Explorando Propriedades do Roteamento na Internet para a Localização de Diferenciação de Tráfego

Thiago Garrett[1], **Luis C. E. Bona**[1], **Elias P. Duarte Jr.**[1]

[1]Departamento de Informática
Universidade Federal do Paraná – Curitiba, PR – Brasil

{tgarrett,bona,elias}@inf.ufpr.br

Abstract. *Network neutrality advocates that no traffic should ever be discriminated, either in terms of source, destination and/or content. Although several solutions for detecting traffic differentiation (TD) have been proposed, there is a lack of effective solutions for locating the source of TD. In this work, we propose a solution for detecting and locating TD that takes advantage of routing properties of Internet routing, in particular AS peering. The proposed strategy explores differences and similarities of the possible routes between measurement points. The assumptions regarding Internet routing were validated through an experiment executed on PlanetLab. We also evaluated with simulation the accuracy of our proposals.*

Resumo. *A neutralidade da rede preconiza uma rede sem discriminação de tráfego, independente de origem, destino e/ou conteúdo. Apesar de existirem diversas soluções para a detecção de diferenciação de tráfego (DT), há uma lacuna em termos da localização efetiva do ponto da rede onde a DT ocorre. Neste trabalho, propomos uma solução para a detecção e localização de DT, que tira proveito das propriedades do roteamento na Internet entre Sistemas Autônomos. A estratégia explora diferenças e semelhanças dos caminhos entre os pontos de medição. As premissas acerca do roteamento na Internet foram verificadas através de experimentos executados no PlanetLab. A acurácia e eficiência da proposta foi avaliada por meio de simulação.*

1. Introdução

O princípio da Neutralidade da Rede (NR) preconiza que todo tráfego deve ser tratado igualmente, independente da origem, destino e/ou conteúdo, ou seja, diferenciação de tráfego (DT) não é permitida [Garrett et al. 2018]. Regulações acerca da NR já foram implementadas em diversos países, inclusive o Brasil. Uma das principais motivações da NR é garantir que a Internet continue sendo um ambiente aberto que promova inovação, concorrência justa e liberdade de escolha dos usuários [Garrett et al. 2017]. Por outro lado, é notório que provedores de acesso (*Internet Service Provider*, ISP) praticam DT para aliviar situações de congestionamento, sob acordos comerciais ou mesmo para beneficiar seus próprios serviços, entre outras situações.

No entanto, regulações por si só não são suficientes para garantir a conformidade dos ISPs. Além disso, independentemente de regulações, a simples transparência nas práticas de gerência de tráfego pode contribuir para um mercado mais competitivo. O monitoramento da DT é, portanto, essencial.

Existem diversas soluções para detectar DT na Internet [Garrett et al. 2018]. Estas soluções baseiam-se em premissas diversas, apresentando assim capacidades e limitações próprias. Diferentes soluções detectam tipos diferentes de DT, empregando diferentes técnicas de medição e inferência, conforme as suposições feitas. Por outro lado, ainda há poucas estratégias para a localização do ponto da rede em que a DT ocorre [Zhang et al. 2009, Ravaioli et al. 2015, Zhang et al. 2014]. Argumentamos que localizar DT é importante tanto para que o usuário final saiba a quem reclamar, quanto para auxiliar na aplicação das regulações de NR. Neste trabalho é proposta uma solução para detecção e localização de DT baseada em conteúdo.

A solução proposta tira proveito das políticas de roteamento da Internet para identificar em qual sistema autônomo (*Autonomous System*, AS), ou conjunto de ASes, a DT ocorreu. A ideia central da proposta é realizar medições entre diversos pares de *hosts* finais, selecionados de forma que os possíveis caminhos (em nível de AS) entre estes *hosts* sejam suficientemente diferentes. Localiza-se então a DT comparando medições em caminhos que atravessam certos ASes com medições que não atravessam estes ASes.

Os caminhos possíveis entre os pares de *hosts* finais usados nas medições são obtidos a partir da topologia de ASes e suas respectivas relações, seguindo as políticas de roteamento da Internet. Assim, não é necessário utilizar técnicas de medição para descobrir qual caminho o tráfego percorreu.

Neste trabalho, primeiramente apresentamos uma visão geral do roteamento entre ASes na Internet, com foco especial nas propriedades dos caminhos entre os ASes. Em seguida, apresentamos nossas propostas para detecção e localização de DT. Descrevemos então nossas simulações e experimentos, assim como os resultados obtidos. Realizamos três conjuntos de avaliações: um experimento no *testbed* global PlanetLab[1] para verificar se os caminhos de ASes reais seguem as propriedades de roteamento assumidas; uma simulação para verificar a acurácia da nossa proposta de detecção de DT; e uma avaliação da proposta de localização em diferentes cenários de DT.

O experimento no PlanetLab mostrou que a grande maioria dos caminhos observados seguem as regras e propriedades esperadas e que técnicas de medição de caminhos na Internet (como *traceroute*) podem não ser confiáveis. Já os resultados das simulações mostraram que nossa proposta de detecção de DT apresenta boa acurácia em diversas situações. Por fim, a localização de DT não resultou em nenhum falso-positivo ou falso-negativo nos casos em que foi possível identificar o compartamento dos ASes. Além disso, a solução foi sempre capaz de inferir corretamente se a DT ocorreu ou não nos ASes em que os *hosts* finais das medições estão.

O restante deste trabalho está organizado da seguinte forma. A Seção 2 apresenta uma visão geral das regras de roteamento entre ASes na Internet. Descrevemos nossas propostas de detecção e localização de DT na Seção 3. Em seguida, a avaliação das propostas é apresentada na Seção 4. Descrevemos os trabalhos relacionados na Seção 5 e concluímos o artigo na Seção 6.

[1] `https://www.planet-lab.org`

2. Roteamento na Internet

A Internet é formada pela interligação de redes administradas de forma independente, os sistemas autônomos (*Autonomous Systems*, ASes). Cada AS é responsável por um conjunto de prefixos IP e pode estar conectado com outros ASes. Nesta seção apresentamos uma visão geral de como o roteamento na Internet funciona no nível dos ASes.

Pacotes de dados enviados de um *host* final na Internet a outro podem atravessar diversos ASes. A sequência de ASes por onde este tráfego passa é um *caminho de ASes*, neste trabalho chamado simplesmente de "caminho". O caminho a ser percorrido por um pacote é definido pelos próprios ASes, conforme o pacote chega a cada um deles. Quando um pacote chega a um AS, este deve decidir para qual AS vizinho o pacote deve ser encaminhado em seguida. Esta decisão depende do destino final do pacote e do tipo de relação de troca de tráfego que o AS tem com cada um dos seus vizinhos.

As relações de troca de tráfego entre ASes podem ser abstraídas em três tipos [Luckie et al. 2013]: (i) *customer-to-provider* ($c2p$), ou *provider-to-customer* ($p2c$) na direção oposta; (ii) *peer-to-peer* ($p2p$); e (iii) *sibling-to-sibling* ($s2s$). Um AS conecta-se com outro AS para ganhar acesso a outras partes da Internet que não são alcançáveis a partir da sua rede ou da rede dos ASes com os quais já está conectado. Em uma relação $c2p$, um AS consumidor paga um AS provedor pelos serviços de trânsito, ou seja, pelo acesso ao resto da Internet. Em uma relação $p2p$, dois ASes trocam tráfego mutuamente sem cobranças. O tráfego trocado desta forma deve ser entre os próprios dois ASes ou entre os seus consumidores. Em uma relação $s2s$, os dois ASes pertencem a uma mesma organização e trocam tráfego livremente.

Um modelo amplamente aceito na literatura para descrever caminhos na Internet é o Gao-Rexford [Gill et al. 2013]. De acordo com este modelo, os caminhos entre dois ASes quaisquer é uma sequência de ASes em que, para cada AS provendo trânsito (um provedor de trânsito), há um AS consumidor adjacente ao provedor. Assim, há sempre um AS pagando pelo serviço de trânsito. Desta forma, um caminho na Internet deve seguir o seguinte padrão: zero ou mais *links* $c2p$, seguidos de zero ou um *link* $p2p$, seguido de zero ou mais *links* $p2c$. Este padrão configura a propriedade chamada de *valley-free* [Giotsas and Zhou 2012].

Neste trabalho chamamos de *válidos* os caminhos que seguem a propriedade *valley-free*, e de *inválidos* os caminhos que violam a propriedade. Um caminho inválido corresponde a uma sequência de ASes em que pelo menos um provedor de trânsito não está sendo pago por nenhum AS vizinho no caminho. Podem existir diversos caminhos válidos entre dois ASes.

3. Proposta de Detecção e Localização de Diferenciação de Tráfego

Nossa proposta tem o objetivo de detectar DT baseada em conteúdo, ou seja, a discriminação de uma aplicação e/ou protocolo específicos. Um ISP pode identificar a qual aplicação/protocolo um pacote de dados pertence pela porta de destino, cabeçalho e/ou *payload*. A localização de DT é feita com base nas propriedades do roteamento entre ASes na Internet.

A solução proposta recebe como entrada: um par de *hosts* finais entre os quais suspeita-se que DT esteja ocorrendo; uma lista de *hosts* finais disponíveis para se realizar

medições; e um grafo direcionado representando a topologia da Internet em nível de ASes, com as arestas marcadas com a relação entre os ASes correspondentes. A saída consiste em três listas: ASes neutros (não empregaram DT), ASes não-neutros (praticaram DT) e ASes para os quais não se pode afirmar nada (comportamento desconhecido).

A solução proposta segue 4 passos. No primeiro passo, são selecionados dois *hosts* de medição, um próximo a cada *host* inicial, de forma que os possíveis caminhos válidos entre cada par de *hosts* não atravesse os outros dois. No segundo passo, medições são realizadas entre cada par de *hosts*. Em seguida, no terceiro passo, as medições são estatisticamente analisadas, a fim de detectar se houve DT entre cada par de *hosts*. Finalmente, no quarto passo, os possíveis caminhos válidos entre cada par de *hosts* são comparados entre si e com os resultados da detecção. Por exemplo, se não foi detectada DT entre dois *hosts*, e algum AS está presente em todos os possíveis caminhos válidos entre eles, então pode-se afirmar que este AS não praticou DT.

A proposta considera três premissas: (i) a topologia de ASes e suas respectivas relações é conhecida; (ii) a propriedade *valley-free* é válida; e (iii) se um AS discrimina um tipo de tráfego, essa DT sempre ocorrerá independentemente da origem e destino do tráfego. Quanto à primeira premissa, há diversos conjuntos de dados disponíveis que inferem a topologia e relações entre ASes, baseados principalmente nas tabelas de roteamento BGP. Neste trabalho, utilizamos o grafo inferido pelo CAIDA no projeto *AS Rank* [CAIDA 2018a]. Referente à segunda premissa, a propriedade *valley-free* é uma política fundamental do roteamento BGP [Giotsas and Zhou 2012]. De acordo com a terceira premissa, se um AS discrimina uma aplicação específica, todo tráfego desta aplicação será afetado, qualquer que seja sua origem ou destino.

A seguir é descrita a estratégia de seleção de *hosts* de medição, como são realizadas as medições e as propostas de detecção e localização de DT.

3.1. Seleção de Hosts de Medição

Seja $e1$ e $e2$ o par de *hosts* finais entre os quais suspeita-se que DT esteja ocorrendo, conectados aos ASes $E1$ e $E2$, respectivamente. Dizemos que $E1$ e $E2$ são os ASes de borda. O objetivo deste passo é selecionar dois *hosts* de medição $m1$ e $m2$, conectados aos ASes $M1$ e $M2$, respectivamente. Seja $E = \{E1, E2, M1, M2\}$ o conjunto dos ASes a partir dos quais as medições serão realizadas e C o conjunto de ASes no núcleo que interconectam os ASes em E. Os *hosts* de medição $m1$ e $m2$ devem estar no conjunto de *hosts* finais disponíveis para se realizar medições. Dizemos que $M1$ e $M2$ são os ASes de medição e chamamos de A o conjunto de ASes correspondentes aos *hosts* disponíveis para medição.

$M1$ deve ser o mais próximo possível de $E1$. Dizemos que $M1$ é o AS de medição do AS de borda $E1$. Da mesma forma, $M2$ deve ser o mais próximo possível de $E2$. Além disso, há um outro critério de seleção, relacionado aos caminhos entres estes ASes. Os *hosts* de medição $m1$ e $m2$ são selecionados de forma que, no grafo de ASes, nenhum caminho válido entre quaisquer dois ASes de E atravesse algum outro AS de E, ou seja:

- Nenhum caminho válido entre $E1$ e $E2$ atravessa $M1$ ou $M2$
- Nenhum caminho válido entre $E1$ e $M2$ atravessa $M1$ ou $E2$
- Nenhum caminho válido entre $E1$ e $M1$ atravessa $E2$ ou $M2$
- Nenhum caminho válido entre $M1$ e $E2$ atravessa $E1$ ou $M2$

- Nenhum caminho válido entre $M1$ e $M2$ atravessa $E1$ ou $E2$
- Nenhum caminho válido entre $E2$ e $M2$ atravessa $E1$ ou $M1$

A busca dos ASes $M1$ e $M2$ que satisfaçam estes critérios é feita sob dois parâmetros: a diferença máxima de tamanho dos caminhos válidos examinados em relação ao menor caminho válido, σ, e a distância máxima no grafo entre os ASes de medição e os ASes de borda correspondentes, δ. Primeiramente, listamos todos os caminhos válidos entre $E1$ e $E2$ cujo tamanho seja até σ maior que o caminho mais curto entre estes ASes. É necessário examinar todos os caminhos, já que o caminho de fato percorrido pelos pacotes não é conhecido e pode mudar [Cho et al. 2017]. Além disso, é comum na Internet haver caminhos maiores do que o caminho mínimo, como observamos nos experimentos descritos na seção 4. Então, buscamos em A os dois ASes de medição ($M1$ e $M2$) que cumpram os critérios acima. Começamos esta busca pelos ASes de medição com distância 1 no grafo até os ASes de borda correspondentes (vizinhos), indo até os ASes de medição com distância δ.

A ideia central desta seleção é permitir medições que garantidamente não irão atravessar certos ASes. Nossa solução é então capaz de comparar as mesmas medições quando atravessando um certo AS com as medições quando não atravessando este AS. A propriedade *valley-free* torna possível a existência de caminhos com essas características, já que limita os possíveis caminhos entre ASes ao dividir a topologia de ASes em *Tiers*.

3.2. Medições

As medições são feitas a partir de dois tipos de tráfego, aplicação e base, enviados simultaneamente entre dois *hosts* finais durante t segundos. O fluxo de aplicação consiste em um tráfego previamente preparado, referente a uma aplicação, como Netflix ou Skype por exemplo, ou em uma porta específica. O fluxo base é similar ao fluxo de aplicação: ele pode ser o mesmo fluxo, mas criptografado [Molavi Kakhki et al. 2015], gerado sob demanda com um *payload* aleatório [Kanuparthy and Dovrolis 2010] e/ou em um porta diferente [Beverly et al. 2007]. É importante que ambos os fluxos tenham características similares, como taxa de envio e tamanho dos pacotes, para que sejam comparáveis [Weinsberg et al. 2011]. A ideia é que, se um fluxo contém mais pacotes, a chance dele ser afetado por descarte de pacotes é maior, por exemplo, mesmo que DT não esteja acontecendo, já que existem mais pacotes deste fluxo nas filas dos roteadores do que de outros fluxos.

Para cada fluxo, as seguintes medições são feitas no *host* receptor: taxa de perda de pacotes, latência, taxa de transferência e intervalos de chegada dos pacotes. Os fluxos são enviados r vezes entre cada um dos 6 diferentes pares de ASes em E, nos dois sentidos (*downstream* e *upstream*). É necessário repetir os fluxos diversas vezes para diminuir o efeito de ruído nas medições [Weinsberg et al. 2011]. Portanto, cada fluxo será enviado $12 * r$ vezes: r vezes, em 2 sentidos, entre 6 pares de ASes.

3.3. Detecção de DT

A detecção de DT a partir das medições obtidas no passo anterior é baseada no teste de hipótese Kolmogorov-Smirnov (KS) [Noether 2012]. O teste KS é um teste não-paramétrico que verifica a igualdade de duas distribuições de probabilidade. Utilizamos este teste para comparar as distribuições de diferentes conjuntos de amostras: medições

do fluxo de aplicação e medições do fluxo base. A ideia é que, em uma comunicação neutra, as medições para ambos os fluxos seguirão a mesma distribuição: ambos os fluxos foram enviados simultaneamente e de forma igual, então serão recebidos de forma igual [Weinsberg et al. 2011]. O intervalo de confiança empregado é de 95%.

Após as medições do passo anterior, temos r conjuntos de medições para cada par de ASes em E, em cada sentido. Cada conjunto de medições contém 4 métricas (taxa de perda de pacotes, latência, taxa de transferência e intervalos de chegada dos pacotes) para cada tipo de fluxo (aplicação e base). Empregamos o teste KS para comparar cada métrica de ambos os fluxos, para cada uma das r repetições. Se para pelo menos uma das 4 métricas e em pelo menos um sentido o teste KS identificar que as distribuições são diferentes em todas as r repetições, então detectamos DT entre os ASes correspondentes.

Dependendo de como um AS implementa a DT, diferentes métricas podem ser afetadas. Assim, é possível que haja diferença nas medições apenas para algumas métricas. Por exemplo, se um AS encaminha pacotes de uma aplicação específica com menor prioridade, é possível que apenas as medições relacionadas ao atraso dos pacotes sejam afetadas, enquanto as medições relacionadas à perda de pacotes não. Da mesma forma, é possível que um AS discrimine um tipo de tráfego apenas em um sentido, ou ainda que o caminho entre dois ASes efetivamente percorrido pelo tráfego em cada sentido seja diferente.

3.4. Localização de DT

Neste passo, a partir dos resultados do passo anterior e dos possíveis caminhos válidos, os ASes são classificados em 3 conjuntos: ASes neutros, ASes não-neutros e ASes de comportamento desconhecido (para os quais não se pode afirmar nada).

Primeiramente, obtemos no grafo de ASes todos os possíveis caminhos válidos (com tamanho de acordo com σ) entre cada par de ASes medido no passo anterior. Todos os ASes presentes nestes caminhos são então considerados de comportamento desconhecido. Este passo consiste em verificar o resultado da detecção entre cada par de ASes e tentar inferir, com base nos possíveis caminhos que as medições atravessaram, o comportamento de cada AS, filtrando assim o conjunto de ASes de comportamento desconhecido. A eficácia deste passo depende da seleção de bons *hosts* de medição, que irão melhor auxiliar neste processo de filtragem.

Verificamos então os pares de ASes para os quais não foi detectada DT. Para cada par destes, buscamos os ASes que estão presentes em todos os caminhos previamente computados entre eles – o que irá incluir o próprio par, já que estão nos extremos de todos os caminhos. Estes ASes garantidamente foram atravessados pelas medições, já que estão em todos os possíveis caminhos válidos. Como não foi detectada DT entre eles, todos são então considerados neutros e, portanto, removidos do conjunto de ASes de comportamento desconhecido e adicionados no conjunto de ASes neutros.

Em seguida, verificamos os pares de ASes para os quais foi detectada DT. Para cada par destes, buscamos algum AS que está presente em todos os caminhos e é o único AS ainda com comportamento desconhecido em todos os possíveis caminhos. Este AS garantidamente foi atravessado pela medição, e não há nenhum outro AS suspeito. Assim, este AS é considerado não-neutro e, portanto, removido do conjunto de ASes de comportamento desconhecido e adicionado no conjunto de ASes não-neutros.

4. Avaliação das Propostas

Nesta seção é reportado inicialmente um experimento realizado na Internet para validar as premissas acerca do roteamento na Internet. Em seguida são descritos experimentos executados com simulação para avaliar a acurácia da proposta de detecção de DT, e a proposta de localização de DT considerando diferentes cenários.

4.1. Validando Premissas Acerca dos Caminhos na Internet

Realizamos um experimento no PlanetLab com o objetivo de validar nossas premissas acerca dos caminhos na Internet, em relação ao grafo de ASes e à propriedade *valley-free*. Neste experimento, medimos os caminhos no nível dos ASes entre diversos *hosts* do PlanetLab e uma grande quantidade de prefixos da Internet. Primeiramente, obtivemos a lista de prefixos da Internet, com os respectivos ASes, disponibilizada pelo CAIDA [CAIDA 2018b], referente a maio de 2018. Também obtivemos as informações referentes às relações dos ASes em outubro de 2018 a partir do projeto *AS Rank* do CAIDA [CAIDA 2018a], utilizadas para a construção do grafo de ASes. Diversos ASes estão relacionados a mais de um prefixo. Nestes casos, um prefixo foi escolhido pra cada AS. Além disso, um pequeno número de ASes foi desconsiderado pois acabou não entrando no grafo, resultando em uma lista com 60578 prefixos.

Utilizamos 25 *hosts* do PlanetLab, a partir dos quais medimos periodicamente os caminhos para todos os prefixos da nossa lista. O experimento durou cerca de 2 semanas. As medições foram feitas utilizando a ferramenta *traceroute*. Para cada medição obtida, mapeamos os endereços IP para o respectivo AS utilizando a lista de prefixos do CAIDA. Assim, transformamos os caminhos obtidos pelo *traceroute* em caminhos consistindo de ASes. Nota-se que é comum alguns *hosts* não responderem ao *traceroute*, além de *hosts* que informam endereço IP inválido. Nestes casos, não temos como saber que o respectivo AS está no caminho, a não ser que outro *host* dentro da rede do mesmo AS envie resposta.

Em seguida, verificamos a validade de todos os caminhos medidos no grafo de ASes. Um caminho é válido se segue a propriedade *valley-free* e inválido caso contrário. Classificamos os caminhos para os quais não conseguimos obter todos os ASes (por algum problema no *traceroute*) como "desconhecidos", a não ser que, ignorando os erros de medição, o caminho seja válido: nesse caso, consideramos que pelo menos um *host* do AS respondeu corretamente. Houve também caminhos retornados pelo *traceroute* para os quais não foi possível fazer a validação pois continham arestas não presentes no grafo.

Ao final, obtivemos caminhos para 58.993 ASes distintos, totalizando 3.913.887 medições que resultaram em 1.731.636 caminhos diferentes. 905.856 caminhos distintos foram considerados desconhecidos, 8.036 inválidos e 773.449 válidos. 44.295 caminhos continham ASes vizinhos para os quais não havia aresta correspondente no grafo de ASes. Os caminhos válidos atingiram 48.572 ASes diferentes. Podemos observar a partir destes valores que, dos caminhos conhecidos, a maioria é válido (93,66%), enquanto apenas 0,97% são inválidos e 5,36% não existem no grafo. Porém, cerca de 52% dos caminhos são desconhecidos, o que decorre da limitação de executar medições com *traceroute*.

Com o objetivo de verificar se é comum caminhos reais na Internet serem maiores que os caminhos mais curtos possíveis – tendo em vista o parâmetro σ de nossas propostas – comparamos o tamanho dos caminhos válidos medidos com os caminhos válidos mais curtos encontrados no grafo de ASes. Em nosso experimento, cerca de 56% das medições

atravessaram caminhos com tamanho mínimo, enquanto cerca de 32% atravessaram caminhos com tamanho 1 *hop* maior do que o tamanho mínimo no grafo e cerca de 10% foram caminhos com tamanhos maiores em 2 *hops* que o mínimo.

4.2. Simulações da Proposta de Detecção de DT

Realizamos diversas simulações com o objetivo de verificar se a proposta é capaz de detectar DT entre dois *hosts* finais, em diferentes situações. Avaliamos também os parâmetros t e r (duração e repetições de cada fluxo, respectivamente). As simulações foram realizadas no *framework* de simulação OMNeT++[2]. Definimos diversos cenários de DT e variamos, em cada cenário, o volume de tráfego de fundo. Repetimos as simulações 10 vezes ($r = 10$) para cada valor de tráfego de fundo e para cada cenário de DT. Em cada simulação, dois fluxos com duração de 30 segundos ($t = 30s$) foram enviados entre *hosts* finais. Um fluxo corresponde a um tráfego de baixa prioridade, que sofrerá degradação, e o outro fluxo corresponde a um fluxo de alta prioridade, que não sofre nenhuma alteração na rede. Empregamos então nossa solução para detectar diferenciação entre estes dois fluxos. Avaliamos também valores menores para t e r a partir de subconjuntos dos dados obtidos.

A topologia das simulações é mostrada na Figura 1. Quatro tipos de tráfego são gerados por quatro fontes diferentes: (i) tráfego de alta prioridade a partir de um host final e (ii) tráfego de baixa prioridade, gerados por outro host final; além de (iii) tráfego de fundo de alta prioridade, e (iv) tráfego de fundo de baixa prioridade, ambos gerados por diversos *hosts* finais, conforme parâmetros da simulação. Todas as fontes de tráfego estão conectadas a um módulo do simulador que representa a Internet, responsável por tratar os tipos de tráfego de formas diferentes, conforme os cenários de DT definidos. Este módulo conecta-se com um roteador que encaminha o tráfego para os respectivos destinos. Para cada tipo de tráfego há um *host* receptor, nos quais as medições são feitas. Todos os *links* – das fontes para o módulo de DT, então para o roteador e finalmente até os receptores – têm largura de banda 100Mbps e atraso de propagação de 10ms.

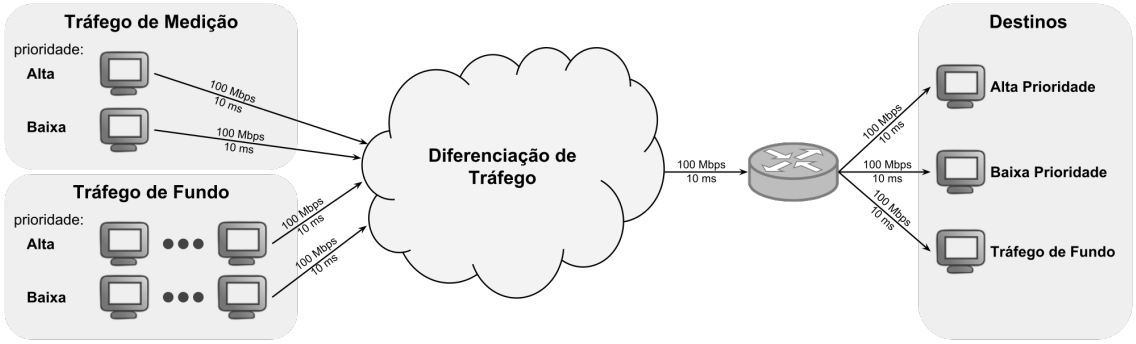

Figura 1. Topologia das simulações.

Os fluxos de alta e baixa prioridade são gerados simultaneamente e da mesma forma. Ambos consistem em um *stream* contínuo de pacotes UDP. Este protocolo foi utilizado pois permite um controle mais fino de como o tráfego é gerado em relação ao protocolo TCP. O tamanho dos pacotes varia aleatoriamente de 450 a 550 bytes. O

[2] `https://omnetpp.org`

intervalo de envio varia aleatoriamente de 343,35μs a 419,65μs, resultando em uma taxa de envio média de 10Mbps. O tráfego de fundo é gerado da mesma forma, porém o tamanho dos pacotes varia de 400 a 600 bytes. Em nossas simulações, a quantidade de fluxos de tráfego de fundo de cada prioridade variou de 0 a 5. Assim, os volumes totais de tráfego de fundo, juntando alta e baixa prioridade, foram de 0, 20, 40, 60, 80 e 100Mbps. Os valores aleatórios utilizados seguem distribuição uniforme.

Os diversos cenários de DT foram implementados conforme três parâmetros: atraso, vazão e probabilidade de descarte. Assim, a DT é feita empregando diferentes valores para estes parâmetros para cada prioridade de tráfego. Definimos 19 cenários de DT. Um deste cenários é o *Neutro*, no qual o módulo de DT emprega os mesmos valores para ambas as prioridades: de 90 a 100ms de atraso, vazão de 90 a 100Mbps e probabilidade de descarte de 1%. Nos demais cenários, valores que afetam negativamente os parâmetros são empregados para os fluxos de baixa prioridade. Os valores do cenário *Neutro* são sempre empregados para o tráfego prioritário. A tabela 1 mostra os valores para todos os cenários de DT. Na tabela, cenários denominados *DelayX* indicam atrasos maiores em cerca de X% como mostrados na coluna correspondente. Cenários *RateX* indicam vazão menor em cerca de X%. *DropXY* correspondem a uma probabilidade de descarte de X,Y%. Alguns cenários combinam aumento do atraso e diminuição da vazão (*DelayRateX*).

Tabela 1. Cenários de DT das simulações.

Cenário de DT	Atraso (ms)	Vazão (Mbps)	Descarte (%)
Neutro	90-100	90-100	1
Delay5	**90-105**	90-100	1
Rate5	90-100	**85-100**	1
DelayRate5	**90-105**	**85-100**	1
Delay10	**90-110**	90-100	1
Rate10	90-100	**80-100**	1
DelayRate10	**90-110**	**80-100**	1
Delay15	**90-115**	90-100	1
Rate15	90-100	**75-100**	1
DelayRate15	**90-115**	**75-100**	1
Delay20	**90-120**	90-100	1
Rate20	90-100	**70-100**	1
DelayRate20	**90-120**	**70-100**	1
Drop11	90-100	90-100	**1,1**
Drop12	90-100	90-100	**1,2**
Drop13	90-100	90-100	**1,3**
Drop14	90-100	90-100	**1,4**
Drop15	90-100	90-100	**1,5**
Drop20	90-100	90-100	**2**

Em cada simulação, comparamos as distribuições de cada uma das 4 métricas (latência, intervalos de chegada dos pacotes, taxa de transferência e taxa de perda de pacotes) para cada um dos dois fluxos (baixa e alta prioridade), conforme a proposta de detecção. Os valores avaliados para o parâmetro t foram 5, 10, 15, 20, 25 e 30s, para o parâmetro r foram de 1 a 10, e os níveis de tráfego de fundo foram de 0 a 5. Por motivo de espaço, mostraremos a seguir apenas os resultados mais relevantes. Todos os resultados podem ser visualizados em `http://www.inf.ufpr.br/tgarrett/`

`sbrc19-graficos`.

Primeiramente, observamos em quantas das 10 repetições, *individualmente* foi detectada diferença entre os dois fluxos para cada métrica, que chamamos de taxa de detecção. A Figura 2 mostra a taxa de detecção para o cenário *Neutro*. Neste cenário ambos os tráfegos têm o mesmo tratamento não discriminatório na rede. Portanto uma detecção de DT neste caso corresponde a um falso-positivo. Entretanto, conforme descrito anteriormente, a detecção de DT é feita não usando valores individuais (mostrados nos gráficos da Figura 2) mas o conjunto de diversas repetições e a combinação de todas as métricas (avaliado mais à frente nesta seção). Cada um dos gráficos da Figura 2 corresponde a uma métrica, em cada um são plotados resultados para cada valor de t. A taxa de detecção está no eixo vertical, e diferentes níveis de tráfego de fundo estão no eixo horizontal. No gráfico mais à esquerda (Latência), observa-se valores maiores para a taxa de detecção; nos demais as taxas são menores, sendo quase zero no gráfico da taxa de transferência (terceiro da esquerda para a direita).

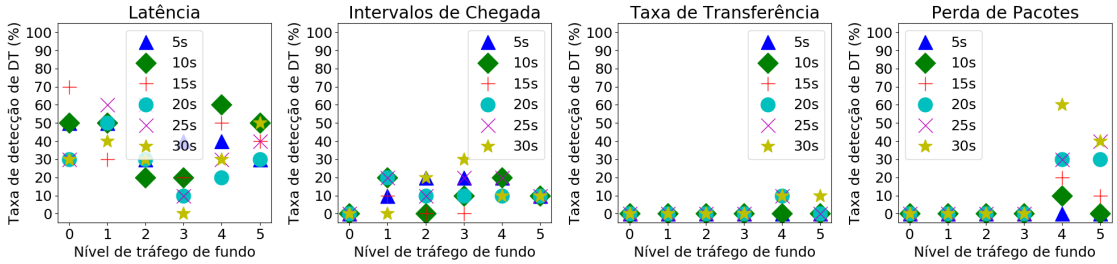

Figura 2. Taxa de detecção no cenário *Neutro*.

Mostramos agora resultados para cenários com discriminação. Para os cenários DelayX e DelayRateX, com aumento do atraso dos pacotes do tráfego de baixa prioridade, a taxa de detecção tanto para latência como para intervalos de chegada foi de 100% para todos os valores de t e tráfego de fundo. Já para as outras duas métricas, a taxa de detecção foi alta apenas para $t \geq 15s$. Nos cenários RateX, as taxas de detecção foram mais altas para intervalos de chegada quando $t \geq 20s$. Nos cenários que alteram perda de pacotes, apenas o cenário *Drop20* teve valores consistentemente altos para a taxa de detecção quando $t \geq 20s$. Mostramos alguns desses resultados a seguir.

A Figura 3 mostra a taxa de detecção para o cenário *Rate5*. Observa-se que a taxa de detecção foi maior para intervalos de chegada e $t \geq 15s$. A Figura 4 mostra a taxa de detecção para o cenário *Drop20*. Observa-se que, para perda de pacotes e taxa de transferência, a taxa de detecção foi mais alta quando $t \geq 20s$.

Avaliamos também nosso critério de detecção combinando as 4 métricas para diferentes repetições r. A Figura 5 mostra se a detecção foi ou correta ou incorreta no cenário *Neutro*. Cada gráfico corresponde a um valor diferente de t (5s, 10s e 30s). Nestes gráficos o eixo vertical corresponde ao tráfego de fundo e o eixo horizontal ao número de repetições r. Destaca-se que, mesmo com a alta taxa de detecção observada para a latência neste cenário (Figura 2), houve poucos falsos-positivos, todos para poucas repetições $r \leq 2$.

Nos cenários que alteram o atraso do pacotes, a detecção foi correta para todos os valores de r, t e tráfego de fundo. Nos cenários que alteram apenas a vazão, a detecção

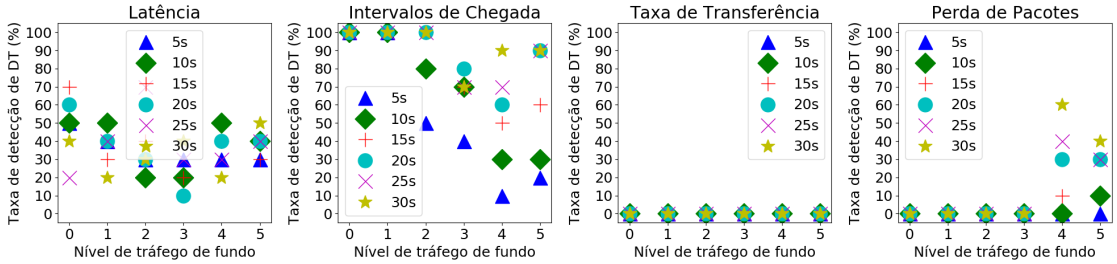

Figura 3. Taxa de detecção no cenário *Rate5*.

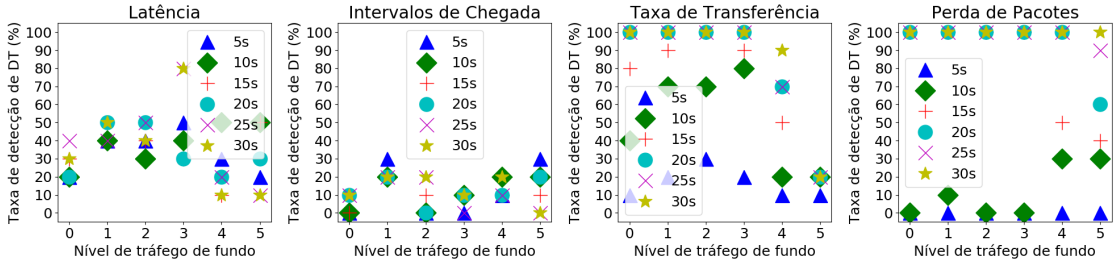

Figura 4. Taxa de detecção no cenário *Drop20*.

foi sempre correta para $t \geq 20s$, com exceção do cenário *Rate5*, que mostramos a seguir. Nos cenários que alteram perda de pacotes, a detecção teve resultados mais corretos no cenário *Drop20*, para $t \geq 20s$, também mostrado a seguir.

A Figura 6 mostra os resultados de detecção para o cenário *Rate5*. Observa-se que a detecção foi correta para $t \geq 15s$ e para nível de tráfego de fundo até 2, para todos os valores de r. A Figura 7 mostra os resultados de detecção para o cenário *Drop20*. Observa-se que a detecção foi sempre correta para $t \geq 20s$ e tráfego de fundo até 4. Para $t = 30s$, a detecção foi correta em todos os níveis de tráfego de fundo.

4.3. Cenários de Localização de DT

Realizamos uma avaliação das capacidades e limitações da proposta de localização de DT em diferentes cenários. O grafo de ASes utilizado neste estudo foi o mesmo utilizado anteriormente no experimento descrito na Subseção 4.1.

Primeiramente, escolhemos os dois ASes de borda que, dentre os ASes correspondentes aos *hosts* utilizados no experimento do PlanetLab, têm os maiores graus no grafo de ASes. Então, listamos todas as possíveis combinações de ASes de medição que satisfazem os critérios descritos na Seção 3. Utilizamos $\delta = 1$ e $\sigma = 0$, ou seja, os ASes de medição são vizinhos dos respectivos ASes de borda, e apenas os caminhos válidos mais curtos são examinados. 37.770 combinações de ASes de medição ($M1$ e $M2$) para os dois ASes de borda escolhidos ($E1$ e $E2$) foram geradas.

Para cada uma das 37.770 combinações de ASes de medição, obtivemos todos os caminhos entre os pares de ASes de borda/medição. Então, listamos todos os ASes diferentes presentes nestes caminhos. Geramos então uma série de cenários diferentes. Em cada cenário, um AS é o responsável pela DT, e um caminho para cada par de ASes de borda/medição é selecionado como o que efetivamente seria percorrido pelo tráfego.

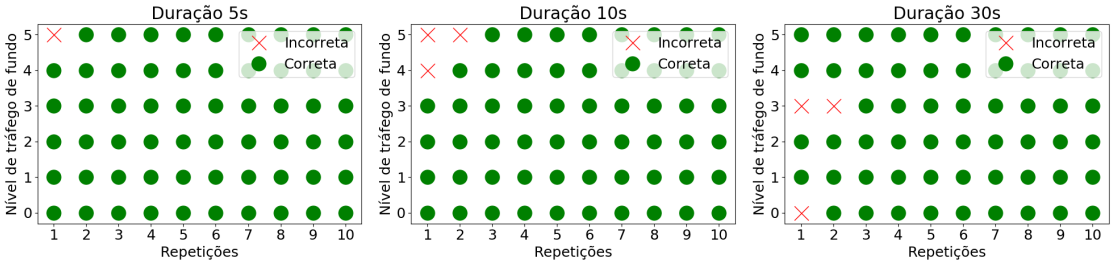

Figura 5. Resultados da detecção de DT para o cenário *Neutro*.

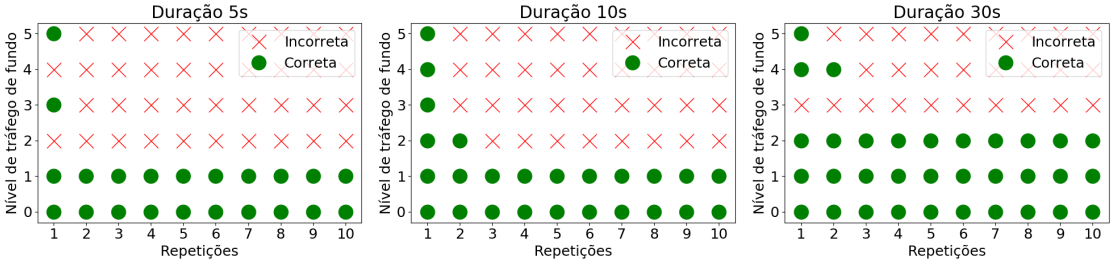

Figura 6. Resultados da detecção de DT para o cenário *Rate5*.

35.605.706 cenários foram gerados desta forma.

Para avaliar a localização em cada cenário gerado, simulamos as medições correspondentes. Para um dado cenário, na medição entre cada par de ASes de borda/medição detecta-se DT se o AS responsável pela DT naquele cenário está no caminho selecionado para o respectivo par. Assim, é possível que nenhuma medição acuse DT, já que o AS responsável por DT pode não estar presente em nenhum dos caminhos selecionados para um cenário.

Verificamos então quantos ASes, em média, foram corretamente e incorretamente inferidos como neutros, não-neutros e de comportamento desconhecido, em todos os cenários gerados. Considerando todos os cenários, cerca de 28% dos ASes neutros foram identificados corretamente, em média. Todos os demais 72% dos ASes estão no núcleo. Em cerca de 22% dos cenários o AS não-neutro foi identificado corretamente. Em nenhum cenário um AS foi classificado incorretamente. A principal conclusão é que em *todos* os cenários os ASes de borda e de medição tiveram seu comportamento corretamente identificado (neutro ou não-neutro). Portanto, nossa proposta é capaz de inferir corretamente se os ASes de borda/medição são os responsáveis ou não pela DT. Porém, nos casos dos ASes no núcleo, nada se pode afirmar na maioria dos casos: mais medições são necessárias.

5. Trabalhos Relacionados

Diversas soluções para detectar DT na Internet foram propostas na última década [Tariq et al. 2008, Lu et al. 2010, Kanuparthy and Dovrolis 2010, Dischinger et al. 2010, Weinsberg et al. 2011, Molavi Kakhki et al. 2015]. Um *survey* sobre detecção de DT pode ser encontrado em [Garrett et al. 2018]. Estas soluções são baseadas em medições de rede e inferência estatística. Em geral, elas fazem medições a partir de um ou mais

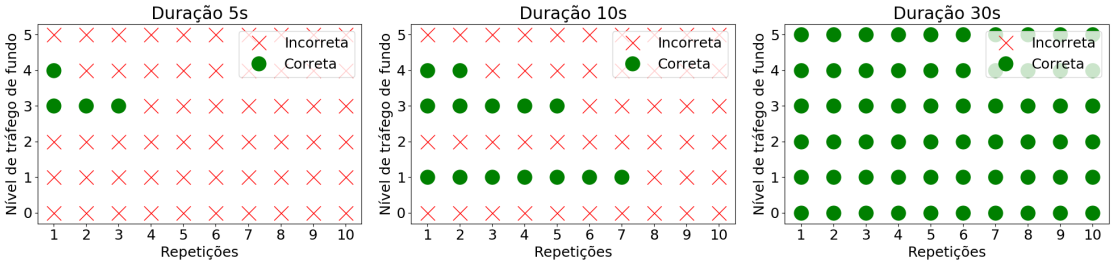

Figura 7. Resultados da detecção de DT para o cenário *Drop20*.

hosts, empregando diferentes tipos de tráfego. As medições são então comparadas para verificar se houve alguma diferença significativa entre diferentes conjuntos de amostras.

São poucas as propostas para localizar DT na Internet. Em [Zhang et al. 2009] e [Ravaioli et al. 2015], os autores utilizam medições baseadas em TTL (como o *traceroute*) para localizar DT. Este tipo de medição, como mostramos na Seção 4, não é universalmente suportada na Internet e não produz resultados confiáveis. Diversos ASes limitam a quantidade de respostas ICMP que enviam, por exemplo. Nossa proposta busca localizar DT baseada nos possíveis caminhos entre os *hosts*, sem efetuar medições explícitas do caminho. Já em [Zhang et al. 2014], os autores assumem o conhecimento completo da topologia da rede, em nível de *hosts*, para identificar em que ponto a DT ocorre. Esta informação é difícil de ser obtida corretamente, prejudicando a eficácia da solução.

6. Conclusão

Apesar de existirem diversas soluções para a detecção de DT na Internet, ainda há uma lacuna em termos de uma solução eficaz para localizar em que ponto da rede a DT está ocorrendo. Neste trabalho apresentamos uma solução que não apenas detecta DT, mas também localiza qual AS, ou conjunto de ASes, empregou DT. A solução proposta tira proveito de propriedades do roteamento na Internet, em particular a propriedade *valley-free* que permite que ASes de medição sejam selecionados de forma que os caminhos entre os *hosts* de medição sejam suficientemente diferentes, possibilitando a identificação de ASes neutros e não-neutros.

Um experimento executado no PlanetLab mostrou que a propriedade *valley-free* é válida na maioria dos caminhos medidos corretamente. Também concluímos que o uso de *traceroute* gera resultados limitados, motivando a nossa abordagem. Para avaliar nossa proposta de detecção de DT, realizamos uma série de simulações que mostraram que a estratégia proposta de combinar diversas métricas foi capaz de detectar corretamente a DT em diversas situações. Realizamos também uma avaliação em diversos cenários de DT para verificar as capacidades e limitações da proposta de localização de DT. Essa avaliação mostrou que nossa solução é sempre capaz de inferir corretamente o comportamento dos ASes de borda e de medição, mas, como esperado, não é capaz de inferir o comportamento dos ASes no núcleo na maioria dos casos.

Trabalhos futuros incluem a investigação de estratégias para inferir o comportamento dos ASes no núcleo. Realizar uma avaliação da proposta na Internet também poderá contribuir para ajustar melhor os parâmetros. Além disso, detectar e localizar DT baseada na origem/destino dos pacotes também é um tópico relevante no contexto de NR.

Referências

Beverly, R., Bauer, S., and Berger, A. (2007). *The Internet Is Not a Big Truck: Toward Quantifying Network Neutrality*, pages 135–144. Springer.

CAIDA (2018a). CAIDA AS Rank. `http://as-rank.caida.org/`. Acessado em 20 de Novembro de 2018.

CAIDA (2018b). Routeviews Prefix to AS mappings Dataset for IPv4 and IPv6. `http://www.caida.org/data/routing/routeviews-prefix2as.xml`. Acessado em 20 de Novembro de 2018.

Cho, S., Nithyanand, R., Razaghpanah, A., and Gill, P. (2017). A Churn for the Better: Localizing Censorship Using Network-level Path Churn and Network Tomography. In *International Conference on Emerging Networking EXperiments and Technologies*, CoNEXT '17, pages 81–87. ACM.

Dischinger, M., Marcon, M., Guha, S., Gummadi, K. P., Mahajan, R., and Saroiu, S. (2010). Glasnost: Enabling End Users to Detect Traffic Differentiation. In *USENIX Conference on Networked Systems Design and Implementation*, pages 27–27. USENIX Association.

Garrett, T., Dustdar, S., Bona, L. C. E., and Duarte Jr., E. P. (2017). Ensuring Network Neutrality for Future Distributed Systems. In *International Conference on Distributed Computing Systems (ICDCS)*, pages 1780–1786.

Garrett, T., Setenareski, L. E., Peres, L. M., Bona, L. C. E., and Duarte, E. P. (2018). Monitoring Network Neutrality: A Survey on Traffic Differentiation Detection. *IEEE Communications Surveys Tutorials*, 20(3):2486–2517.

Gill, P., Schapira, M., and Goldberg, S. (2013). A Survey of Interdomain Routing Policies. *SIGCOMM Comput. Commun. Rev.*, 44(1):28–34.

Giotsas, V. and Zhou, S. (2012). Valley-free violation in Internet routing – Analysis based on BGP Community data. In *IEEE International Conference on Communications (ICC)*, pages 1193–1197.

Kanuparthy, P. and Dovrolis, C. (2010). DiffProbe: Detecting ISP Service Discrimination. In *IEEE INFOCOM*, pages 1–9.

Lu, G., Chen, Y., Birrer, S., Bustamante, F. E., and Li, X. (2010). POPI: A User-Level Tool for Inferring Router Packet Forwarding Priority. *IEEE/ACM Transactions on Networking*, 18(1):1–14.

Luckie, M., Huffaker, B., Dhamdhere, A., Giotsas, V., and claffy, k. (2013). AS Relationships, Customer Cones, and Validation. In *Internet Measurement Conference*, IMC '13, pages 243–256. ACM.

Molavi Kakhki, A., Razaghpanah, A., Li, A., Koo, H., Golani, R., Choffnes, D., Gill, P., and Mislove, A. (2015). Identifying Traffic Differentiation in Mobile Networks. In *Internet Measurement Conference*, pages 239–251. ACM.

Noether, G. E. (2012). *Introduction to statistics: the nonparametric way*. Springer.

Ravaioli, R., Urvoy-Keller, G., and Barakat, C. (2015). Towards a General Solution for Detecting Traffic Differentiation at the Internet Access. In *International Teletraffic Congress (ITC)*, pages 1–9.

Tariq, M. B., Motiwala, M., and Feamster, N. (2008). NANO: Network Access Neutrality Observatory. In *7th ACM Workshop on Hot Topics in Networks (Hotnets-VII)*.

Weinsberg, U., Soule, A., and Massoulie, L. (2011). Inferring traffic shaping and policy parameters using end host measurements. In *IEEE INFOCOM*, pages 151–155.

Zhang, Y., Mao, Z. M., and Zhang, M. (2009). Detecting Traffic Differentiation in Backbone ISPs with NetPolice. In *Internet Measurement Conference*, pages 103–115. ACM.

Zhang, Z., Mara, O., and Argyraki, K. (2014). Network Neutrality Inference. *SIGCOMM Computer Communication Review*, 44(4):63–74.