
92  swtc[1] -> Queue () -> DelayUnqueue(0.5) -> [0]rt;
93
94  //Roteador Neutro - 34000 >
95  swtc[2] -> [0]rt;
96
97  //Restante
98  ipclass[4] -> [0]rt;
99
100 c0[2]-> ip;
101 c1[2]-> ip;
102
103 rt[0] -> DropBroadcasts -> FixIPSrc(192.168.121.1) -> [0]arpq0;
104 rt[1] -> DropBroadcasts -> FixIPSrc(192.168.122.1) -> [0]arpq1;
```