Monitoring Network Neutrality: A Survey on Traffic Differentiation Detection

Thiago Garrett, Ligia E. Setenareski, Leticia M. Peres, Luis C. E. Bona,
Elias P. Duarte Jr., *Senior Member, IEEE*

***Abstract*—Network Neutrality is becoming increasingly important as the global debate intensifies and governments worldwide implement and withdraw regulations. According to this principle, all Internet traffic must be processed without differentiation, regardless of origin, destination and/or content. Neutrality supporters claim that traffic differentiation can compromise innovation, fair competition and freedom of choice. However, detecting that an ISP is not employing traffic differentiation practices is still a challenge. This work presents a survey of strategies and tools for detecting traffic differentiation on the Internet. After presenting basic neutrality definitions as well as the worldwide debate, we describe ways that can be used by an ISP to implement traffic differentiation, and define the problem of differentiation detection. This is followed by a description of multiple existing strategies and tools. These solutions differ mainly on how they execute network measurements, the metrics employed, as well as traffic generation techniques, and statistical methods. We also present a taxonomy for the different types of traffic differentiation and the different types of detection. Finally, we identify open challenges and future research directions.**

***Index Terms*—Network Neutrality, Traffic Differentiation, Statistical Inference, Network Measurement**

I. Introduction

THE Internet has become the worldwide interconnection of billions of individuals through providers operated by government, industry, academia, and private parties. Supporting the Internet fast growth is a challenge [1], not only due to technical issues, but also to economical factors. For instance, in order to decrease and/or postpone investments in the network infrastructure, Internet Service Providers (ISPs) may employ discriminatory traffic management techniques [2]. Their motivations can be manifold: ISPs may seek to obtain competitive advantages or to increase the number of customers or charge higher fees, both from users and content/service providers.

Discriminatory traffic management practices are applied to prioritize or degrade specific types of traffic over others [3] – based on content, protocol, origin or destination, for example. These practices are often called Traffic Differentiation (TD). Note that traffic differentiation can be applied for a myriad of reasons. For example, TD can be used to control congestion by throttling bandwidth-hungry applications, such as P2P file sharing and video streaming [4], [5]. TD can also be adopted because of commercial agreements, by which content/service providers pay extra fees to get their traffic prioritized, the so-called fast-lanes [6]. Other reasons include obtaining competitive advantage by which an ISP prioritizes the traffic of its own services or degrading (or even blocking) traffic from competitors [7], [8].

TD is part of the long and controversial debate regarding Network Neutrality (NN) [9]. A large number of countries worldwide have enforced NN with regulations [6]. Examples include Japan [10], Norway [11], Canada [12], Chile [13], Colombia [14], [15], South Korea [16], Brazil [17], [18], Mexico [19], USA [20], India [21], and the European Union [22]. A definition of NN common to these several regulations states that, in a neutral network, every type of traffic must be treated equally, regardless of its origin, destination and/or content, i.e. TD is not allowed [23].

One of the central topics in the global NN debate is how to ensure that the Internet continues to be an environment that fosters innovation for all interested parties [24]. On the one hand, TD might threaten three concepts that were essential to the Internet current success [25]: innovation, fair competition, and consumer's freedom of choice. For instance, TD would allow ISPs to control which services would have a better chance of succeeding, by prioritizing their traffic [2]. In such scenario, new services and innovative solutions might struggle, since they would not be able to fairly compete against the already well-established services [26], [27]. On the other hand, less restrictions to the ISPs might result in a more competitive market [9]. There are even those that argue that the consumers should be able to decide which portion of their traffic is to be prioritized [28].

Regardless of regulations and the outcome of this global debate, we argue that TD practices should be transparent, since they can can significantly affect end-users and content/service providers. Regulations alone cannot guarantee ISP compliance, and even in a non-regulated environment, transparency should be a basic requirement of ISP users as well as service/content providers. Note there exist TD practices that are not covered by regulations [29], [30]. It is thus essential to monitor the presence of TD on the Internet [31]. The purpose is not only to increase transparency but also to check whether ISPs are complying with regulations.

However, detecting TD is not a trivial task [32]. ISPs may implement TD in a myriad of different ways. Traffic may be discriminated based on protocol, origin, destination, and payload, among others example [33]. Furthermore, several techniques may be employed, such as traffic shaping [34], traffic policing [35] and even discriminatory internal routing [36]. Another challenge is to discover where within the ISP

All authors are with the Department of Informatics, Federal University of Parana. P.O. Box 19018, Curitiba, PR 81531-980, Brazil.
E-mails: {tgarrett,ligia,lmperes,bona,elias}@inf.ufpr.br

TD is taking place; this can be truly hard as there can no prior knowledge of the internal structure of the network [37]. To make things worse, there are several other factors besides TD that can affect traffic performance and be misinterpreted as TD [32]. Examples include congestion, cross-traffic and load balancing.

A large number of strategies and tools have been proposed to detect TD [3], [32], [33], [36]–[42]. These solutions are based on network measurements and statistical inference. Since there is no way determine the properties of an arbitrary network in a state that cannot be precisely described, existing solutions rely on end-to-end measurements to infer possible discriminatory behaviors. In general, they take measurements from one or several end-hosts, employing different types of traffic and probes. The measurements obtained are then analyzed to determine whether there was a significant difference over different sets of samples. Robust statistical models are necessary to distinguish between TD and performance variations caused by other phenomena.

The existing solutions for TD detection are based on different assumptions, leading to different capabilities and limitations. Different solutions may detect different types of TD, using different techniques. For instance, several of these solutions generate synthetic traffic between two end-hosts that allow the comparison of the end-to-end performance of of different applications. Some assume the existence of neutral traffic, which establish a baseline that allows detection based on comparison results. Other solutions take measurements for individual hops along the route between two end-hosts, in an attempt to identify exactly where TD occurred. There are also solutions that passively capture the traffic from different applications, instead of generating traffic or issuing probes.

In this survey, we first describe the worldwide debate around NN. Then we set a common ground for understanding the problem of detecting TD given the existing solutions. We give an overview of TD in the Internet, as well as definitions of neutrality and the problem of detecting TD. Next, we describe several existing solutions for the problem. We then define a taxonomy for the different types of TD and the different types of detection. We also identify the most common techniques employed by existing solutions, and compare the solutions according to our taxonomy and common features. Finally, we identify open challenges and future research directions. To the best of our knowledge, this is the first survey on the problem of detecting TD in the context of NN. We hope the survey will allow for a better understanding of the concepts and techniques related to TD and its detection, helping future research efforts in this increasingly important problem.

The rest of the survey is organized as follows. In section II, we present an overview of the worldwide NN debate. In section III, basic concepts that serve as a basis for the rest of the survey are presented. Existing solutions for detecting TD are then described in section IV. We then present a taxonomy of the different types of TD and TD detection in section V. Then, in subsection VI, we consolidate the state of the art by identifying the most common techniques employed by current solutions, and the main challenges for detecting TD. We also compare the solutions according to the defined taxonomy. In section VII we and identify open challenges and future work. The conclusion follows in section VIII.

II. Network Neutrality: The World Debate

This section presents an overview of the NN debate that has been going on worldwide for the past 15 years. First, we present some of the major issues that have defined the debate, and next list several real NN violation cases that have happened in multiple countries around the world and serve as a strong motivation for this paper.

A. Introduction to the NN Debate

The worldwide NN debate started in 2002, when the *Federal Communications Commission* (FCC), the regulator of telecommunications in the USA, changed the classification of the broadband service in the country. The service was previously classified as a common telecommunication service, like for example fixed-line telephones. The classification was changed to *information service* [43], dissociating broadband services from the laws regulating telecommunications. With this new classification, ISPs obtained the power to prioritize or block certain types of traffic over others. The previous classification (as a telecommunication service) had ensured a neutral Internet. The new situation started the debate over *Network Neutrality*, term coined by Tim Wu [44] in 2002.

In 2003, Tim Wu and Lawrence Lessig, one of the creators of Creative Commons [45], sent a letter to the FCC with a proposal for a neutral Internet [46]. This proposal established a trade-off between the freedom of end-users regarding their Internet connections, and the freedom of ISPs to determine the management policies they adopted in their own networks.

Since then, an increasing number of individuals, companies, as well as private and public institutions have joined the NN debate. Several content providers, such as Google and Netflix, advocate for NN, while the opposition is mostly comprised of ISPs. The scientific community have also joined the debate, proposing solutions to detect NN violations that have helped the discussions, as well as the definition and enforcement of regulations that have been established around the world.

These regulations consist of rules, principles and/or laws for ensuring a neutral Internet. The way in which the regulations were defined around the world varied greatly. Examples of countries that already have some kind of NN regulation include (listed in chronological order): Japan [10], Norway [11], Canada [12], Chile [13], Colombia [14], [15], South Korea [16], Brazil [17], [18], Mexico [19], USA [20], India [21], and the European Union [22].

B. Real Cases of NN Violations

NN violation cases have been reported increasingly frequently as the number of Internet users worldwide grows and more governments implement NN regulations. We present below, in chronological order, several real cases of NN violations from around the world. These cases were not only reported in scientific papers and dissertations, but also by the press and by users in the Internet itself.

In July 21st, 2005, members of the *Telecommunications Workers Union* (TWU), initiated a strike against Telus, a Canadian ISP. In the following day, Telus blocked its consumers from accessing the webpage *Voices for Change*, created and maintained by TWU members. Telus claimed that the terms of service established with consumers allowed the ISP to block any webpage [47]. In July 28th, Telus ceased the block due to a preliminary injunction.

In July 24th, 2007, the *Web Tripwires* tool was released [48]. The tool detects content modification in webpages. The data obtained in the first 20 days of operation showed, among other results, that ISPs performed intentional modification on the traffic from 46 of a total of 50171 hosts that were monitored.

Also in 2007, in a discussion forum from the *DSLReports* webpage [49], Topolski reports that the Comcast ISP employed equipments from Sandvine [50] in order to manipulate communication sessions generated by P2P applications. According to Topolski, the Sandvine device processed all packets ingressing the ISP network, and interrupted any P2P traffic exceeding a certain threshold rate determined by Comcast. Topolski also claimed that these interruptions were performed by injecting forged reset (RST) packets, from the TCP protocol to disrupt application flows.

In April 2009, Kendrick reported in the *Gigaom* webpage [7] that the German ISP T-Mobile was blocking traffic from Skype in all its networks, and this was confirmed by the ISP. T-Mobile claimed that the reasons for blocking all Skype VoIP traffic in its networks were only technical and not economical. According to the ISP, the high traffic from the application would hinder the performance of the network, which would result in consumers blaming the ISP in case the application was not working properly.

In July 2009, *British Telecommunications* (BT), the regulator of telecommunications in the United Kingdom, was accused of throttling, i.e. limiting the maximum rate for video streaming from the BBC TV channel [51]. BT claimed that all of its traffic management practices aimed at optimizing the experience for all consumers.

In 2010, the authors of [5] presented arguments both in favor, and against NN, based on the case of P2P blocking performed by the Comcast ISP. The authors claimed that P2P services do not affect the quality of Internet services, they only transfer the need for investments from content providers to ISPs. According to the authors, the only harm caused by P2P applications in a neutral network would be that ISPs would not be able to charge extra fees from content providers in order to transport their traffic.

In February 2011, a non-profit organization named *GreatFire* [52] was created. This organization monitors the status of censored webpages and keywords in China. The censorship would be deployed by the so-called *Great Firewall of China*. The webpage of the organization helps Chinese Internet users to access some blocked content, to test their connections, and also publishes data regarding the monitored webpages and keywords. For instance, from the 68066 domains monitored, 6651 are blocked in China as of August 2017.

In April 2011, the authors of the *CensMon* [53] tool, which detects censorship on the Internet, conducted an experiment using the PlanetLab testbed. The experiment employed 174 hosts located in 33 different countries, and lasted 14 days. During this period 4950 web addresses, from 2500 domains, were tested by the tool. The results show that 951 addresses from 193 domains were filtered. Most of the filtered domains, 176, were detected by the host in China.

The European webpage *Respect My Net* [54] was launched in September 22nd, 2011. The webpage allows Internet users to report NN violations. The site maintains a list of all reported cases, along with confirmations and proofs given by users. Cases not considered as NN violations, according to the webpage policy, are deleted. As of August 2017, the webpage has a total of 102 confirmed reports, involving 21 European countries, and 56 ISPs. Among the reported cases, at least three had a significant impact: (i) YouTube traffic throttling in France by the Free ISP, confirmed by 435 users; (ii) DNS blocking for the *thepiratebay.org* webpage in Belgium, by the Mobile Vikings ISP, confirmed by 18 users; and (iii) blocking of TCP port 25 for all SMTP services by the Belgian ISP Belgacom, except for its own service, confirmed by 21 users.

A study regarding two cases of NN violations, in the USA and Canada, was published in 2012 [4]. In these two cases, ISPs employed *Deep Packet Inspection* (DPI) techniques to identify P2P applications, in order to block or throttle their traffic. This discriminatory practice resulted in protests, legal processes, among other reactions. The study describes the impact of DPI on political and economical aspects of the Internet, such as innovation, competition, and transparency. The authors report that their study was based on data obtained by the *Glasnost* [38] tool, which is described in this survey in Section IV.

The authors of the *Adkintun* [55] tool, described in Section IV of this survey, report in [56] three cases studies of the tool in Chile, between 2011 and 2013. In one of these cases, Adkintun was employed by request of the Chilean telecommunications regulator (SUBTEL), to evaluate the behavior of two Chilean ISPs, VTR and Movistar, which jointly control about 80% of the broadband services in the country. Results show that the *download* bandwidth during the night for both ISPs was significantly lower than the bandwidth specified by their contracts. In a second case, the state-owned TV channel reported that the number of complaints from Internet users significantly increased after the launch of the Adkintun tool, as well as the quality of the service provided by ISPs. The third case presented by the authors consists of a legal process against SUBTEL, accusing the regulator itself of not taking action against ISPs which were not fully complying with the Chilean NN law. This process was based on data collected and published by the Adkintun tool, which was maintained by SUBTEL. According to the authors, this was the first case in which the infrastructure of a public institution, whose main purpose was to ensure NN, was used against itself. The authors also report that Adkintun has been collecting data since September 2011, and was already used by more than 10000 users.

The HAKOMetar [57] tool, described in section IV, was employed by end-users in Croatia between November 2012 and March 2013. During this period, more than 25000 mea-

surements were made on end-user Internet connections. Results show dozens of cases in which the bandwidth effectively delivered to the users was significantly lower than the contracted bandwidth. The authors report that these measurements motivated complaints from users against 3 of the 16 ISPs measured. The data obtained by HAKOMetar was attached to these complaints, which had results favorable to the users.

In 2013, Anderson presented a study reporting BitTorrent traffic throttling in Iran [58]. Data collected by users that employed the *Network Diagnostic Tool* (NDT), hosted on the M-Lab measurement platform [59], were analyzed. Results showed two long periods in which BitTorrent traffic was throttled. Between November 30th, 2011 and August 15th, 2012, the throughput was decreased in average by 77%. Between October 4th and November 22nd, 2012, the throughput was decreased in average by 69%.

In June 2013, readers from the online newspaper *Zambian-watchdog.com*, from Zambia, reported being unable to access the webpage [60]. The newspaper is considered the fourth most popular webpage in Zambia, after Facebook, Google, and YouTube. Tests performed using the OONI tool [61] revealed that the newspaper webpage was the only one being blocked in the country.

The *Web Censorship Monitoring Tool* (WCMT) was presented in the Master dissertation of Shadi Esnaashari [62]. The tool was employed, from July to September 2013, to detect the blocking of services and webpages by ISPs and different organizations networks, in Wellington, New Zealand. Results show that all evaluated organizations and ISPs blocked some kind of content, but the variety of the types of contents blocked by different networks was very large. The author claims that this variety shows a lack of criteria for defining which contents should be blocked.

In 2013, Shankesi proposed, in his Ph.D. thesis, an infrastructure for detecting network manipulation called Friend-sourcing [63], based on crowdsourcing. This infrastructure allows users to get help from their social network contacts to detect whether its traffic is being tampered with. The author conducted experiments with 54 users in India. Results show that 64 web addresses were blocked by several ISPs in the country.

In February 2014, a user from the Reditt discussion forum reported a case of traffic throttling when using a VPN service, specifically the OpenVPN standard port [64]. The user claimed that if another port was employed, then traffic was not throttled. Several other users confirmed this report. Also in 2014, Brodkin reports that the average throughput of the Netflix service in the Verizon and Comcast ISPs decreased during three to four months [65].

On February 1st, 2016, van Schewick sent a report to the FCC president, stating that the *Binge On* service from the T-Mobile ISP was violating NN by hindering freedom of expression on the Internet [66]. According to van Schewick, in November 2015, T-Mobile, the third largest mobile ISP in the USA, launched this service (*Binge On*) which offered unlimited video streaming from 42 selected providers, such as Netflix, Amazon, Hulu, and HBO, without accounting on the monthly data caps, a practice called *zero-rating*. The author claims that this practice is a case of TD, since the ISP is favoring a set of services over others. In February 7th, 2016, the Verizon ISP is also accused of violating NN by practicing zero-rating with the mobile video service *Go90* [8]. This service did not account video traffic from Verizon itself to monthly data caps.

On March 2nd, 2016, *Public Knowledge*, a non-profit organization which defends NN and other user rights on the Internet, registered a complain to the FCC regarding the *Stream TV* service from the Comcast ISP [67]. The complaint accuses Comcast of zero-rating the service. Public Knowledge also requested the FCC to interrupt the service [68].

In March 24th, 2016, Netflix declared that it had limited the rate of its own video streaming to 600 Kbps for users accessing the service from mobile networks [29]. According to Netflix, the purpose of this practice was to protect users from additional charges due to exceeding data caps. However, this practice was not employed for consumers of at least 2 ISPs in the USA, since "historically those two companies have had more consumer-friendly policies". In March 25th, 2016, the *American Cable Association* (ACA) declared to be against this practice from Netflix [69]. According to ACA, the FCC should also investigate content providers and review the NN regulations to include restrictions against content providers, in addition to the restrictions against ISPs. The FCC stated in reply that although current regulations do not include content providers, the behavior of Netflix adds new components to the NN debate and regulations [70].

On April 1st, 2016, a group formed by more than 50 organizations of public interest and consumer protection requested the FCC to take action against zero-rating practices [71] adopted by ISPs in the USA, such as Verizon, AT&T and T-Mobile. According to the group, these practices harm free competition, innovation, limit consumer choice, and increase prices.

These NN violation cases which took place in the 5 continents make it clear that the subject is very complex, and that monitoring and enforcing compliance with NN regulations has become a truly critical task worldwide.

III. Traffic Differentiation: Definitions

In this section we present the definitions that serve as a basis for the rest of the survey. After giving a brief overview of traffic differentiation in the Internet, we define what a neutral network is. Next, we describe how TD may be implemented by an ISP. Finally we we define the TD detection problem.

A. Internet Traffic Differentiation in the Context of Network Neutrality

The Internet is a global network that consists of several interconnected Autonomous Systems (ASes). Each AS comprises a collection of Internet routing prefixes and is controlled by an administrative entity called Internet Service Provider (ISP). ISPs are hierarchically organized in three tiers. Tier 1 ISPs correspond to the core Internet backbone, which consists of high performance networks which interconnect the tier 2 ISPs on a global scale. Tier 2 ISPs provide global connectivity

to the tier 3 residential ISPs, which provide Internet access to end-hosts. An end-host is any computer or device connected directly to a tier 3 ISP or a gateway providing Internet connectivity to a local network. End-hosts form the so-called edge of the Internet.

The Internet was originally designed following two principles that are essential in the context of NN [72]: the *end-to-end* principle and the *best-effort* principle. The end-to-end principle states that messages exchanged between two end-hosts are sent in packets that are forwared by autonomous routers. A router simply forwards a packet to the next hop so that the packet will reach the destination through the shortest path. In particular, a router cannot define or control the complete route that a packet traverses from the origin to the destination. The best-effort principle states that every packet must traverse the network as fast as possible. A router employs a queue to manage the incoming packets. If the queue grows and uses all the space available, the router should drop the next incoming packets, regardless of their content, origin, destination or any other feature.

Actually, there exist several different scheduling algorithms both for dropping arriving packets and for determining which packets should be forwarded and removed from the buffer. Some of the most common types of schedulers are [33], [40]: (i) First Come First Served (FCFS), in which the packets that arrived first are forwarded first; (ii) Strict Priority (SP), in which the scheduler always give priority to a specific type of traffic; (iii) Leaky Bucket, in which maximum rates are defined for each type of traffic; (iv) Token Bucket, in which a limit is defined for the average rate of each type of traffic; (v) Weighted Fair Queuing (WFQ), in which the maximum rates for the different types of traffic are based on weights; (vi) Drop-Tail (DT), which drops all new incoming packets when the buffer is full; and (vii) Weighted Random Early Detection (WRED), in which low priority packets have a higher probability of being dropped.

We call *neutral schedulers* those that do not differentiate traffic. FCFS and DT, for example, are neutral schedulers. Non-neutral schedulers are those that may be employed to discriminate between different types of traffic, either by dropping or delaying packets that are classified as low priority. For example, the Leaky Bucket scheduler might be employed to enforce a maximum rate of a specific type of traffic. Active Queue Management techniques (AQM) [73] may also be employed to differentiate traffic.

Note that according to existing NN regulations [74], some traffic management practices are considered "reasonable" even if they prioritize or degrade different some types of traffic. These are thus exceptions in which traffic differentiation is allowed [75], [76]. Usually, a traffic management practice is considered reasonable if it is beneficial to the network and its users as a whole. Examples include addressing illegal content (e.g. piracy, spam, or viruses), or prioritizing DNS queries. Another reasonable practice, considered by some regulations, is to prioritize the so-called *specialized services* [74], in order to meet their QoS requirements. An example of a specialized service is real-time health services (e.g. remote health monitoring). The main focus of this work is on TD detection, thus determining whether TD is legal/beneficial is out of the scope of the work.

B. Definition: Network Neutrality

There is no unique definition for NN, several different definitions can be found in literature [23], [77]–[80]. However, it is possible to say that most definitions, including those employed by regulations worldwide take into account whether TD is going on in the network. Thus, a network is defined as neutral if all data packets are treated equally in that network, i.e. unreasonable TD practices are not allowed. Therefore, an Internet Service Provider (ISP) cannot slow down, prioritize or block any type of specific traffic, regardless of its origin, destination and/or content. In addition, a NN *violation* corresponds to any unreasonable practice that causes some particular traffic to be treated differently from others.

The Internet end-to-end and best-effort principles are behind the NN definition, and they imply that all routers should forward every packet in a neutral fashion, without prioritizing any subset of packets over others [72]. Every type of traffic is then subject to the same conditions. Therefore, in a neutral network, all routers must employ neutral schedulers. For instance, if all routers in an AS employ only the FCFS and DT schedulers, the network is neutral [33], since packets will always be forwarded and dropped (if the buffer is full) in the order they arrive, regardless of other features.

C. Implementing Traffic Differentiation

There are several mechanisms for implementing TD in a network. Each ISP might employ different mechanisms that better suit its own interests and the characteristics of its network. However, regardless of the specific mechanisms, TD can be employed by an ISP in one or several routers, at the ingress or egress points, or in internal routers.

Figure 1 shows a high-level description of how TD can affect traffic between two end-hosts. The figure is agnostic to the actual characteristics of the network and specific TD mechanisms employed. The ISP network employing TD is divided into different logical components. Any traffic traversing the AS is classified and treated accordingly to the assigned class. A control component defines how traffic is classified and differentiated. These logical components run on top of the actual physical network and may be implemented in several different ways.

Traffic classification may be based on several criteria [81], [82], such as TCP/UDP port, source address, destination address, application protocol, information obtained from deep packet inspection (DPI), the previous-AS from which the traffic came or the next-AS to which the traffic will be forwarded, traffic performance or behavior, a combination of these or any other more complex criteria. Furthermore, these criteria may change over time. Based on the classification, traffic is treated differently according to the assigned class.

There are several ways for implementing traffic classification on a real network. For instance, classification can be run at the ingress point of an AS and class information can be inserted in the packet header (of some AS internal protocol),

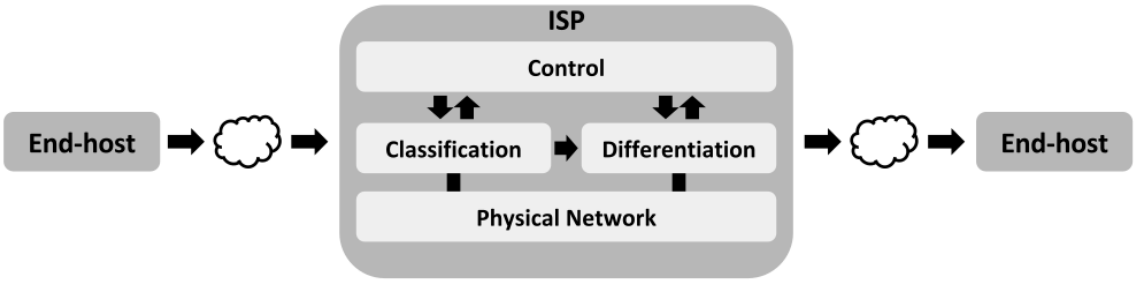


Figure 1. High-level description of TD on an ISP network.

informing the following routers how that packet/traffic should be treated. Another strategy is to configure all routers to both classify and discriminate traffic.

TD may also be employed using several different mechanisms and deployed in several different configurations. Furthermore, these mechanisms may change over time. For instance, as with classification, TD may be performed only at the ingress point of an AS, by all or by some routers in the network. Another possibility is to have specialized devices called middleboxes [83] to perform TD.

The most common TD mechanisms are traffic shaping [34] and traffic policing [35]. These mechanisms differ in the way routers process incoming packets given their classes. Traffic policing employs non-neutral schedulers to limit the rate of low-class traffic by dropping packets more often. Traffic shaping limits the traffic rate by delaying low-class packets, employing schedulers that prioritize high-class traffic when forwarding or dropping packets. Other examples of TD mechanisms include: forged TCP reset (RST) packets injection, forcing TCP connections to abruptly end; and forwarding traffics through separate paths depending on their classes, one of them being purposely less congested, the so-called fast-lane [6].

Different TD mechanisms may affect traffic in different ways. For instance, traffic policing may result in larger loss rates, while traffic shaping may result in larger delays. If packets of two different classes are forwarded along two different routes, and just one of them is congested, the packets will experience significantly different delays and loss rates.

An ISP may dynamically control how traffic is classified and differentiated in its network. There are different possible approaches for implementing the control module of Figure 1. For instance, it can be a person or a system that automatically reconfigures routers and other devices in the network according to some predefined criteria. We note that SDN is a technology that allows sophisticated classification and differentiation mechanisms to be easily deployed and managed [84].

D. TD Detection: Problem Definition

In this work we address the problem of detecting TD in the context of NN. We assume an external observer that does not have access to network configurations and internals. The problem consists of inferring whether some given network traffic is being treated differently from other traffic. In other words, the TD detection problem consists in determining whether different types of traffic are experiencing different performance levels due to unreasonable discrimination in the network only because of their different features (e.g. source, destination, port, content, etc.) A related problem is to identify exactly which features are triggering TD. Another related problem is to identify where (in which AS or ASes, router or routers) the TD occurred.

TD may not always impact the traffic traversing a network, such as when there is not much traffic and the discriminatory practices do not result in any extra delay or loss. In such cases, we say that TD is *non-observable*, since it is not feasible to infer whether TD is being employed, at least based only on external observations. Similarly, TD is *observable* when it effectively impacts the network traffic, for example by increasing its delay or loss rate.

IV. A Survey of Tools and Solutions for TD Detection

In this section, we present several solutions for the problem of detecting TD on the Internet. These solutions were designed with different goals and assumptions, employing thus different techniques to achieve such goals under the assumptions made. They rely on network measurements for inferring the presence of TD. This is often achieved by checking if different types of traffic were treated differently while traversing the network.

The rest of this section is organized as follows. We describe several existing solutions for the problem of detecting TD on the Internet, from subsection IV-A to IV-J, in order of publication date. We finish the section presenting other works related to monitoring NN in subsection IV-K.

A. Gnutella Rogue SuperPeer (2007)

The *Rogue SuperPeer* (RSP) [42] is a strategy to measure port blocking in the Internet. This strategy detects whether traffic on specific ports (corresponding to specific applications or classes of applications) is being blocked between end-hosts and a measurement host. Port blocking is an important and straightforward strategy that can be used by network operators to control which type of traffic is allowed on their networks. Although it can be used for fair reasons, such as blocking worms, it can also be used for anti-competitive or economic purposes, for example an operator can block services with which it is competing.

The main principles behind the design of the Rogue SuperPeer are: generality, range, quantity, and minimal participation, described as follows. By generality the authors mean that any arbitrary TCP or UDP port number (from 0 to $2^{16} - 1$) can be tested. By range means that a large range of networks across the Internet are tested. The quantity is a large number of hosts are tested. Finally, minimal participation means that the participation is not active, coordinated, or cooperative, users are engaged in the process of testing without even noticing it.

The RSP infrastructure consists of two separate machines: the Rogue SuperPeer itself and a measure host. The Rogue SuperPeer itself is a superpeer of a P2P network, in this case the authors used Gnutella[1]. This superpeer joins the network and is advertised as any other superpeer. When a new peer connects to the RSP, they issue queries and responses according the normal protocol. However, the process is slightly modified so that these new peers will trigger port blocking measurements. The main idea is to induce a large number of globally distributed hosts to attempt connecting to a specific IP address using the TCP ports being evaluated.

Gnutella is a P2P network comprised of two types of hosts: superpeers and peers (also called clients or leaves). Each superpeer is connected to other superpeers and to a set of peers. In order for a new peer to join the overlay network, it first contacts a superpeer. The superpeer may then accept or reject the new peer connection. If the peer request is accepted, it stays connected to that superpeer. However, if the connection attempt is rejected, the superpeer replies with a "busy" message. This response includes an indication of other superpeers (IP/port) that might be contacted by the new peer to join the network.

In the RSP strategy, after a new peer sends a connection request to the Rogue superpeer, the superpeer sends back a "busy" reply, and refers the new peer to the measurement host, using a particular port to be evaluated. Figure 2 illustrates how the RSP strategy works. A new peer sends a connection request to the RSP (1). The RSP then refuses the new peer, replying with a busy message referring the new peer to the IP address of the the measurement host and the port to be evaluated. The new peer then initiates a connection with the measurement host (2). The port number referred by the RSP changes every 5 minutes, in order to evaluate a large number of ports. The measurement host and the superpeer both register incoming connections from the new peers.

Determining whether a port is *not* blocked is done as follows. If at least one peer, redirected by the RSP, successfully connects to the measurement host, then the port used for this connection is not blocked. However, if no peers connects to the measurement host on a given port referred by the Rogue superpeer, there are two possibilities: either all peers ignored the referral, or the port is blocked. The authors empirically concluded that the probability of a new peer ignoring the RSP referral is about 80%. The authors then determined that at least 50 referrals are necessary to infer that a port was blocked, with a confidence level of 99.5%.

[1]http://www.gnutellaforums.com

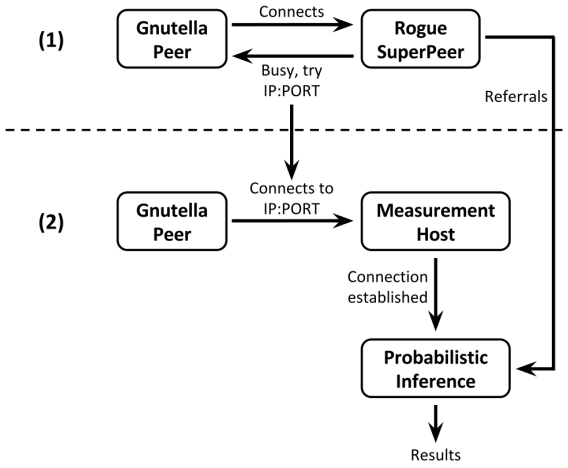


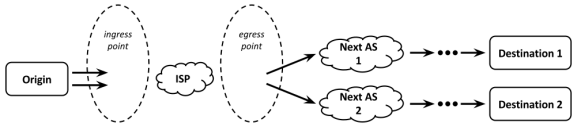
Figure 2. Gnutella RSP strategy.

Experiments with the RSP strategy were executed for 2 months. During this period, approximately 150,000 referrals were generated for about 72,000 distinct Gnutella peers, which were distributed in approximately 31,000 different prefixes, which can be considered a significant fraction from Internet. The results show that of the 31,000 prefixes, in 256 at least a port was blocked. The most frequently blocked port was 136 and those blocked less frequently were 80 (HTTP), 6346 (Gnutella), and 6969 (which was used for comparison). Some email ports (25, 110 and 143) were blocked twice as often as the comparison port 6969. The other most frequently blocked services were: FTP, SSH, Bittorrent and VPNs. The authors also report that some universities and ISPs blocked ports often used by P2P networks (1214, 4662, 6346, 6881). Furthermore some ISPs in the Canada, the USA and Poland blocked Skype ports.

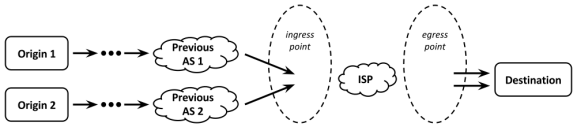
The RSP strategy addresses a specific case of TD, which is port blocking. It also addresses the subproblem of locating which ISP is performing TD, by aggregating several measurements from a given prefix. The strategy is based on hybrid active/passive measurements and must be executed on a P2P network, such as the Gnutella network. However, some issues are not addressed by the authors. For instance, the Gnutella RSP strategy cannot always tell whether a port blocking is being performed by the peer ISP or by the ISP of the measurement host. Furthermore, an ISP may be blocking all Gnutella traffic based on the application protocol, regardless of port numbers.

B. NetPolice (2009)

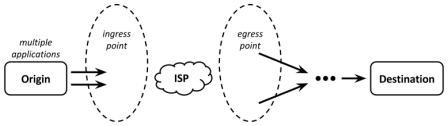
NetPolice [37] (a previous version of which was named NVLens [85]) is a tool for detecting TD in the backbone of the Internet (Tier 1). The authors argue that when TD is executed in the backbone the impact is stronger that when it is executed by ISPs that are closer to the border, since TD in the backbone potentially affects a larger amount of Internet traffic. NetPolice is able to locate which ISP is performing TD. The tool measures the loss rate experienced by different types of traffic, sent from multiple sources, as they traverse a target



(a) TD based on next AS.



(b) TD based on previous AS.



(c) TD based on destination port or payload.

Figure 3. How NetPolice detects different types of TD.

ISP. TTL-based probes are employed in order to discover paths traversed by packets in the network, including the internal path of the target ISP.

NetPolice detects TD triggered by content and routing, assuming that traffic is classified based on header, payload, or using routing policies. Figure 3 shows how NetPolice detects different types of TD. In Figure 3a, measurements are made using the same source to multiple destinations. These destinations are chosen in a way so that after the packets leave the target ISP they enter different ASes. This strategy allows NetPolice to check if the target ISP is employing TD based on the next AS of the packets. In Figure 3b, measurements are made using a single destination and multiple sources, selected in such a way that the packets entering the target ISP come from different ASes. This allows the tool to check if the target ISP is employing TD based on the previous AS from which the packet came. Figure 3c, show yet another case in which measurements are made using the same source and the same destination, but traffic is generated for multiple applications (changing destination port or payload). This allows the tool to detect TD triggered by application/content.

The strategy employed by NetPolice to detect TD is based on 4 steps and is shown in Figure 4. The first step consists in discovering paths that traverse the target ISP, from multiple origins. A large number of route traces (using for example the traceroute command) are issued from multiple sources to a large number of Internet destinations (prefixes). This process allows NetPolice to estimate the the distances between ingress and egress points of the target ISP, as well as the previous AS to the ingress points as well as the next ASes to egress points. With this information NetPolice pre-computes the TTL values to reach each pair of ingress/egress points of the target ISP, from all sources. The set of paths discovered and related information obtained in this step is called “path view”.

In the second step, a set of paths on which measurement are to going to be executed are selected from the “path view”, as it is unfeasible to run measurements on all paths. This set of paths should give a good coverage of the internal network of the target ISP. In order to avoiding unnecessary work, the choice of which paths to execute measurements must be clever, in order to avoid source and destination pairs that that pass through the same ISP internal paths or paths that do not traverse the target ISP. This selection is modeled as an optimization problem, with the following constraints: each tuple (origin, input, output) must be traversed at least R times by paths to different destinations; each tuple (input, output, destination) must be traversed at least R times by paths from different sources; finally, there can be no more than m paths from the same source. The set of paths on which to execute measurements is called “tasks” and is sent to the next step of NetPolice.

Measurements are executed in the third step, using traffic generated for different applications: HTTP, BitTorrent, SMTP, PPLive and VoIP. Measurements are executed as follows. Periodically, at each 200 seconds, and for each application, two measurement probes are sent: one with TTL set so that its reaches the ingress point (*in*), and another with TTL set to reach the egress point (*eg*). The loss rate for the internal path of the target ISP is then obtained by subtracting the loss rate measured for the egress point from the loss rate measured for the ingress point.

Finally, in the fourth step, NetPolice uses the obtained measurements to infer whether the target ISP is employing TD based on content or routing. The inference employs the KS (Kolmogorov-Smirnov) test in order to compare the distributions of the measured data. The detection of TD by content is then done by comparing the data distributions of each application with the distribution of HTTP application data. NetPolice assumes the HTTP traffic is a baseline for detecting TD, i.e. HTTP traffic is assumed not to be discriminated. KS tests are applied to determine if the measurement data obtained for a given application is significantly different from the data measured for the HTTP-based application, thus characterizing TD. TD based on routing is detected in a similar way, but comparing the distributions for data obtained for different paths and the same application.

The authors discuss the reduction of noise effects from different points of view. Inaccuracy of loss rate measurements can be caused by an overloaded prober, especially due to high CPU utilization. The authors mention that a reasonable limit is 65% CPU utilization. Another factor to consider is ICMP rate limiting, which is done to prevent router overload; NetPolice avoids this problem by keeping a large probing interval, on the order of hundreds of seconds. Another noise reduction factor to consider is the loss on the reverse path. As NetPolice uses single-ended probes to to measure loss rates, they can be inflated due to reverse path loss. The authors report an experimental result that the loss rate increases with the packet size; they thus use the loss rate measured by 40-byte probe packets as the upper bound of the loss rate on the reverse path. Finally, although some ISPs perform load balancing using ECMP (Equal Cost Multi Path) between a pair of ingress/egress points to improve the performance, and this can be a problem given the measurement strategy of NetPolice, it was not detected in any ISP evaluated.

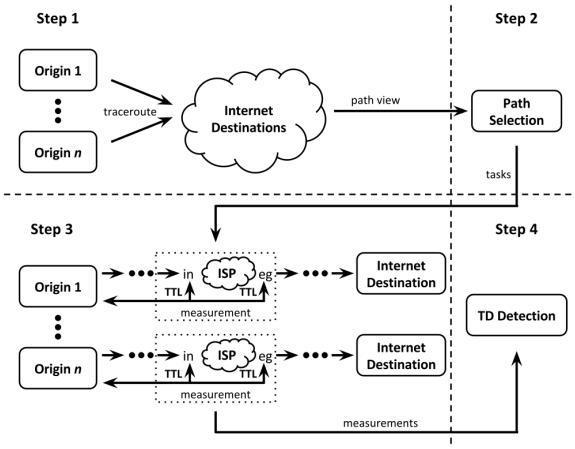


Figure 4. NetPolice TD detection.

Experimental results reported for NetPolice were obtained in the PlanetLab. 18 ISPs distributed across 3 continents were evaluated over a period of 10 weeks. The results show that 4 ISPs performed TD on 4 applications and 10 ISPs performed TD based on the previous AS of the packets. The packet loss rates measured in these cases were up to 5% different. The authors also observed, from the results obtained, that TD can depend on the load of the network. For some ISPs, NetPolice detected that the values assigned to the TOS field of the IP packet header were strongly related to TD, and this assignment of values is usually based on the destination port not on content (thus DPI was not done). Another observation was that different routers do apply TD in the same way.

NetPolice addresses both the problem of detecting TD and the subproblem of locating which ISP is performing TD. It is also one of the few solutions that detects TD triggered by the path traversed by the traffic. Note however that the TTL-based probes techniques employed may result in false-negatives, e.g. because the ICMP protocol is not supported. Another possible limitation of the NetPolice strategy is the set of paths on which measurements are executed. The paths traversed between the same origin and destination, by different types of traffic, may not be the same. Thus, the paths obtained in the path discovery step may not be the same path traversed when the measurements are made for different applications. For instance, the packets may be forwarded to an egress point different than expected.

C. NANO (2009)

NANO (Network Access Neutrality Observatory) [32], [86] is the first system that detects neutrality violations that does not test specific applications/ports nor specific discrimination mechanisms. NANO infers whether an ISP is discriminating traffic based on the performance data obtained passively. If the performance of an application measured in an ISP network is statistically significantly lower than the performance of the same application measured in the networks of other ISPs, it is possible that TD is being employed. NANO uses a causal inference model to establish a relationship between observed performance degradation and the ISP policies. NANO employs passive monitoring, i.e. it is based on measurements of the real traffic of the observed applications while they are running.

Some of the main features of NANO for TD detection are, according to the authors: (i) several other other strategies detect discrimination based on specific traffic characteristics such as port or content, whereas NANO has a more general approach, measuring the performance of the applications regardless of the specific TD mechanisms employed by the ISPs; (ii) the fact that NANO passively monitors traffic makes it more difficult for ISPs to detect that NANO is being used and escape; and (iii) while NANO compares metrics from the same application executed on different ISPs, several other solutions compare different application metrics in the same ISP.

NANO's TD detection strategy presents three major challenges: (i) the TD mechanism employed by the ISP is not known in advance, so the detection strategy must be generic; (Ii) the standard performance of an application at a particular ISP is not known beforehand, making it difficult to detect possible degradations, since there is no baseline for comparison; and (iii) many factors other than TD can cause application performance degradation, such as overhead, geographic location, the particular software and hardware being used, and other network features.

The different factors, besides TD, that can cause degradation in the performance of an application, are represented in the statistical model used by NANO by confounding factors [87]. Thus, it is necessary to identify the confounding factors and to collect data not only from the applications, but also from these confounds. NANO TD detection strategy is therefore based on the comparison of the performance of the same application executed on different ISPs using measurements with similar confounds. An example of a confounding factor is the time of day: one should not compare measurements taken at different times, since application performance varies depending on the time of the day (due to a higher/lower system load, for example).

NANO uses a stratification technique [88] to group performance measurements according to the corresponding confounds. This technique places measurements in strata, so that the confounds for the measurements in each stratum have similar values. Three types of confounding factors are defined: (i) client-related confounds (examples include software that may affect the performance of the measured application, such as the operating system or a specific Web browser); (ii) network-based confounds which are related to the network (such as properties of the network path, or geographic location, for example); and (iii) time-based confounds (such as time of the day, for example, that can affect the performance of the application being measured).

After the stratification, NANO estimates for each stratum how much the performance of the application changes when accessed through an specific ISP, in comparison with the performance observed when not using that ISP, which is called the baseline performance. The average performance is computed as the average performance of the application executed on all other ISPs within the same stratum, except the ISP being evaluated. These estimates represent a quantification of the causal relationship between each ISP and the possibility

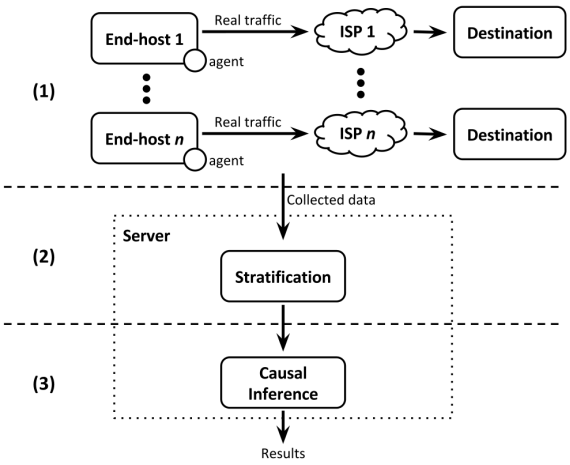


Figure 5. NANO TD detection.

that TD is being executed.

In the last step of its TD detection strategy, NANO aggregates the estimates of all strata and verifies if the values obtained are statistically significant. The central idea is that if, on average, the performance of an application was significantly degraded on a specific ISP, then there is a causal relationship between the ISP and the practice of TD.

NANO is implemented in two parts: the agents and a server. An agent runs on each client host and is responsible for monitoring application performance by measuring its real traffic from that client host. The metrics employed are specific to each application, whichever is most appropriate for each one. In addition to application performance data, agents also collect data about the confounding factors. All data acquired by agents is periodically sent to the server. Agents are implemented as network sniffers, analyzing all packets received and sent by the host. The NANO server receives all data collected by the agents and is responsible for performing TD detection based on this data.

Figure 5 shows how NANO works. Agents are deployed on several end-hosts, each host executes an agent (1), being responsible for passively monitoring the performance of running applications being evaluated. Data obtained by all agents is sent to a server, which classifies them in strata according to the confounds (2). Finally, the server infers (3), according to a causal model, which ISPs employed TD for each application being evaluated.

To evaluate NANO, the authors executed experiments using the PlanetLab and Emulab testbeds. Geographically distributed PlanetLab nodes were employed. These nodes executed servers of the applications evaluated. A set of ISPs was created in Emulab, each with a different set of clients. Each ISP provided Internet connectivity to its clients. Thus, clients could only access the applications hosted on PlanetLab nodes through the ISPs, allowing the emulation of different TD practices and different confounding factors.

Experimental results showed that NANO is able to detect TD in different ways and for different types of applications, provided that all confounding factors are known and measured. The NANO detection strategy proved to be generic enough,

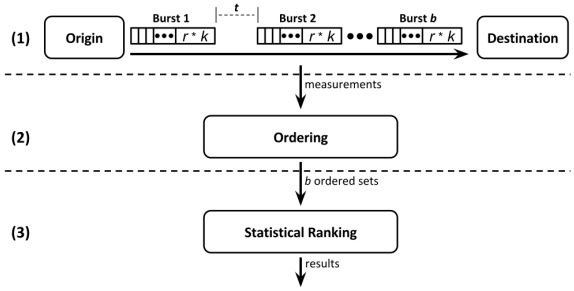


Figure 6. POPI TD detection.

detecting traffic discrimination even without prior knowledge of which TD policies were employed by the ISPs. However, if NANO does not take into account all confounds, the causal relationship between the ISP and TD can lead to mistakes, either false-negatives and false-positives. As it is impossible to to identify *all* relevant confounding factors or even to decide whether a given set of confounding factors is enough, it is not straightforward to apply NANO to detect TD in complex real networks.

D. POPI (2010)

POPI [36] is a tool that uses end-to-end measurements to detect whether non-neutral schedulers are being employed by an ISP. In particular, POPI detects whether packets of different types are being forwarded with different priorities, i.e. POPI detects packet forwarding prioritization. The authors assume that TD is performed by using a non-neutral scheduler, which were described in Section III. If only neutral schedulers are employed, then all packets are forwarded according to the arrival order. On the other hand, if a non-neutral schedulers are being used, the loss rates will be different for different types of traffic under congestion. POPI measures the loss rate for different types of traffic to infer whether different priorities were assigned for the types of traffic measured.

POPI works in three steps, shown in Figure 6. In the first step (1) measurements are obtained after a series of packet bursts are injected. In the second step (2) the measurements are used to order the types of traffic with larger loss rates for each burst. In the third step (3), a statistical analysis is made to detect the prioritization of specific types of traffic during the bursts. Each step is described in more details below.

In the first step (1), POPI measures the loss rate for k different types of traffic, generated with different destination ports and/or payload. These measurements are made on b bursts of packets, between a pair of end-hosts. The bursts are triggered in intervals of t seconds. Each burst is composed of r rounds, and in each round k packets are sent – one for each type of traffic, in random order. Thus in each burst $r * k$ packets are sent in sequence. According to the authors, the value of t should not be too low, so a burst does not interfere with the next, but should also not be too high, so that the whole measurement process ends within a too short period of time, making it less susceptible to cross-traffic variations. The authors also claim that it is necessary to send a large number

of packets to saturate the path between the end-hosts causing congestion and thus packets are dropped.

In the second step (2), the loss rates for all types of traffic in each burst are computed and ordered. The traffic type with the largest loss rate in any given burst is put in the first position, the traffic type with the second largest loss rate is put the second position, and so on. According to the authors, if all types of traffic are treated equally, the positions of different types will be random for the different bursts, since packets of different types are sent randomly in each round. However, if some traffic types have low priority, they will always be in the first positions after the ordering (higher losses). At the end of this step, there will be b sets of types of traffic, ordered according to the the loss rate observed for each burst.

In the third step (3), a statistical analysis is made to verify if there was prioritization of any specific type of traffic along the bursts. According to the authors, if POPI compared only two different types of traffic, it would be enough to just compare the measurements obtained for each type to determine if one had a different priority of the other. However, to compare more than two types of traffic it is necessary to group them according to the assigned priorities. In order to check whether the relative positions of k measurements are consistently repeated over b observations, the statistical *Problem of N Rankings* [89] can be applied. The solution adopted POPI consists of computing the average position for each type of traffic over all bursts (*Average Normalized Ranks*) and group with a hierarchical divisive method the traffic types whose averages are similar. This process results in groups of types of traffic ordered according to their priorities. In a neutral network, this would result in a single group, since all types would have a similar average, i.e. the same priority.

To evaluate POPI, the authors first performed simulations using the NS2 network simulator. In these simulations two pairs of end-hosts were used. One of these pairs was responsible for simulating background traffic, while the other pair simulated the execution of POPI. In the topology used in the simulations, the communication between the two pairs crosses the same two routers, responsible for simulating the prioritization of certain types of traffic, with a maximum bandwidth of 100 Mbps. The simulations were executed for $k = 32$ traffic types and $b = 32$ bursts. Incremental values were used for r - the number of rounds per burst. The upload rate for background traffic ranged from 10 to 90 Mbps. The results obtained in these simulations showed that POPI was effective even in the presence of a large amount of background traffic: low priority packets were always dropped before the those with higher priority. Another result was for the value of r. When $r < 18$ the measurement traffic was not able to cause congestion, thus no losses were observed making impossible to infer anything. As the value of r increases, losses are observed more frequently for low priority traffic. Based on the results, the authors state that $r > 30$ is sufficient to obtain reliable results. Thus, $r = 40$ was used in the experiments executed in the PlanetLab, described below.

Experiments were conducted in the PlanetLab to evaluate POPI in a real environment and to find possible real cases of prioritization. In these experiments 162 nodes of the testbed were employed, spread around the world. POPI was executed on all pairs of nodes and in both directions for each pair. The values used for the variables were: $k = 26$, $b = 32$, $r = 40$ and $t = 10s$. The size of the packets employed was 1500 bytes, which generated an average bandwidth consumption of 1.04 Mbps. The results detected traffic prioritization for 15 node pairs. The authors also ran experiments using other metrics besides the loss rate. Unfortunately although these other metrics present lower overhead, they were not able to detect most of the prioritization cases that had been detected in the experiments based on the packet loss rate.

E. DiffProbe (2010)

The Differential Probing (DiffProbe) method [33] detects delay and loss differentiation. By using statistical methods, DiffProbe is able to detect that a non-neutral scheduler is being used, results are reported for schedulers SP (Strict Priority) and Weighted Fair Queueing (WFQ). Furthermore DiffProbe is also able to detect packet dropping policies, results are presented for WRED (Weighted Random Early Detection). According to the authors, the proposed method is a new class of network tomography. DiffProbe assumes that an ISP classifies each packet as either high (H) or low (L) priority; low priority traffic suffers longer delays and higher losses according the scheduler and packet dropping policy adopted by the ISP. DiffProbe compares an application flow with a probing flow, both sent simultaneously. The main idea is that if one of the flows is treated differently, a difference in performance will be observable if they are sent at the same time. DiffProbe measures the loss rate and the delays of two different flows sent simultaneously between a client and a measurement host.

The application flow is generated based on recorded real traffic, results are presented for two applications Skype and Vonage. The application flow employs the same transport protocol, packet sizes, ports, payload and transmission intervals as the original traffic. The probing flow is used as a baseline for comparison. The authors claim that this baseline traffic should be different enough from the application traffic, so it is not classified the same way. However, the probing flow must have features in common with the application flow, such as packet sizes, so that performance results for both flows can be compared. The probing flow is generated as the application flow is sent through the network. In order to detect port or application differentiation, then the probing flow can have the same size as the last packet sent in the application flow, while the payload is random and the destination port are different. It is assumed that this destination port is not likely to be discriminated, i.e. has high priority at the ISP under test. If the discrimination is based on other features such as packet sizes, packet inter-arrival times, etc. then the rate of the probing flow must also be randomized.

The two flows are sent at the same rate at first. After an interval, the sending rate of the probing flow is increased. The idea is to saturate the link with a larger amount of packets from the probing flow. DiffProbe never alters the sending rate of the application flow, since that might change the ISP classification of that particular type of traffic. TD detection

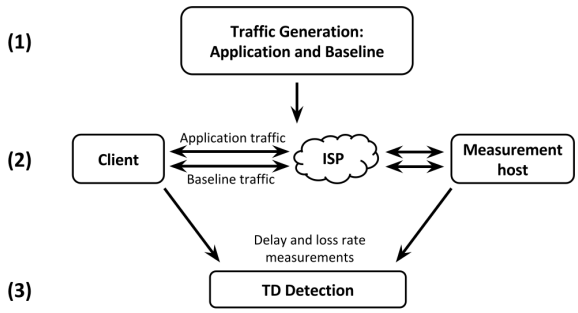


Figure 7. DiffProbe TD detection.

is made comparing the measurement distributions of the two flows. DiffProbe employs the Kullback-Leibler divergence for delay distributions, and the two-proportion z-test for loss rate distributions.

Figure 7 shows how DiffProbe works in three steps. The first step (1) consists in generating the two flows, which are sent in the second step (2) simultaneously from a client to a measurement host, and afterwards in the opposite direction from the measurement host to the client. In the third step (3), the measured delays and loss rates from both flows are statistically analyzed in order to infer whether TD was employed or not.

The authors evaluated DiffProbe using the NS-2 simulator and also emulation, with a client connected to a residential ISP and a server hosted at an university. TD was emulated by a router between the end-points. Both simulations and experiments in the emulated environment showed that, whenever TD was observable and the amount of generated traffic saturated the link employed, the detection was accurate.

DiffProbe detects if a specific application traffic is being discriminated between two end-hosts. It assumes that the baseline traffic is not discriminated, which may lead to false-negatives if this is not true. Furthermore, DiffProbe requires path saturation, which represents significant network overhead.

F. Glasnost (2010)

Glasnost [38] is a tool that allows Internet end-users to check if their ISPs are employing TD. It was designed as an easy-to-use tool that can be accessed via Web and requires no technical knowledge. Glasnost has already been used by thousands of Internet end-users around the world. It was initially designed for detecting TD on BitTorrent traffic, but can also be used for to detect differentiation on any traffic of any application. It was shut down in May 2017.

Figure 8 illustrates the usage of Glasnost. Initially, the end-user accesses the Glasnost webpage[2] and is redirected to a measurement server (Figure 8a). Users may be redirected to one of several different measurement servers, making it harder for ISPs to employ techniques against a specific server. The user then downloads the client application (Figure 8b). The client application is a Java applet that is executed at the end-user Web browser. The client connects to the measurement server and execute a series of tests (Figure 8c), after which the results are shown to the user.

Glasnost detects that the traffic from a given application is suffering differentiation by sending two flows in sequence, between a client and a measurement server. One flow corresponds to the target application, and the other is the baseline traffic generated for comparison purposes. The application traffic consists of messages of the real application. Glasnost assumes that a given application is suffering differentiation and that the ISP identifies the application based on destination port or application protocol. The baseline flow is identical to the application flow in terms of the number of messages and message sizes, however the the payload is different being defined randomly.

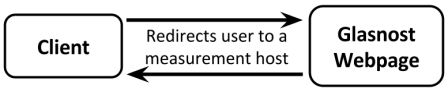
The measurement server computes the throughput for each flow. Each flow between the client application and the measurement server lasts several seconds, long enough for TCP to reach a stable state. The tests are repeated multiple times in order to reduce the noise of the measurements obtained. In the end, the measurement server processes the obtained data and displays the results page to the user. The computed metrics are the minimum, maximum, and median of the measured throughput.

The maximum throughputs observed for each flow are then compared to infer if the flows were treated differently. If the difference is higher than a threshold σ, than Glasnot concludes that TD occurred. The authors claim that this threshold represents a trade-off between the ability of the system to detect TD and the generation of false-positives. For instance, if σ is 50%, Glasnost will detect TD only if the maximum throughput achieved by one of the flows was half the maximum throughput of the other, possibly leading to false-negatives. On the other hand, if σ is small, e.g. 5%, Glasnost can commit a mistake and claim that there was TD when in reality the different might have been because of cross traffic. The authors claim that 20% is a good value for σ.

The authors report that in 2010 Glasnost detected that 10% of BitTorrent users suffered TD. Among the detected cases, differentiation occurred mostly on the upstream flow, with a few cases of TD on the downstream flow. One surprising result is that, after it was concluded that an ISP was practicing TD, only 21% of the ISP users were effectively affected (median). The authors list 3 possible explanations: (i) only users generating a large amount of traffic have been affected; (ii) only some parts of the ISP were affected; and (iii) TD was applied only during specific periods of the day, such as during peak times, for example. The authors also report that about 6% of users claimed that the tool did not detect TD that they believed to be suffering. One possible explanation for this is that the decision to minimize the number of false positives increases the number of false negatives.

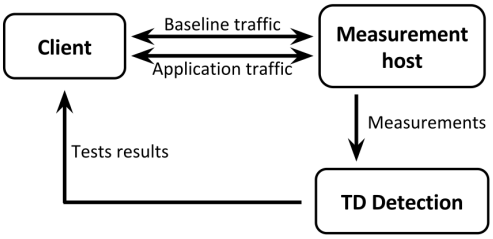
Some of the Glasnost authors had developed earlier the BTTest tool [90] which clearly served as inspiration for Glasnost. BTTest detects if an ISP is blocking BitTorrent traffic. The operation of BTTest is very similar to that of Glasnost, except that BTTest only detects traffic blocking and only for BitTorrent. BTTest was available for a period of 17 weeks in 2008 and 2009, in which more than 47,300 end-users

[2]http://broadband.mpi-sws.org/transparency/glasnost.php



(a) The Glasnost webpage redirects the user to a measurement host.

(b) The user downloads the client application (applet) from the measurement host.



(c) A series of tests are executed and results are returned to the user.

Figure 8. How an end-user makes use of the Glasnost tool.

employed the tool around the world. The results obtained in this period show that in about 8% of the tests BitTorrent was blocked, mainly in the USA. In addition, the vast majority of blockings, about 99%, occurred on upstream data rather than downstream. Another tool, BonaFide [91] is based on Glasnost but focused on detecting TD in mobile networks. The tool was developed for the Android system and works in a way that is very similar to Glasnost, but with some modifications related to restrictions of mobile devices. A BonaFide client application running on a mobile device communicates with a measurement server that runs the tests. Each test consists of flows, as in Glasnost. BonaFide supports several application protocols, such as VoIP and BitTorrent, for example.

In synthesis, Glasnost is able to detect whether the throughput of certain types of traffic are being limited between two end-hosts when compared with a baseline traffic. The measurement technique employed may result in false-positives, depending on the network load and cross traffic, or in false-negatives, if the baseline traffic is classified in the same way as the differentiated traffic. Furthermore, the throughput cannot be used to assess the performance of applications that do not produce high amounts of traffic.

G. Packsen (2011)

PackSen [40] is a system that detects if an ISP is employing a traffic shaper to assign different priorities to different types of traffic. In addition to detecting the presence of TD, the solution also infers which scheduler is being employed and its properties. The main idea is to compare the probability distributions of traffic features at the source and at the destination. If a significant difference is detected it may indicate the presence of traffic shaper between source and destination. Packsen is thus able to detect discrimination based on application protocol, port, time of the day, source, destination, among others.

The solution generates two different flows between two end-hosts, one flow is employed a baseline for comparison, the other flow is from a specific application under test. A basic assumption is that the baseline flow is not suffering any type of discrimination. Packsen generates the two flows interleaved and with exactly the same bandwidth. Packsen then makes measurements to infer the presence of a traffic shaper. Packsen measures the inter-arrival times of packets from both flows, as well as the bandwidth. The main idea is that if there is a non-neutral traffic shaper in the path between the two end-hosts, the arrival of packets of a discriminated flow will present substantial differences from the way they were sent.

Three different statistical methods are employed. The methods are increasingly expensive in terms of computational power required. The first method employs short flows and present low computational overhead and detects the presence of a shaper. This method compares the inter-arrival time distributions of the two flows, using the Mann-Whitney U-test [92]. In case TD is detected, the second method is used which infers which scheduler was employed, and with which parameters, such as the weight assigned to each flow for instance. By comparing the bandwidth required at the source with the the bandwidth required at the destination. The second method is not robust in the presence of cross-traffic, especially when other applications are generating a large amount of traffic simultaneously with Packsen measurements. The situation is particularly complex if the cross traffic has an influence on the classification of the application flow and not on the baseline flow. A third method is then proposed, which is more computationally expensive than the others and consists of repeating the measurements several times until the results are reliable.

Packsen is run on three main types of components shown in Figure 9: a client, an experiment server and measurement hosts. The client connects to the experiment server and requests an experiment to be executed (1). The experiment server chooses one experiment and returns to the client. The client then chooses one available measurement host, informing the experiment which should be executed (2). The measurement host and the client run the experiment and collect data from the traffic generated. The data is then sent to the experiment server where it is stored for further analyses (3).

The authors first evaluated Packsen in a controlled environment, a private testbed. This environment allowed the emulation of several types of traffic shapers, with different parameters, as well as different combinations of cross traffic. Experiments were also executed on PlanetLab on about 1000 hosts in order obtain results in a real large environment. The results obtained in the local testbed showed that Packsen detected, with a low margin of error, both the occurrence of TD as well as the parameters used by the shapers, even in the presence of cross traffic. Only a single false-negative was

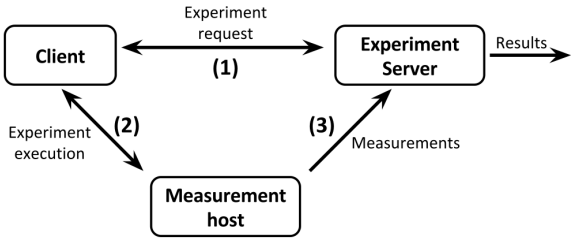


Figure 9. Packsen experiment execution.

recorded in these experiments, in which there was TD but Packsen did not detect it. In the PlanetLab experiments TD was detected in only 0.7% of the tested host pairs (4 out of 518).

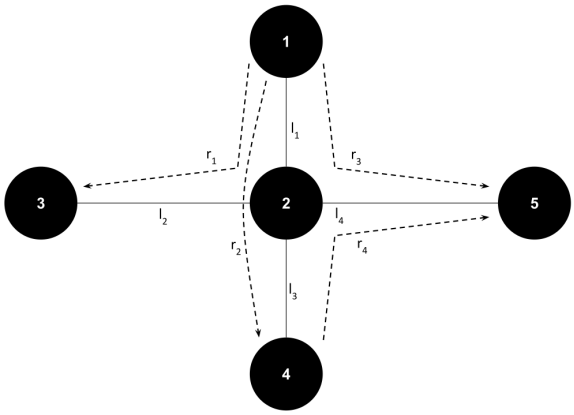
In synthesis Packsen detects the presence of a non-neutral traffic shaper by comparing the performance of an application traffic with a baseline traffic that is assumed to be neutral.

H. Network tomography inference (2014)

An algorithm based on network tomography for inferring the presence of TD in a network is proposed in [41]. The algorithm is capable of identifying exactly on which link, or sequence of links, TD is occurring. The strategy is based on end-to-end measurements, i.e. it just uses external observations without any need for internal network measurements. The authors provide formal proofs indicating under which conditions the algorithm achieves these results. We call the proposed algorithm as the tomography TD solution.

Network tomography [93] was originally proposed to allow the inference of network features such as the delay or loss rate of an internal link, only using end-to-end measurements. Network tomography usually combines multiple end-to-end measurements from different vantage points to infer properties of the network. The tomography TD solution employed by the authors builds a system of equations $y = Ax$ in which y is a vector containing the end-to-end measurements, A is the routing matrix that specifies the relation between network links and end-to-end paths (i.e. it specifies which links are in each path), and x is a vector with the properties to be inferred for each link. An estimate for x is obtained by solving the system, or choosing one solution if there are multiple solutions (e.g. the solution with the highest probability). This tomography technique can only work with additive metrics, i.e. the sum of the measurements for each link of a path must be equal to the measurement obtained for the whole path (end-to-end). As examples, both delay and loss rate are additive metrics.

Figure 10 shows an example of the network tomography technique employed by the solution. Figure 10a shows a network with 5 hosts, with sequential identifiers from 1 to 5, interconnected by links $l_i, 1 \leq i \leq 4$. In the example four end-to-end measurements are executed over paths shown in the figure as $r_j, 1 \leq j \leq 4$. Figure 10b shows the routing matrix A for the measured paths. In this matrix, rows correspond to the paths and columns to the links. An entry of this matrix is set to 1 if the corresponding path traverses the corresponding link, and 0 otherwise. Figure 10c shows the



(a) Example of a network with 5 hosts and four end-to-end measurements.

	l_1	l_2	l_3	l_4
r_1	1	1	0	0
r_2	1	0	1	0
r_3	1	0	0	1
r_4	0	0	1	1

(b) Routing matrix A.

$$y_1 = x_1 + x_2$$
$$y_2 = x_1 + x_3$$
$$y_3 = x_1 + x_4$$
$$y_4 = x_3 + x_4$$

(c) The system of equations obtained from the four end-to-end measurements.

Figure 10. Network tomography technique employed by the solution.

resulting system $y = Ax$. In this system, $y = \{y_1, y_2, y_3, y_4\}$ and $x = \{x_1, x_2, x_3, x_4\}$, y_j corresponds to the measured value for path r_j and x_i being the metric to be estimated for link l_i. For instance, if link l_4 is non-neutral, there may be an inconsistency in the measurements corresponding to paths r_3 and r_4, since they share this link. In this case, the value of x_4 would be effectively different for each of the measurements, resulting in a inconsistent system that has no solution.

This network tomography technique assumes that the network is neutral: all traffic from any path is treated equally on all links. In case this is not true, it becomes impossible to use the measurements obtained from different paths as a function of individual link metrics, and thus the resulting system of equations has no solution. Therefore, while conventional tomography techniques try to build solvable systems, the algorithm used in the tomography TD solution seeks unsolvable systems that reveal NN violations. Thus, the main idea of the algorithm is that, when the network is not neutral, observations made from different vantage points will be inconsistent with each other.

The algorithm receives as input the topology of the network and a set of end-to-end measurements along with the corresponding paths on which the measurements were made. The output is a set of non-neutral link sequences, i.e. on which links, or sequence of links, TD occurred. The end-to-end measurements may use different types of traffic with the

same source/destination, or the same type of traffic with different source/destination pairs, making it possible to identify different TD triggers. It is thus possible to detect TD based not only on content but also on source/destination.

As mentioned above, the algorithm searches for link sequences that result in an unsolvable system. For each sequence of links that are in more than one path, the algorithm builds a system using all the measurements that traverse that sequence and checks if it has a solution. If the system does not have a solution, the sequence of links is non-neutral. If the system has a solution, the link sequence is neutral or TD is not observable (i.e. it is a false-negative). In other words, the algorithm confronts measurements executed on paths that traverse the same part of the network, trying to find inconsistencies that may be caused by TD. The authors claim that this algorithm generates no false-positives, since measurements executed on paths with only neutral links will always result in a solvable system. The authors also claim that the solution generates a small number of false-negatives, in which the algorithm mistakes as neutral link sequences that in reality are not neutral.

To evaluate the algorithm, two series of experiments based on emulation were carried out. The first experiment considered a topology with a single non-neutral link. In this experiment, all measurements were executed through this link. Different scenarios were tested, varying the behavior of the non-neutral link. In all cases the algorithm succeeded in detecting that the link was not neutral. In the second series of experiments a topology with several non-neutral links was used. Each of these links presented a different behavior. As in the first experiment, the algorithm always correctly detected the non-neutral links.

The authors also discuss the challenges to implement this tomography TD solution in a real environment. The most feasible option, according to the authors, is to use several end-hosts that periodically make end-to-end measurements of the paths between them and send the obtained measurements to a central server. It is also necessary to discover the topology of the network under analysis. Furthermore, another challenge is to collect measurement from a large enough number of different vantage points.

I. ChkDiff (2015)

ChkDiff [3], [94] is a tool for TD detection on ISPs that serve the domestic market (tier 3). The tool first captures user traffic from a normal session and then replays a version of that traffic so that it remains within the user ISP. ChkDiff measures packet loss and delays. The tool is able not only to detect TD but also to identify at which router TD occurred. The authors state that the strategy for measuring and detecting TD is independent of specific applications and the TD mechanisms employed by the ISP. Whatever the discriminated applications or techniques employed, TD typically will result in longer packet delays and more losses for the end-user.

The user traffic captured by ChkDiff is called a trace. This trace consists of a set of applications being executed by the user. The captured trace is used with minimal changes: this ensures that the traffic shapers that the trace traverses will have the same behavior as if the packets were being generated by the user running the the same applications. Only two modifications are made. The first is in the TTL field, so that packets only reach some desired hop. The second is that all packets have the same size, thus avoiding different transmission times.

ChkDiff takes its measurements by reproducing the captured trace several times, from an end-host. The TTL is progressively incremented so that at each time the trace is transmitted it reaches one more router. When a packet arrives at the final router which is reached when the TTL field gets to zero, the router sends an ICMP Time Exceeded message back to the source host. ChkDiff measures the packet delay and losses with respect these ICMP responses: the delay is the RTT measured from the time the original packet is sent to the time at which the ICMP message arrives. A packet loss corresponds to an ICMP message that is not received. Thus ChkDiff evaluates routers that are close to the user seeking for router behavior that identifies that TD has occurred. ChkDiff assumes that there is non-neutral scheduler just before that router.

ChkDiff performs a statistical analysis to infer whether the traffic has suffered TD or not. The tool compares the delay and loss rate measured for a particular router with the same metrics measured for the rest of the traffic to the same router. If measurements obtained for some particular type of traffic are significantly greater those obtained for the rest of the traffic, then ChkDiff concludes that the router has applied TD to that traffic. Thus, the baseline used by ChkDiff is the whole traffic: NR states that a non-discriminated flow is treated in the same way as all other traffic, i.e. the measurements obtained for some traffic that is discriminated will stand out in relation to the rest of the traffic. In a simplified example, if the packet loss measured for some type of traffic is around 50%, while the loss measured for the rest of the traffic is around 10%, it is possible to conclude that the ISP is employing TD.

ChkDiff works in 4 steps as shown in Figure 11. In the first step (1) real user traffic is captured resulting in a trace. In the second step (2) the trace is preprocessed, generating a set of modified traces. In the third step (3), the set of modified traces is replayed and measurements are obtained. In the fourth step (4) the statistical analysis is performed to infer whether any traffic was discriminated and to locate where it was discriminated. The four steps are described in more detail below.

In the first step, the tool captures real user traffic from an end-host during a normal session. As ChkDiff employs measures on the upstream traffic, it favors data-intensive applications such as file sharing, VoIP, and instant messaging.

In the second step, ChkDiff processes the captured trace. This preprocessor generates a set of traces that will be replayed in the next step. The trace is separated into flows, grouping packets according to 5 items: source and destination address, source and destination port, and transport protocol. All packets have the same size, in order to avoid different transmission times, which could result in false positives, as delays must be comparable. Several new traces are then generated with

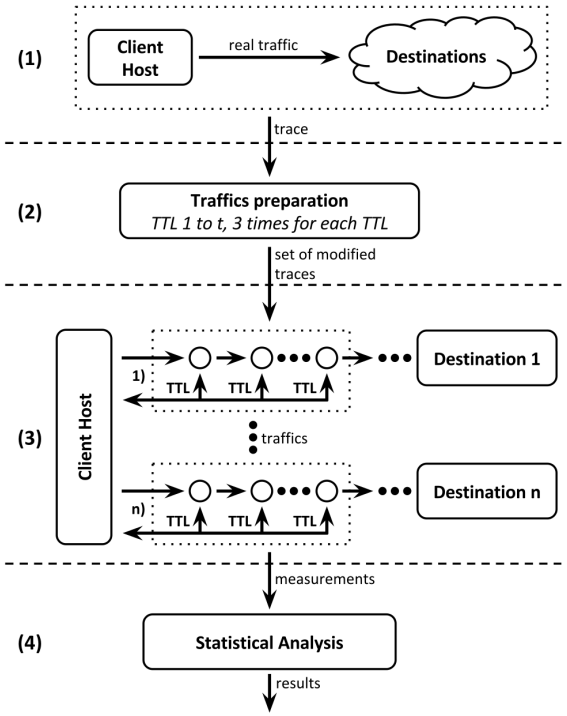


Figure 11. ChkDiff TD detection.

packets of the same size separated in multiple flows. In each of these new traces, the packets are reordered and the required value is set to the TTL field; the values range from 1 to t and 3 traces are created for each TTL value. Thus, a set of $3t$ traces is generated, containing the same packets but in different order and with different TTL values. The authors claim that using t equal to 3 or 4 is enough to traverse a typical tier-3 the ISP.

Packets are randomly reordered in each trace created in step 2, but keeping the global order of packets of a given flow. Reordering is necessary to ensure that all flows are treated under the same network conditions. According to the authors, this technique is also useful for minimizing problems such as background traffic and limitations on the maximum rate of ICMP responses that routers employ. At the end of this step, we have a set of modified traces, ready to be reproduced.

In the third step, measurements are taken as traces are sent. Let h be the TTL of the packets of one of these traces. Each packet is sent to the original destination address and port. When one of the packets reaches the h_{th} hop the corresponding router sends back an ICMP message to the source host. As mentioned above delays and losses are measured based on these ICMP messages.

The fourth step consists of the statistical analysis to infer if there was discrimination of some flow and to identify where discrimination took place. ChkDiff only uses flows for which at least 20 received ICMP responses. As described previously, 3 traces are generated for each TTL value. As confirmed in the experiments briefly described below, the authors conclude that using 3 traces helps decrease the number of false positives. The idea is that if a given flow fails in the statistical test three times for the 3 traces, ChkDiff concludes that the flow suffered discrimination. For delay metric, ChkDiff compares the delay distribution of each flow with the delay distribution of the rest of the trace. ChkDiff checks the delay distributions with the Kolmogorov-Smirnov test. In a neutral network, this test is expected to indicate that the two distributions are equal. Thus, if a flow suffered delays greater than the rest of the trace, the test for this flow has failed. With regard to the packet losses, ChkDiff checks whether the packet loss for each flow is significantly different from the packet loss of the rest of the trace. ChkDiff employs a probabilistic test inspired by a binomial distribution. If a flow presented losses greater than expected, the probabilistic test for this flow fails and the hypothesis is false. When TD is detected at some hop h, it is observable for all hops after h; ChkDiff assumes that the shaper is placed between hops $h-1$ and h.

ChkDiff was first evaluated running in a neutral environment, with no TD, and later in a non-neutral environment. In both cases user traffic was captured on 3-minute sessions. During this period, three types of applications were executed: images were uploaded in a social network; Web browsing on news sites, and messages were sent with chat applications.

In the first set of experiments, executed on the neutral environment, ChkDiff was executed 100 times in a network in which the the second hop router did not discriminate any traffic; however when a single trace was sent for each TTL value , about 30% of the executions presented 1 to 3 false positives. When experiment was executed with two traces for each TTL value: there was no false-positive. Based on these results, the authors fixed at 3 the number of traces to be generated for each TTL value as mentioned above.

The second set of experiments was executed on a non-neutral environment, and initially only one type of flow was to be discriminated. Subsequently multiple discriminated flows were used, with different fractions of the trace containing different discriminated flows. The source host was connected to a middlebox implementing a traffic shaper with the Dummynet [95] tool. The middlebox which was then connected to a router in which the TTL of the packets expired. TD was implemented in two different ways: by limiting the bandwidth of selected flows and by discarding packets from the selected flows.

In the experiments with only one discriminated flow, ChkDiff was able to detect 100% of flows which suffered bandwidth limitation. When TD was based on packet dropping, some false negatives were observed (discrimination occurred but was not detected). In the experiments with multiple discriminated flows, ChkDiff's statistical analysis stopped working correctly when the fraction of discriminated flows increased to 80% or more. ChkDiff was also evaluated and presented good results when the rate of ICMP responses from the router was limited.

J. VPN (2015)

A solution for detecting TD in mobile networks is presented in [39]. The goal is to measure whether an arbitrary application from end-user devices such as smartphones and tablets are suffering TD. The main idea is to first capture the application traffic and then replay it twice: once using a VPN (encrypted tunnel), and once using a conventional non-encrypted channel. A statistical analysis is then performed on the measurements

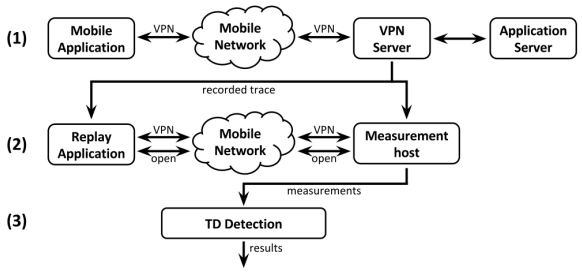


Figure 12. TD detection in mobile networks based on VPN.

obtained in order to infer if there was a TD. The metrics are the throughput, loss rate, and delay. In this work, we call this solution the "VPN solution", since it uses a VPN encrypted tunnel.

The authors assume that TD is performed by a traffic shaping middlebox which is in the network of the end-user ISP. Two commercial traffic shapers were used for validating the solution. The VPN solution does not consider as TD when the rate enforced by these devices is equal to or higher than the rate at which the traffic is generated. The authors also assume that traffic classification can be based on header, payload or traffic behavior. Furthermore, as mentioned above, the solution employs a VPN to reproduce the previously captured trace. Thus, it also assumes that VPN traffic is not discriminated.

Figure 12 illustrates the three steps of the VPN solution. In the first step (1), a VPN server captures real traffic of a mobile application while it communicates with the application server through the VPN encrypted tunnel. The captured trace is then replayed twice in the second step (2), this time the trace is sent from a replay application to a measurement host both using a VPN (encrypted IPSec tunnel) and using a conventional non-encrypted channel. The measurement host obtains information about the throughput, loss rate and delay from the trace replays. TD detection is performed in the third step (3), based on the collected measurements. The solution employs a statistical test based on KS in order to compare the different distributions and infer the presence of TD.

According to the authors, by using a VPN for traffic recording, they were able to design a mobile application capable of recording the traffic from any other mobile application without the need for special permissions or modifications in the operating system, since traffic is captured by the VPN server that intermediates the communication between the end-user device and the application server. However, there are some potential limitations to the proposed solution. Detecting TD only when the shaping rate is lower than the sending rate of the application may lead to false-negatives, specially considering that TD may only take place under congestion, which is not induced by the solution. Moreover, the solution was designed and validated assuming that TD is implemented by ISPs using traffic shaping middleboxes, which may also lead to false-negatives, since there is several ways to implement TD. Furthermore, cross-traffic may impact both recording and replaying, and thus should be taken into account. The detection may be also hindered if VPN traffic is discriminated by the ISP.

There are other solutions for monitoring NN in mobile networks. BonaFide [91] is an adaptation of Glasnost [38], described previously in this section, focused on mobile networks. BonaFide is an Android application that detects TD in a mobile network in the same way as Glasnost does for the traditional Internet. BonaFide can be seen a tailored version of Glasnost that complies to the restrictions of mobile devices.

WindRider [96] is a mobile application for detecting NN violations in a mobile network. It performs active and passive measurements. Active measurements are made using the MLab (Measurement Lab) [59] platform. Several applications using different ports are generated between a mobile device and a MLab server, in order to check if any portion of the traffic is being treated differently. Passive measurements are made directly on the mobile device. The application collects the delays experienced by different webpages, and the explicit feedback from the end-users regarding the different applications.

Furthermore, the authors of [97] advocate the creation of a "citizen observatory" of NN for mobile networks, by employing crowdsensing-based measurements and the open data paradigm. The authors claim that using a crowdsensing approach for making measurements on a mobile network can take advantage of the increasing number of smartphones, tablets and other mobile devices. Furthermore, making all measurements publicly available (an open data approach) would allow the creation of a "citizen observatory", containing NN-related information regarding different ISPs, thus increasing the transparency of mobile networks.

K. Other related works

This subsection describes other works that are related to NN monitoring, but not necessarily to the detection of TD. For instance, some of these works address other practices that may also be considered as violations of NN, or are used just to measure network performance, not to infer the presence of TD. We compiled these tools in four categories, described below: QoS under-provision, censorship, content modification, and network performance measurement platforms and techniques.

1) QoS under-provision: Laws and regulations of several countries state that a violation of NN occurs when the Quality-of-Service provided by an ISP is lower than that contracted by the user. In this way, ISPs must deliver exactly the Quality of Service (QoS) established in the contract. There are several solutions for monitoring the delivered QoS given the corresponding Service-Level Agreements (SLA) [55]–[57], [98]–[112]. Some of these solutions, namely HAKOMetar and Adkintun described below, were developed due to the interest of governments in ensuring the compliance of networks with NN-related regulations.

HAKOMetar [57] is a tool that allows an end-user to check the QoS delivered by his/her ISP. The tool was developed by HAKOM, the regulatory agency of telecommunications in Croatia. The goal of the agency is to employ HAKOMetar to increase the transparency and competition in the broadband market. The tool was created based on previous results regarding traffic management practices, obtained from experiments conducted in Croatia [113]. The tool relies on

active measurements of bandwidth, between an end-host and measurement hosts, to infer if the end-user is receiving the same bandwidth as announced by the ISP. According to the authors, the results confirm that HAKOMetar effectively increased the transparency in the Croatian broadband market, since consumers were able to check if their ISPs were really delivering the contracted bandwidth.

Adkintun [55], [56] is a solution for monitoring the QoS offered by ISPs in Chile. The solution was developed by NIC Chile Research Labs by request of SUBTEL, the regulatory agency for telecommunications in that country, with the purpose to monitor the compliance of ISPs with the Chilean NN law. Adkintun may be installed in end-users' devices or embedded in residential routers, provided to selected consumers. The tool periodically performs active measurements between end-hosts and several measurement hosts distributed over the country. Several metrics are employed, such as throughput, delay, and loss rate. All results obtained by Adkintun are publicly available through a website. The authors claim that Adkintun is helping consumers to protect their rights; the tool has been used as basis for complaints and even legal processes involving ISPs and SUBTEL. A similar tool, Adkintun Mobile [114], [115], was also developed to monitor the QoS of the mobile networks in Chile. This tool employs a combination of passive and active measurements obtained from mobile devices.

2) Censorship: The freedom of choice of end-users regarding the content they wish to access is also part of the worldwide NN debate. There are several solutions with the purpose to detect censorship in the Internet [53], [61], [116]–[118]. These solutions periodically perform measurements, creating a "census" of topics, services and websites that are blocked and/or filtered. A comprehensive survey on censorship detection in the Internet has recently been published [119].

3) Content modification: The modification of content generated either by users or providers can be employed to discriminate against unwanted traffic or to obtain advantages. Examples include: modifying the content of a website (such as inserting advertisements); injecting forged packets into the communication of end-hosts; and modifying the content of packets (for corrupting BitTorrent data, for example). There are some solutions for detecting such practices. Switzerland [120] detects the modification and injection of packets in the Internet. In [48], the authors present a solution for detecting modifications such as the injection of advertisements or malicious code in the pages of websites as they are being sent to the users.

NNSquad Network Measurement Agent (NNMA) [121] is a tool for monitoring multiple metrics related to the network activity of a set of hosts. In the context of NN, the main measurements made by NNMA refers to the injection of forged TCP reset (RST) packets. A RST packet terminates the connection between two end-hosts, thus ISPs may inject such packets in order to stop unwanted traffic [122], such as BitTorrent. While NNMA does not directly measure the impact of RST injection on different types of traffic, this technique could certainly be employed for TD detection.

4) Network measurement platforms and techniques: Network measurement platforms and services [59], [123]–[129] are used to acquire different measurements that can be used to detect TD. These solutions continuously monitor several network properties possibly from several ISPs, also allowing for a comparison of different ISPs. A complete survey on Internet measurement platforms has been recently published [130]. Furthermore, several network measurement techniques [83], [131]–[134] may also be employed for detecting TD, by comparing the measurements obtained for different types of traffic. We describe below two network measurement solutions which were designed specifically with NN issues in mind, i.e. obtaining measurements that can be employed for detecting NN violations.

Network Neutrality Bot (Neubot) [135], [136] is a software platform for continuously obtaining distributed measurements on the Internet. Neubot enables the implementation of solutions for verifying the QoS provided by ISPs based on the obtained measurements. Neubot performs several different measurements periodically on multiple end-hosts, and all data is made public. Neubot measurements include different application protocols, such as HTTP, BitTorrent, RTP, and VoIP. Neubot does not implement TD detection, since it only collects measurements. Neubot has been running in the MLab platform since February 2012, making use of the several measurement hosts provided by MLab. The authors claim that the measurements collected by Neubot allow for a systematic evaluation of the services provided by ISPs, which might contribute to the NN worldwide debate with real data.

Netalyzr [137] is a network measurement service, aimed at evaluating an end-user Internet connection, collecting data which may be further used for identifying NN violations. Netalyzr runs on end-user browsers, and makes measurements by communicating with several measurement hosts. The measurements correspond to different protocols (such as TCP, UDP, HTTP, and DNS), the end-user local network (NAT and buffers) and ISP (such as IPv6 support, content modification, port filtering, bandwidth and delay). All measurements are publicly available, contributing to a deeper understanding of QoS and NN issues.

V. TAXONOMY

Given the fact that the multiple existing solutions for TD detection have been proposed independently and often using not only different approaches and features, but also different terms for the same concepts, objectives, and techniques employed, in this section we define a taxonomy with the purpose of unifying the description of the different types of TD and TD detection under a unifying framework.

The proposed taxonomy was built taking into account the existing solutions described in section IV. The purpose is to have a common ground to understand the differences and similarities between the solutions. In section VI-B we compare the existing solutions based on the taxonomy presented in the current section.

This section is organized in two subsections: in the first we present a taxonomy of TD, in the next a taxonomy of TD

Detection. We make use of feature diagrams to present the taxonomy. Feature diagrams [138] are hierarchically arranged sets of features, with different types of relationship between features and sub-features – both optional and mandatory features.

A. Traffic Differentiation: A Taxonomy

The feature diagram in Figure 13 presents a taxonomy for traffic differentiation. In the diagram, TD has four main features, which represent different aspects of TD: triggers, traffic classification, differentiation mechanisms, and perceived discrimination. The triggers are the conditions or characteristics of the traffic that may lead an ISP to employ TD. Traffic classification indicates which features are used by an ISP to classify the traffic. The differentiation mechanisms used by an ISP to implement TD are classified according to how they affect the traffic. Finally, the perceived discrimination describes how users or monitoring systems perceive TD. We further describe each feature and its subfeatures below.

1) Trigger: An ISP may start to discriminate traffic because of specific traffic properties or under certain conditions, or even because of a combination of properties and conditions. We call triggers these factors that lead an ISP to start TD. We compiled three types of triggers:

a) *Application*: TD can be triggered by an application, which is discriminated by the ISP. For instance, an ISP may avoid congestion by slowing down bandwidth-hungry applications (e.g. P2P and video streaming), or it may prioritize traffic from its own applications or from business partners.
b) *Path*: in TD triggered by path, all the traffic coming from, or going to specific end-hosts or traversing specific ASes may be discriminated. For instance, an ISP may prioritize traffic coming from a certain content provider or a neighbor AS due to commercial agreements (e.g. fast-lanes).
c) *Network state*: this TD trigger is employed depending on the state of the network. For instance, an ISP may employ TD only on links with high load, or at specific times of a day (e.g. peak hours).

For instance, if an ISP degrade all traffic from a specific application or with a specific destination address, we say that in this case TD is triggered by application and path. The solutions presented in section IV are able to detect different combinations of these triggers. Furthermore, a solution may be agnostic to the trigger, i.e., it does not make any assumption regarding which triggers are employed, being able to detect TD regardless of the triggers.

2) Traffic Classification: Several different properties may be used to identify the triggers presented above and assess the priority of the corresponding traffic, as described previously in subsection III-C. We identified four categories of traffic classification:

a) *Header*: classification based on header information, e.g. source and destination addresses and/or ports, transport protocol used, application protocol used, type of service (TOS) required, among many others.
b) *Payload*: classification based on application data, either using information from the application PDU header (e.g. HTTP or BitTorrent headers), or based on application payload, which can be identified using DPI and pattern matching.
c) *Traffic behavior*: classification on flow rate, flow duration, average packet size, inter packet interval, number of connections, total bandwidth.
d) *Routing*: classification based on source and/or destination end-hosts or networks, previous and next ASes.

For instance, an ISP may identify from which application some traffic corresponds to by checking packet headers – e.g. destination port. Some solutions for detecting] TD, however, are agnostic to specific classification methods, i.e., they don't make assumptions regarding how ISPs classify traffic.

3) Differentiation Mechanism: There are several mechanisms that an ISP may employ to implement TD, as described previously in subsection III-C. Different mechanisms may affect the traffic in different ways. We identified four categories for these TD mechanisms:

a) *Block*: block mechanisms interrupt all traffic by simply not forwarding packets, or by injecting connection termination messages (e.g. messages with the FIN or RST flags set in the TCP protocol).
b) *Delay*: delay mechanisms either increase or decrease the delay of packets. These mechanisms may, for instance, prioritize packets according to their type (e.g. traffic shaping) and/or forward packets through internal routes that are faster or slower.
c) *Drop*: drop mechanisms degrade the traffic by dropping packets according to some criteria (e.g. traffic policing).
d) *Modify*: modification mechanisms alter packets, header and/or payload. For instance, an ISP may reduce the TCP window size to force the sender to slow down, or even modify specific application protocol fields to manipulate the application behavior (such as in transparent proxies).

For instance, a traffic shaper may be employed by applying some non-neutral scheduler to the traffic, forwarding different types of traffic according to priorities. This is an example of a delay mechanism, since its main goal is to delay low-priority traffic, prioritizing other traffic. Some solutions, however, make no assumptions regarding which TD mechanisms are employed.

4) Perceived Discrimination: TD is perceived by users and monitoring systems in several different ways. This is an important aspect of NN, as can be confirmed in the case studies reported in subsection II-B. These features reflect how users perceive and report TD, and are often the basis of proposed regulations and compliance surveillance.

a) *Longer delays*: users perceive longer delays to receive data from the network.
b) *Increased jitter*: the variation of the delay is high enough to disrupt specific applications.
c) *Throttling*: monitoring systems can perceive TD as a reduction of the available bandwidth, however this is often perceived by users as longer delays or unrespon-

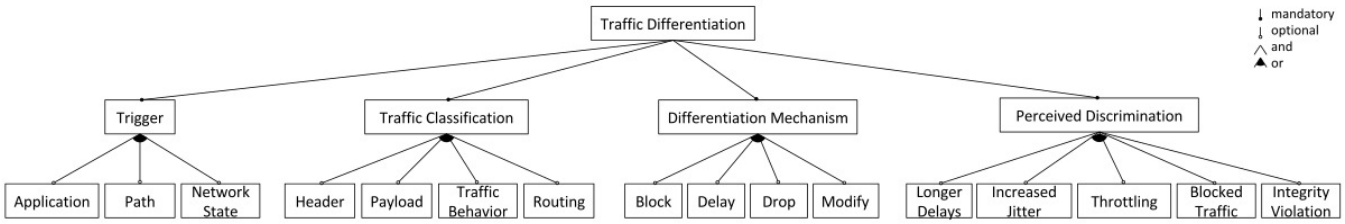


Figure 13. TD feature diagram.

sive services, which are common in the case of video streaming applications.

d) *Blocked traffic*: users do not receive any or part of the packets of some particular application.

e) *Integrity violation*: the received information has been modified in the network in an unauthorized way.

For instance, some tools measure TD from the point of view of end-users, reporting delays or jitter that are higher than expected, besides throttling (bandwidth reduction), non-authorized modifications, or even blocked traffic which is some cases have been reported to be nation-wide.

B. TD Detection: A Taxonomy

The feature diagram in Figure 14 presents a taxonomy for TD detection. We identified four main features: measurements, monitoring architecture, traffic, and inference mechanism. These features represent different aspects of strategies for detecting TD. In order to detect TD, *measurements* are made, performed by hosts organized in different *topologies*. The *traffic* employed in such measurements may also have different properties. The data obtained may be processed in different ways to *infer* the presence of TD. We further describe these features below.

Note that this taxonomy does not represent the way any particular solution was designed, nor how new solutions should be designed. It is meant to organize concepts and features to allow comparisons. We hope though that the taxonomy can be an useful framework to help creating new solutions.

1) Measurements: Since the internal properties of ASes are not known *a priori*, TD detection solutions rely on measurements that are made outside the network in order to infer what happens inside. These measurements are thus made from end-hosts and, in the context of the TD detection problem, have three fundamental characteristics:

a) *Metrics*: there are several possible metrics that may be employed to assess different types of traffic. Different TD mechanisms affect traffic in different ways, thus different metrics may also be employed. For instance, traffic policing may have a larger impact on the loss rate than on delay, since it favors dropping instead of queueing packets in order to enforce a maximum rate. Traffic shaping would have the opposite impact. Throughput may be equally affected by shaping and policing, since both dropping and delaying will cause less packets to be transferred during a given interval of time. The most common metrics employed by the existing solutions presented in section IV are delay, loss rate and throughput, but other metrics are also used.

b) *Target*: measurements might be taken at *end-hosts*, not taking into consideration routers in the path; alternatively measurements can be taken relative to some router. For instance, a solution may measure the upload bandwidth of some end-host, and the download bandwidth at the other end, ignoring how the routers in the path interfered in the measurements. Moreover, a solution may measure the loss rate experienced by some traffic when traversing a specific router in an ISP network.

c) *Strategy*: measurements might be made following different approaches, such as active, passive, or hybrid. These different approaches are further described below in subsection VI-A.

2) Monitoring Architecture: Measurements might be made using different topologies, and may require control of some, or all, of the hosts participating in the measurements. We defined three different aspects of a monitoring architecture:

a) *Controlled Infrastructure*: a monitoring solution may require *control* of one or several hosts in order to make measurements. It may also take advantage of a pre-existent infrastructure to make the measurements. For instance, a solution may explicitly run a measurement application on two end-hosts, which would require the installation of the application in the hosts, or it could make measurements based on some already existing activity, for example in a P2P network, which would not require any installation or modification of the end-hosts to be done.

b) *Vantage points*: measurements may be taken from different "points of view". For instance, a solution may require only measurements between a pair of end-hosts in order to infer TD, or it may require measurements from multiple pairs of end-hosts, aggregating and processing the collected data afterwards.

3) Traffic: Different types of traffic may be employed to make measurements and infer TD. Most existing solutions, described in section IV, employ two or more types of traffic in their measurements and check if there was any significant difference in the performance measured for each type. Traffic can be of two types:

a) *Real*: a TD detection solution may take advantage of already existing traffic. For instance, passive measurements only observe traffic, making measurements without introducing any new traffic in the network. Furthermore, a solution may record some real traffic and use it later

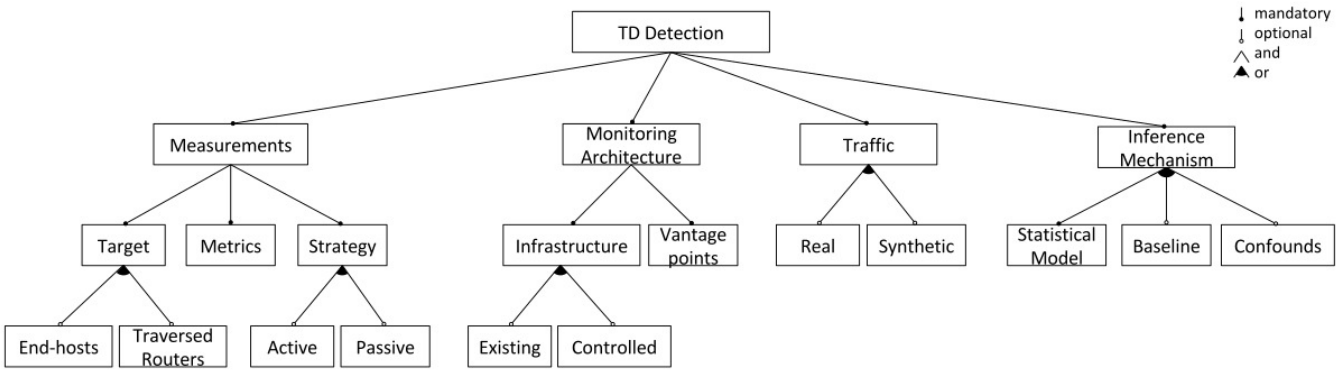


Figure 14. TD detection feature diagram.

to perform measurements, instead of generating synthetic traffic.

b) *Synthetic*: a solution may generate traffic to execute measurements. For instance, some solutions generate different types of traffic by modifying some previously recorded real traffic, while others create traffic only by following the specifications of an application protocol.

4) Inference Mechanism: Different approaches are possible in order to detect TD from the obtained measurements. We identified three main features related to TD inference:

a) *Statistical model*: different statistical methods can be employed to analyze the collected data and make inferences regarding TD. Hypothesis tests may be employed to compare different sets of measurements.

b) *Baseline*: most solutions compare the measurements regarding different types of traffic with some baseline traffic, which is often assumed to be non-discriminated. This baseline for comparison may be obtained in several different ways. For instance, some solutions assume that a specific type of traffic of some application (e.g. HTTP) doesn't suffer TD, and thus can check how the traffic generated by other applications compare with the baseline.

c) *Confounds*: confounds are the other factors that may have an impact on the traffic. Different solutions consider different confounds, and have different approaches to deal with them. For instance, some solutions repeat the same measurements several times in order to decrease noise (e.g. from cross-traffic).

VI. Consolidation of the State of the Art

The purpose of this section is to consolidate the state of the art by presenting (i) a compilation of the techniques employed by the TD detection solutions described in Section IV; (ii) a comparison of the solutions; and (iii) a discussion of TD detection challenges given the state of the art.

A. Techniques Employed by the TD Detection Solutions

In this subsection we present a compilation of the different techniques employed by the solutions described in Section IV to detect TD.

1) Passive/Active and Hybrid Measurements: As described in section III, different TD mechanisms affect traffic in different ways, such as increasing the delay or loss rate. Thus, existing solutions rely on network measurements to identify performance conditions that represent symptoms that part of the traffic is being treated differently from other parts. Network measurements may be active, passive, or hybrid.

Passive measurements consist of observing the real network traffic, without generating any new flows or packets. Measurements are usually made at both ends by evaluating performance characteristics for sending and receiving packets. For instance, a sniffer implements passive measurements, which captures and analyses all network traffic.

Active measurements generate traffic, i.e. probes or flows between one or more pairs of end-hosts. Measurements are then made to evaluate the performance of the probes, for example. For instance, the source host sends a file to an destination host, using the FTP protocol, then the performance is evaluated through metrics such as throughput, packets loss rate and delay, among others.

Hybrid measurements consist of any combination of active and passive approaches.

2) TTL-based Probes: Some solutions employ a technique based on the ICMP protocol in order to locate where TD occurred. This technique consists of sending measurement probes with a predefined TTL (Time To Live) value. The idea is to force the probe to reach up to a specific router between the source and the destination.

In this technique, the TTL field of each probe is set to some value i, and is sent from one end-host to another. Unless it reaches its destination, the probe travels only up to the i-th router. Each router along the path between the end-hosts decreases the TTL field by one. When the value is 0, the probe is dropped, i.e. it is not forwarded to the next router. Furthermore, an ICMP *Time exceeded* message is sent back to the source host. The measurements are then made based on the ICMP responses. For instance, the loss rate is the ratio of ICMP responses not received, and the delay is the time interval from the instant the probe is sent to the time instant the ICMP response is received. This technique can thus be used to make measurements for each router along the path between a pair of end-hosts, by sending multiple probes with incremental TTL

values.

3) Path Saturation: Depending on the traffic management policies adopted by an ISP, TD may be employed only when the network is congested. For instance, if an ISP employs traffic shaping for queuing and forwarding packets according to different priorities, different delays and/or loss rates will only be observed if the packets are effectively being queued. If the routers are able to forward packets fast enough, no queuing will effectively happen. Thus no packet will be delayed in favor of others, and since queues do not fill up, no packets will be selectively dropped.

Therefore, TD may only be observable when the network is congested. Thus, active measurements often generate a large amount of traffic to saturate the path between two end-hosts, forcing TD to happen, i.e. forcing packets to be delayed and/or dropped, or bucket tokens to expire, for example.

4) Client-Server Measurements: Several solutions make measurements based on traffic generated between two end-hosts, following a client-server model. These end-hosts are often called the *client host* and a server which is called the *measurement host*. On the one hand, this approach allows for several different types of measurements, made both on the client host and on the measurement host, as well as total control of the traffic – such as what is sent, which protocols are employed, which responses are expected, etc. On the other hand, a problem with this technique is that it is not possible to evaluate other paths other than the path between the two end-hosts, neither does it allow the detection of TD triggered by path (such as specific origins or destinations).

5) Measurements Involving Multiple Hosts: Some solutions make use of multiple hosts instead of a client-server pair. These solutions may make measurements from several hosts, and/or to multiple hosts or prefixes. The idea is usually to acquire data from multiple vantage points, and confront them in order to infer some property – which may indicate TD and where it took place.

For instance, some solutions control a large number of hosts, and generate traffic between them. Others send measurement probes from one or more hosts to multiple internet destinations prefixes, in order to make the traffic traverse different paths, through different ISPs.

Another related technique is network tomography [93]. It consists of inferring properties of network internal links (such as delay or loss rate) only using end-to-end measurements, i.e. measurements for particular internal links are not available – in the TTL-based probes these such measurements are available. Network tomography usually combines several end-to-end measurements, from different vantage points, to infer properties of a common core.

6) Traffic Recording: In order to make active measurements using real traffic, some solutions record the traffic in advance and replay it afterwards, as many times as needed. Traffic recording can be done for instance by capturing the traffic of some applications as it is being executed by some user in a real network [139]. Some solutions reproduce recorded traffic exactly as it was captured, while others make modifications before reproducing.

This technique allows the use of traffic from any arbitrary application or protocol, whether standard or not. However, traffic recording often requires special permissions.

7) Traffic Emulation: This technique consists in generating artificial traffic that mimics an application or protocol. In this way, measurements can be made for any type of traffic, varying features such as port number, application protocol, payload, sending rate, packet size, among others. Several existing solutions also employ this technique to create a baseline traffic, which is assumed to be non-discriminated – e.g. carrying randomized payload.

Traffic emulation may be performed based on previously captured real traffic, or can be artificially generated [140]. The latter requires the protocol and application behavior to be well known, and is also usually based on statistical models. Artificial traffic, however, may not be realistic enough, being treated by ISPs differently than the real traffic would be [39].

8) Traffic Shuffling: In the traffic shuffling technique, different flows are sent simultaneously, with their packets interleaved in random order. Some solutions shuffle the packets each time they are sent, in order to decrease the bias. For instance, if packets are sent always in the same order, they might get queued in the same way every time, and this may be misinterpreted as TD: buffers gets full and most dropped packets are of a single application, which may lead to the conclusion that the traffic is being discriminated, i.e. a false positive. This can be avoided by sending the traffic multiple times, with packets in random order. Note, however, that the relative order of packets of an application should be kept the same, since changing their order would change the behavior of the application, which might affect the traffic classification.

9) Relative Discrimination: A common approach for inferring TD is to compare measurements taken for different types of traffic, in order to determine if they have been treated differently, i.e. if one traffic was prioritized or degraded in relation to the other. The rationale is that if no TD was employed, the distributions of measurements for two different types of traffic will be statistically similar, since both traffics experienced the same network conditions. However, if one type is treated differently, the corresponding measured distributions will be significantly different: one type of traffic was discriminated relative to the other. Most solutions that employ this technique assume the existence of a baseline traffic that is non-discriminated traffic. The strategy then is to compare other types of traffic with the baseline traffic.

Hypothesis testing is frequently used in order to infer relative discrimination. For instance, the hypothesis may be that the two sets of measurements are drawn from the same distribution. If the test fails, then the hypothesis is false; in this case the measurements are significantly different, and thus TD is characterized. Examples of tests employed by the existing solutions include the Kolmogorov-Smirnov (KS) test [89], Kullback-Leibler test (KL), two-proportion z-test, and Mann-Whitney U-test [92].

10) Measurements Clustering: Some solutions try to do clustering on the obtained measurements in order to compare several different sets of measurements, instead of just two as in the relative discrimination technique described above.

Measurements for different types of traffic, from different sources or even from different ISPs may be grouped together and compared afterwards.

Measurements may be clustered, for example, according to the corresponding confounds – such as grouping measurements from hosts that geographically close to each other. Another example is clustering different types of traffic according to the measured performance, building a ranking based on performance classes – which should result in a single class in a neutral network.

B. Comparison of the TD Detection Solutions

The solutions described in section IV differ mainly on how they make measurements and compare the obtained data, presenting different monitoring architectures, metrics, employing different traffic generation strategies, and statistical methods. Most solutions perform active measurements between one or more pairs of hosts – employing traffic corresponding to different applications – and compare the obtained measurements in order to detect significant variations. Other solutions perform measurements on routers along the path between one or more pairs of hosts, in order to identify exactly where TD happened. There are also solutions that employ passive measurements of the real traffic generated by different applications. Table I describes how each solution addresses each different aspect of TD detection defined previously in the taxonomy presented in Figure 14.

Each solution is designed with different goals and assumptions in mind. Glasnost, for example, targets end-users, being an easy-to-use online tool that requires no technical knowledge, while NetPolice targets backbone ISPs, and requires more technical knowledge to be deployed and run than Glasnost. The solutions presented in this section employ different sets of techniques to achieve their goals under the assumptions made. Table II shows which of the techniques described previously in subsection VI-A each solution employs. Note that some solutions are based on similar sets of techniques. However, even when they are similar, the same techniques may be implemented in different ways, achieving different results.

In Table III, we map our TD taxonomy, as defined in Figure 13, to the existing solutions. The table shows which types of TD each solution is capable of detecting. The cells in black indicate that the solution is agnostic to a specific feature, i.e., it does not make any assumptions regarding that feature[3].

C. A Discussion on TD Detection Challenges

We discuss next some of the main challenges for detecting traffic differentiation. The solutions described in section IV address some of these challenges. We also identify open challenges and future work in subsection VII.

1) Challenges of End-to-end Measurements: Internal properties of the ISP networks are not known *a priori*. The topology, scheduling algorithms employed, specific devices in the network and how they are configured are examples of information that is not publicly available. Therefore, that information cannot be used to check if a network is neutral. Furthermore, since it is not feasible to test all possible types of traffic, at most what can be done is to run some tests in order to try to find cases in which TD can be identified. TD detection solutions rely on end-to-end measurements, from which they infer whether TD is being used or not.

Some solutions, however, make assumptions regarding specific characteristics of the network, such as the presence of traffic shapers, or support for specific protocols. For instance, some measurement strategies rely on the ICMP protocol, which is not universally supported by routers on the Internet [3]. However, some routers limit the rate of ICMP responses.

2) Confounds: Several other factors besides TD may result in observable differences on the for different types of traffic – the so-called confounds [87]. Examples include different routes, cross-traffic, congestion, geographic location, time of day, software, hardware, and other characteristics of the network (e.g. signal quality in mobile networks).

Measurements obtained in different periods of the day should not be compared, since the performance of applications may vary depending on the time they are executed. Furthermore, comparing traffic between different pairs of end-hosts is not always possible, since different hosts and the routes between them may have completely different characteristics. An ISP may also employ routes with different characteristics for different types of traffic due to reasons other than TD, such as load balancing or peering agreements. Congestion may not affect different types of traffic in the same way, as it depends on characteristics such as packet sizes, protocols employed, frequency of communication, among others.

Therefore, robust statistical models are necessary for obtaining reliable results [32], avoiding false-negatives and false-positives. There is no automated way for enumerating all confounds or to check if a set of confounds is enough. Depending on the approach adopted for detecting TD, it may be necessary to identify the confounds and collect data about them in addition to measuring the performance of applications.

3) Cross-traffic: Among the confounds described above, one of the most relevant which can have a deep impact on TD detection is cross-traffic. Traffic generated by other sources other than the TD detection solution may impact the measurements made, and thus how precisely TD can be inferred. Cross-traffic may be present in the same host and/or in the same local network. A large amount of cross-traffic may affect different types of traffic in different ways, this is particularly complicated when the cross-traffic presents a large variation over time.

4) TD Location: In addition to detecting the presence of TD, another challenge addressed by some solutions is to locate where TD occurred. Determining that TD is taking place at a particular point of a network is not a trivial task, since there is no prior knowledge of so much information about the network internals. In order to address this challenge, some solutions employ measurement techniques on an internal path, while others combine measurements from multiple points of view.

5) TD Detection Evaluation: Validating a new solution in the wild may not be feasible, since there is no knowledge

[3]The authors of Packsen did not specify which types of traffic classification their solution assumes, neither that no assumption is made.

Table I
HOW EACH SOLUTION ADDRESSES EACH ASPECT OF TD DETECTION

		Gnutella RSP	NetPolice	NANO	POPI	DiffProbe	Glasnost	Packsen	Tomography	ChkDiff	VPN
Measurements	**Metrics**	Connectivity	Loss rate	Depends on application	Loss rate	Delay and loss rate	Throughput	Inter-arrival times, sent and received bandwidth	Any additive metric	Delay and loss rate	Throughput, loss rate, and delay
	Target: end-hosts	Measurement host	–	Multiple end-hosts	Pair of end-hosts	Pair of end-hosts	Pair of end-hosts	Pair of end-hosts	Multiple end-hosts	–	Pair of end-hosts
	Target: traversed routers	–	Ingress and egress points	–	–	–	–	–	–	First few hops	–
	Strategy	Hybrid	Active	Passive	Active	Active	Active	Active	Active	Active	Active
Monitoring Architecture	**Vantage points**	Multiple end-hosts and a measurement host	Multiple end-hosts	Multiple end-hosts	A pair of end-hosts	A pair of end-hosts	Multiple end-hosts and measurement hosts	Multiple end-hosts and measurement hosts	Multiple end-hosts	From a single end-host	A pair of end-hosts
	Existing Infrastructure	Gnutella P2P network	–	–	–	–	–	–	–	–	–
	Controlled Infrastructure	A superpeer and a measurement host	Multiple end-hosts in different networks	Multiple end-hosts in different networks	A pair of end-hosts	A pair of end-hosts	End-host, measurement hosts, and a web server	End-host, measurement hosts and an experiment server	Multiple end-hosts	A single end-host	An end-host, a VPN server and a measurement host
Traffic	**Real**	Gnutella protocol	–	Existing real traffic on end-hosts	–	–	–	–	–	Records the end-user real traffic	Records an application real traffic
	Synthetic	Referrals	HTTP, BitTorrent, SMTP, PPLive and VoIP	–	Several bursts containing multiple types of traffic	Based on applications real traffic	Based on real traffic	Not specified	Not specified	Reproduces traffic with modifications	Reproduces traffic through different channels
Inference Mechanism	**Statistical model**	Probabilistic model to infer if ports were blocked	Compare the performance of different applications with the performance of the baseline	Group measurements with similar confounds and compare them with the baseline	Rank the measurements for each burst and verify if any given type of traffic was consistently ranked higher than others	Compares the performance of an application with the performance of the baseline	Compares the performance of an application with the performance of the baseline	Compares the performance of an application with the performance of the baseline	Employs network tomography to combine measurements from different vantage points and find non-neutral links	Compares each type of traffic reproduced with all the other types	Compares the recorded traffic performance when encrypted (VPN) and non-encrypted (conventional open communication)
	Baseline	–	Assumes that HTTP traffic doesn't suffer TD	The baseline is the average performance across several ISPs	–	The baseline is generated based on the target application and is assumed to be non-discriminated	The baseline is generated based on the target application and is assumed to be non-discriminated	The baseline is assumed to be non-discriminated	–	The baseline is the whole traffic, expect for the type being evaluated	The baseline is the encrypted traffic, reproduced in a VPN, which is assumed to be non-discriminated
	Confounds	Gnutella clients might ignore the referrals	Compare only measurements relative to a same core (AS)	Require the identification of all relevant confounds (related to the hosts, network and time)	Sends multiple bursts of packets in random order	Both traffics have same properties, such as packets size and sending intervals	Employ different measurement hosts to avoid evasive measures from ISPs	Repeats the measurements several times; keeps a constant bandwidth for both traffics	–	The size of all packets is standardized	–

Table II
TECHNIQUES EMPLOYED BY EACH SOLUTION

	Gnutella RSP	NetPolice	NANO	POPI	DiffProbe	Glasnost	Packsen	Tomography	ChkDiff	VPN
Active measurement	X	X		X	X	X	X	X	X	X
Passive measurement	X		X							
TTL-based probes		X							X	
Path saturation				X	X					
Client-server	X			X	X	X	X		X	X
Multiple hosts		X	X					X		
Traffic recording					X	X			X	X
Traffic emulation		X		X	X	X	X	X		
Traffic shuffling				X					X	
Relative discrimination		X			X	X	X		X	X
Measurements clustering			X	X				X		

about the internals of the network between the end-hosts. It is important to define methods that avoid biased results. Most existing solutions rely on simulation and emulation in order to validate their strategies.

Simulated or emulated environments, however, do not have the same conditions found in a real environment. ISPs may classify and/or differentiate traffic in several different ways that may not have been covered by the simulation/emulation. Artificial traffic generated by the solutions may also be treated differently than real traffic would be. Solutions may experience thus more false-negatives and false-positives than expected when deployed on the wild.

VII. OPEN CHALLENGES

In section VI, we presented a discussion of TD detection challenges, as well as an overview of existing techniques and solutions, and how they address such challenges. However,

Table III
TYPES OF TD DETECTED BY EACH SOLUTION

		Gnutella RSP	NetPolice	NANO	POPI	DiffProbe	Glasnost	Packsen	Tomography	ChkDiff	VPN
Trigger	Application	X	X	X	X	X	X	X		X	X
	Path		X								
	Network State										
Classification	Header	X	X		X	X	X	?		X	X
	Payload		X		X	X	X			X	X
	Traffic Behavior										X
	Routing		X							X	
Mechanism	Block	X									
	Delay				X	X	X	X			X
	Drop				X	X	X	X			X
	Modify										
Perceived	Longer Delays					X		X		X	X
	Increased Jitter					X		X		X	X
	Throttling		X		X	X	X	X		X	X
	Blocked Traffic	X									
	Integrity Violation										

we envision several other challenges which still need to be further investigated for designing effective solutions that can be considered to be future-proof. We list below some of the open challenges we identified. In [141] the authors propose a model to address several of these challenges on future distributed systems.

A. Measurements and Monitoring

Measurement techniques employed by the existing solutions present limitations. Active measurement strategies often require the path to be saturated first, resulting in high network overhead that may not represent the real conditions in which most applications run. Moreover, some techniques rely on TTL-based probes (which are not universally supported), or prior knowledge of the network topology to infer which ISP is employing TD. Furthermore, some existing solutions require control of a large number of end-hosts to monitor the network, which might not be a realistic assumption.

Another limitation refers to traffic recording techniques, by which previously captured traffic is replayed between two end-hosts. Some applications generate traffic between several pairs of nodes, and not just a single pair. Therefore, it might not be possible to properly mimic every application application by reproducing its traffic only between two end-hosts.

Further investigation on traffic monitoring and measurement techniques, including the metrics used, are also necessary to detect TD triggered by the network state. Another related challenge is the detection of dynamic traffic behavior, such as occurs for example when an ISP employs TD just on specific periods of the day, or when the ISP constantly changes the TD mechanisms over time. Moreover, most current solutions do not address traffic classification based on traffic behavior, or TD mechanisms based on traffic modification.

B. Mobile Networks

There are few solutions specializing on detecting TD in mobile networks. As different confounds and constraints apply to mobile environments, different techniques might be necessary. For instance, measurements in mobile networks may be affected by mobility itself or fluctuations in channel quality, furthermore path saturation is usually not feasible, since mobile devices are usually subject to data caps.

C. ISP Evasion

Most existing solutions generate their own traffic in order to make measurements and infer TD. However, the artificial traffic generated by such solutions might be identified by ISPs [142], which could then evade the TD inference, by prioritizing the measurement traffic, for example.

D. Solution Adoption

In order to achieve meaningful results, some solutions require that a large number of end-users report measurements for several different applications, and from multiple vantage points. Therefore, it is important to create incentives which may increase the adoption of the solution by a large number of users. Another challenge is to allow any arbitrary application to monitor how its traffic is performing compared to others, without having to implement TD detection on its own. This would enable not only end-users, but also applications and services to benefit from TD inference and to contribute to increase its accuracy. Taking advantage of pre-existent infrastructures and/or real traffic monitored passively also allows measurements to be made without the need to control a large number of end-hosts or rely on a large number of end-users.

E. Network Programmability

The infrastructure of the Internet and TD mechanisms employed by ISPs are in constant evolution. As the authors of [39] describe, some commercial traffic shapers are able to identify a large number of applications, and classify the traffic based on payload and port. However, emerging technologies that allow

network programmability, such as SDN [84], will increase the flexibility and enable different types of discriminatory behavior to be easily implemented in the network, thus the techniques employed will not be limited anymore by only what commercial shapers do. We can see that in the future it will not be possible to make assumptions on how TD is implemented. The design of TD detection solutions will have to make room for extensions on the fly, enabling them to keep up with network dynamics and evolution.

F. ISPs and Content Providers are Becoming Indistinguishable

A trend that is easy to see is the fact that commercial agreements between content providers and ISPs are becoming increasingly common and varied. Actually several ISPs are becoming content providers, while content providers are becoming ISPs. Examples of commercial agreements include the usage of Content Delivery Networks (CDN) and the adoption of zero-rating practices. All these factors can result in content prioritization and discrimination and without a doubt pose new challenges on both the ability to detect TD and even more, on the very definition of what constitutes a NN violation.

VIII. Conclusion

NN has become increasingly important worldwide. NN violation cases have been reported in the five continents, making it clear that the subject is very complex, and that monitoring and enforcing compliance with NN regulations has become a truly critical task worldwide.

As the number of Internet users reaches the 4 billion mark, and a myriad of Internet services become available, several governments are creating NN regulations. In several countries TD is illegal. However, regulations are not enough to ensure ISP compliance. Solutions to monitor and enforce NN compliance are necessary. We argue that even without regulations, maintaining the transparency on traffic management practices on the Internet is important by itself and can lead to a more competitive market and foster innovation. Detecting TD on the Internet, however, is still a challenge.

In this work we presented the problem of detecting TD on the Internet, and surveyed existing solutions, highlighting the techniques employed by each one, how they implement these techniques, and what types of TD each solution is capable of detecting. A taxonomy was proposed for the different aspects of TD and its detection. We also identified the main challenges for detecting TD, as well as common techniques employed by existing solutions to solve the problem. We also identified open challenges that should be addressed by future work.

Besides providing a wide view of techniques and existing solutions for the problem of detecting TD, we hope to provide in this work a common basis for future research, hoping that the survey will foster new efforts towards more capable and future-proof solutions.

Future work includes dealing with a myriad of open challenges in TD detection in the Internet. Specific scenarios, such as Smart Cities and the Internet of Things, may provide a good environment for gathering NN-related measurements. Building effective and easy-to-use tools that are accessible even for end-users remains one of the main challenges. However several other relevant topics for research can be listed, including statistical models, streaming analytics, data mining, among other techniques tailored for TD detection. Besides detecting TD, the detection of the motives behind TD is also a relevant topic for future research; for instance an ISP may apply TD to delay investing in the evolution of its network. Note that even if regulations change the problem remains relevant, as it can be translated into keeping Internet traffic management policies transparent in the Internet.

Acknowledgments

This work was partially supported by the Brazilian National Research Council (CNPq), projects 309143/2012-8 and 306248/2016-6. Thiago Garrett has a Ph.D. scholarship from the Brazilian Education Ministry CAPES.

References

[1] K. G. Coffman and A. M. Odlyzko, *Internet Growth: Is There a "Moore's Law" for Data Traffic?* Springer US, 2002, pp. 47–93.

[2] B. van Schewick and D. Farber, "Point/Counterpoint: Network Neutrality Nuances," *Communications of the ACM*, vol. 52, no. 2, pp. 31–37, February 2009.

[3] R. Ravaioli, G. Urvoy-Keller, and C. Barakat, "Towards a General Solution for Detecting Traffic Differentiation at the Internet Access," in *International Teletraffic Congress (ITC)*, September 2015, pp. 1–9.

[4] M. L. Mueller and H. Asghari, "Deep packet inspection and bandwidth management: Battles over BitTorrent in Canada and the United States," *Telecommunications Policy*, vol. 36, no. 6, pp. 462–475, 2012.

[5] F.-Y. Ling, S.-L. Tang, M. Wu, Y.-X. Li, and H.-Y. Du, "Research on the net neutrality: The case of Comcast blocking," in *International Conference on Advanced Computer Theory and Engineering (ICACTE)*, vol. 5, August 2010, pp. V5–488–V5–491.

[6] H. Habibi Gharakheili, A. Vishwanath, and V. Sivaraman, "Perspectives on Net Neutrality and Internet Fast-Lanes," *SIGCOMM Computer Communication Review*, vol. 46, no. 1, pp. 64–69, January 2016.

[7] J. Kendrick, "T-Mobile Germany Blocks iPhone Skype Over 3G and WiFi," 2009, accessed in October 16, 2017. [Online]. Available: https://gigaom.com/2009/04/06/t-mobile-germany-blocks-iphone-skype-over-3g-too

[8] N. Lomas, "Verizon Accused Of Net Neutrality Foul By Zero-Rating Its Go90 Mobile Video Service," 2016, accessed in October 16, 2017. [Online]. Available: https://techcrunch.com/2016/02/07/verizon-accused-of-net-neutrality-foul-by-zero-rating-its-go90-mobile-video-service

[9] A. Joch, "Debating net neutrality," *Communications of the ACM*, vol. 52, no. 10, pp. 14–15, October 2009.

[10] Ministry of Internal Affairs and Communications, "Report from Panel on Neutrality of Networks," 2008, accessed in October 16, 2017. [Online]. Available: http://www.soumu.go.jp/main$_s$osiki/joho$_t$susin/eng/Releases/NewsLetter/Vol18/Vol18$_2$3/Vol18$_2$3.html

[11] Norwegian Communications Authority, "Net neutrality," accessed in October 16, 2017. [Online]. Available: http://eng.nkom.no/technical/internet/net-neutrality/net-neutrality

[12] Canadian Radio-television and Telecommunications Commission, "Review of the Internet traffic management practices of Internet service providers," 2009, accessed in October 16, 2017. [Online]. Available: http://www.crtc.gc.ca/eng/archive/2009/2009-657.htm

[13] Subsecretaría de Telecomunicaciones, "Consagra el Principio de Neutralidad en la Red para los Consumidores y Usuarios de Internet," 2010, accessed in October 16, 2017. [Online]. Available: http://www.leychile.cl/Navegar?idNorma=1016570

[14] El Congreso de Colombia, "Plan Nacional de Desarrollo, 2010-2014," 2011, accessed in October 16, 2017. [Online]. Available: http://www.alcaldiabogota.gov.co/sisjur/normas/Norma1.jsp?i=43101

[15] Comisión de Regulación de Comunicaciones, "Condiciones regulatorias relativas a la neutralidad en Internet," 2011, accessed in October 16, 2017. [Online]. Available: http://www.alcaldiabogota.gov.co/sisjur/normas/Norma1.jsp?i=45061

[16] Korea Communications Commission, "Annual Report 2011," 2012, accessed in October 16, 2017. [Online]. Available: http://eng.kcc.go.kr/download.do?fileSeq=35215

[17] Presidência da República, "Lei 12965," 2014, accessed in October 16, 2017. [Online]. Available: http://www.planalto.gov.br/ccivil$_{0}$3/$_{a}$to2011-2014/2014/lei/l12965.htm

[18] ——, "Decreto 8771," 2016, accessed in October 16, 2017. [Online]. Available: http://www.planalto.gov.br/ccivil$_{0}$3/$_{A}$to2015-2018/2016/Decreto/D8771.htm

[19] Secretaria de Comunicaciones y Transportes, "Ley Federal de Telecomunicaciones y Radiodifusión," 2014, accessed in October 16, 2017. [Online]. Available: http://www.sct.gob.mx/fileadmin/Comunicaciones/LFTR.pdf

[20] Federal Communications Commission, "Open Internet," 2015, accessed in October 16, 2017. [Online]. Available: https://www.fcc.gov/general/open-internet

[21] Telecom Regulatory Authority of India, "Prohibition of Discriminatory Tariffs for Data Services," 2016, accessed in October 16, 2017. [Online]. Available: http://www.trai.gov.in/WriteReadData/WhatsNew/Documents/Regulation$_{D}$ata$_{S}$ervice.pdf

[22] Body of European Regulators for Electronic Communications, "BEREC Guidelines on the Implementation by National Regulators of European Net Neutrality Rules," 2017, accessed in October 16, 2017. [Online]. Available: http://berec.europa.eu/eng/document$_{r}$egister/subject$_{m}$atter/berec/regulatory$_{b}$est$_{p}$ractices/guidelines/6160-berec-guidelines-on-the-implementation-by-national-regulators-of-european-net-neutrality-rules

[23] J. Crowcroft, "Net Neutrality: The Technical Side of the Debate: a White Paper," *SIGCOMM Computer Communication Review*, vol. 37, no. 1, 2007.

[24] D. J. Weitzner, "Net Neutrality... Seriously this Time," *IEEE Internet Computing*, vol. 12, no. 3, pp. 86–89, May 2008.

[25] T. Berners-Lee, "Long Live the Web," *Scientific American*, vol. 303, no. 6, pp. 80–85, 2010.

[26] H. Guo and R. F. Easley, "Network Neutrality Versus Paid Prioritization: Analyzing the Impact on Content Innovation," *Production and Operations Management*, vol. 25, no. 7, pp. 1261–1273, 2016.

[27] A. Cooper and I. Brown, "Net Neutrality: Discrimination, Competition, and Innovation in the UK and US," *ACM Transactions on Internet Technology*, vol. 15, no. 1, February 2015.

[28] Y. Yiakoumis, S. Katti, and N. McKeown, "Neutral net neutrality," in *ACM SIGCOMM*. ACM, 2016, pp. 483–496.

[29] R. Knutson and S. Ramachandran, "Netflix Throttles Its Videos on AT&T, Verizon Networks," March 2016, accessed in October 16, 2017. [Online]. Available: http://www.wsj.com/articles/netflix-throttles-its-videos-on-at-t-verizon-phones-1458857424

[30] P. Maille, G. Simon, and B. Tuffin, "Toward a net neutrality debate that conforms to the 2010s," *IEEE Communications Magazine*, vol. 54, no. 3, pp. 94–99, March 2016.

[31] Body of European Regulators for Electronic Communications, "Summary of BEREC positions on net neutrality," 2012, accessed in October 16, 2017. [Online]. Available: http://berec.europa.eu/eng/document$_{r}$egister/subject$_{m}$atter/berec/opinions/1128-summary-of-berec-positions-on-net-neutrality

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

[33] P. Kanuparthy and C. Dovrolis, "DiffProbe: Detecting ISP Service Discrimination," in *IEEE INFOCOM*, March 2010, pp. 1–9.

[34] ——, "ShaperProbe: End-to-end Detection of ISP Traffic Shaping Using Active Methods," in *Internet Measurement Conference*. ACM, 2011, pp. 473–482.

[35] T. Flach, P. Papageorge, A. Terzis, L. Pedrosa, Y. Cheng, T. Karim, E. Katz-Bassett, and R. Govindan, "An Internet-Wide Analysis of Traffic Policing," in *ACM SIGCOMM*. ACM, 2016, pp. 468–482.

[36] G. Lu, Y. Chen, S. Birrer, F. E. Bustamante, and X. Li, "POPI: A User-Level Tool for Inferring Router Packet Forwarding Priority," *IEEE/ACM Transactions on Networking*, vol. 18, no. 1, pp. 1–14, February 2010.

[37] Y. Zhang, Z. M. Mao, and M. Zhang, "Detecting Traffic Differentiation in Backbone ISPs with NetPolice," in *Internet Measurement Conference*. ACM, 2009, pp. 103–115.

[38] M. Dischinger, M. Marcon, S. Guha, K. P. Gummadi, R. Mahajan, and S. Saroiu, "Glasnost: Enabling End Users to Detect Traffic Differentiation," in *USENIX Conference on Networked Systems Design and Implementation*. USENIX Association, 2010, pp. 27–27.

[39] A. Molavi Kakhki, A. Razaghpanah, A. Li, H. Koo, R. Golani, D. Choffnes, P. Gill, and A. Mislove, "Identifying Traffic Differentiation in Mobile Networks," in *Internet Measurement Conference*. ACM, 2015, pp. 239–251.

[40] U. Weinsberg, A. Soule, and L. Massoulie, "Inferring traffic shaping and policy parameters using end host measurements," in *IEEE INFOCOM*, April 2011, pp. 151–155.

[41] Z. Zhang, O. Mara, and K. Argyraki, "Network Neutrality Inference," *SIGCOMM Computer Communication Review*, vol. 44, no. 4, pp. 63–74, October 2014.

[42] R. Beverly, S. Bauer, and A. Berger, *The Internet Is Not a Big Truck: Toward Quantifying Network Neutrality*. Springer Berlin Heidelberg, 2007, pp. 135–144.

[43] Federal Communications Commission, "FCC Classifies Cable Modem Service as "Information Service": Initiates Proceeding to Promote Broadband Deployment and Examine Regulatory Implications of Classification," 2002, accessed in October 16, 2017. [Online]. Available: http://transition.fcc.gov/Bureaus/Cable/News$_{R}$eleases/2002/nrcb0201.html

[44] T. Wu, "A Proposal for Network Neutrality," 2002, accessed in October 16, 2017. [Online]. Available: http://www.timwu.org/OriginalNNProposal.pdf

[45] L. Lessig, *The Future of Ideas: The Fate of the Commons in a Connected World*, ser. The Future of Ideas: The Fate of the Commons in a Connected World. Random House, 2001.

[46] T. Wu and L. Lessig, "Ex Parte Submission in CS Docket No. 02-52," 2003, accessed in October 16, 2017. [Online]. Available: http://www.savetheinternet.com/sites/default/files/resources/wu$_{l}$essig$_{f}$cc.pdf

[47] I. Austen, "A Canadian Telecom's Labor Dispute Leads to Blocked Web Sites and Questions of Censorship," 2005, accessed in October 16, 2017. [Online]. Available: http://www.nytimes.com/2005/08/01/business/worldbusiness/a-canadian-telecoms-labor-dispute-leads-to-blocked.html

[48] C. Reis, S. D. Gribble, T. Kohno, and N. C. Weaver, "Detecting In-flight Page Changes with Web Tripwires," in *USENIX Symposium on Networked Systems Design and Implementation (NDSI)*, 2008.

[49] R. Topolski, "Comcast is using Sandvine to manage P2P Connections," 2007, accessed in October 16, 2017. [Online]. Available: http://www.dslreports.com/forum/r18323368-Comcast-is-using-Sandvine-to-manage-P2P-Connections

[50] Sandvine, "Intelligent Broadband Networks," accessed in October 16, 2017. [Online]. Available: https://www.sandvine.com

[51] R. Cellan-Jones, "BT accused of iPlayer throttling," 2009, accessed in October 16, 2017. [Online]. Available: http://news.bbc.co.uk/2/hi/technology/8077839.stm

[52] GreatFire.org, "Expanding Online Freedom of Speech in China and Beyond," accessed in October 16, 2017. [Online]. Available: https://en.greatfire.org

[53] A. Sfakianakis, E. Athanasopoulos, and S. Ioannidis, "CensMon: A Web censorship monitor," in *USENIX Workshop on Free and Open Communications on the Internet*, 2011.

[54] Respect My Net, "Report cases of Net Neutrality violations," accessed in October 16, 2017. [Online]. Available: https://respectmynet.eu

[55] J. Bustos-Jiménez, V. Ramiro, F. Lalanne, and T. Barros, "Adkintun: SLA Monitoring of ISP Broadband Offerings," in *International Conference on Advanced Information Networking and Applications Workshops (WAINA)*, 2013.

[56] J. Bustos-Jiménez and C. Fuenzalida, "All Packets Are Equal, but Some Are More Equal Than Others," in *Latin America Networking Conference (LANC)*. ACM, 2014.

[57] M. Weber, V. Svedek, Z. Jukic, I. Golub, and T. Zuljevic, "Can HAKOMetar be used to increase transparency in the context of network neutrality?" in *International Conference on Telecommunications (ConTEL)*, 2013.

[58] C. Anderson, "Dimming the Internet: Detecting Throttling as a Mechanism of Censorship in Iran," 2013, accessed in October 16, 2017. [Online]. Available: http://arxiv.org/abs/1306.4361

[59] C. Dovrolis, K. Gummadi, A. Kuzmanovic, and S. D. Meinrath, "Measurement Lab: Overview and an Invitation to the Research Community," *SIGCOMM Computer Communication Review*, vol. 40, no. 3, 2010.

[60] Mr T., "Zambia, a country under Deep Packet Inspection," 2013, accessed in October 16, 2017. [Online]. Available: https://ooni.torproject.org/post/zambia

[61] A. Filasto and J. Appelbaum, "OONI: Open Observatory of Network Interference," in *USENIX Workshop on Free and Open Communications on the Internet*, 2012.

[62] S. Esnaashari, "Invisible Barriers: Identifying restrictions affecting New Zealanders' access to the Internet," Master's thesis, Victoria University of Wellington, 2014, http://researcharchive.vuw.ac.nz/xmlui/bitstream/handle/10063/3263/thesis.pdf.

[63] R. Shankesi, "Friendsourcing to detect network manipulation," Ph.D. dissertation, University of Illinois, 2013, https://www.ideals.illinois.edu/bitstream/handle/2142/45321/RavinderShankesi.pdf.

[64] "I just doubled my PIA VPN throughput that I am getting on my router by switching from UDP:1194 to TCP:443," 2014, accessed in October 16, 2017. [Online]. Available: https://www.reddit.com/r/VPN/comments/1xkbca/ijustdoubledmypiavpnthroughputthat\i\am

[65] J. Brodkin, "Netflix performance on Verizon and Comcast has been dropping for months," October 2014, accessed in October 16, 2017. [Online]. Available: http://arstechnica.com/information-technology/2014/02/netflix-performance-on-verizon-and-comcast-has-been-dropping-for-months

[66] B. van Schewick, "T-Mobile's Binge On Video Streaming Program," 2016, accessed in October 16, 2017. [Online]. Available: https://prodnet.www.neca.org/publicationsdocs/wwpdf/2216she.pdf

[67] T. Dreier, "Comcast Hit With FCC Complaint Over Net Neutrality Violations," 2016, accessed in October 16, 2017. [Online]. Available: http://www.streamingmedia.com/Articles/News/Online-Video-News/Comcast-Hit-With-FCC-Complaint-Over-Net-Neutrality-Violations-109609.aspx

[68] Public Knowledge, "Petition for the Federal Communications Commission to Enforce Merger Conditions and its Policies," 2016, accessed in October 16, 2017. [Online]. Available: https://ecfsapi.fcc.gov/file/60001526808.pdf

[69] American Cable Association, "ACA Statement On Netflix's Throttling Of Wireless Video Streaming Traffic," 2016, accessed in October 16, 2017. [Online]. Available: http://www.americancable.org/node/5668

[70] M. O'Rielly, "Shining the Spotlight: How FCC Rules Impact Consumers and Industries," 2016, accessed in October 16, 2017. [Online]. Available: https://apps.fcc.gov/edocspublic/attachmatch/DOC-338600A1.pdf

[71] P. S. Campbell, "Public Interest Groups Urge FCC Action Against Zero-Rating," 2016, accessed in October 16, 2017. [Online]. Available: http://www.lexology.com/library/detail.aspx?g=e4fbf6ad-03f4-4a04-83c9-f4220c6dea26

[72] J. Krämer, L. Wiewiorra, and C. Weinhardt, "Net neutrality: A progress report," *Telecommunications Policy*, vol. 37, no. 9, 2013.

[73] R. Adams, "Active Queue Management: A Survey," *IEEE Communications Surveys Tutorials*, vol. 15, no. 3, pp. 1425–1476, October 2013.

[74] Body of European Regulators for Electronic Communications, "What are specialised services and how are they relevant to the Regulation?" accessed in October 16, 2017. [Online]. Available: http://berec.europa.eu/eng/netneutrality/specialisedservices

[75] S. Jordan, "Some Traffic Management Practices Are Unreasonable," in *International Conference on Computer Communications and Networks*, August 2009, pp. 1–6.

[76] S. Jordan and A. Ghosh, "A Framework for Classification of Traffic Management Practices As Reasonable or Unreasonable," *ACM Transactions on Internet Technology*, vol. 10, no. 3, pp. 12:1–12:23, Oct. 2010.

[77] T. Wu, "Network Neutrality, Broadband Discrimination," *Journal of Telecommunications and High Technology Law*, vol. 2, p. 141, 2003.

[78] R. W. Hahn and S. Wallsten, "The Economics of Net Neutrality," *The Economists' Voice*, vol. 3, no. 6, 2006.

[79] Internet Society, "Net Neutrality," accessed in October 16, 2017. [Online]. Available: http://www.internetsociety.org/net-neutrality

[80] P. Ganley and B. Allgrove, "Net neutrality: A user's guide," *Computer Law & Security Review*, vol. 22, no. 6, 2006.

[81] A. Dainotti, A. Pescape, and K. C. Claffy, "Issues and future directions in traffic classification," *IEEE Network*, vol. 26, no. 1, pp. 35–40, January 2012.

[82] C. V. Wright, F. Monrose, and G. M. Masson, "On inferring application protocol behaviors in encrypted network traffic," *Journal of Machine Learning Research*, vol. 7, no. Dec, pp. 2745–2769, 2006.

[83] G. Detal, B. Hesmans, O. Bonaventure, Y. Vanaubel, and B. Donnet, "Revealing Middlebox Interference with Tracebox," in *Internet Measurement Conference*. ACM, 2013, pp. 1–8.

[84] A. Mendiola, J. Astorga, E. Jacob, and M. Higuero, "A survey on the contributions of Software-Defined Networking to Traffic Engineering," *IEEE Communications Surveys Tutorials*, vol. PP, no. 99, 2016.

[85] Y. Zhang, Z. Morley, and M. M. Zhang, "Ascertaining the Reality of Network Neutrality Violation in Backbone ISPs," in *ACM Workshop on Hot Topics in Networks*, 2008.

[86] M. B. Tariq, M. Motiwala, N. Feamster, and M. Ammar, "Detecting Network Neutrality Violations with Causal Inference," in *International Conference on Emerging Networking Experiments and Technologies*, ser. CoNEXT '09. ACM, 2009, pp. 289–300.

[87] J. P. Sander Greenland, James M. Robins, "Confounding and Collapsibility in Causal Inference," *Statistical Science*, vol. 14, no. 1, 1999.

[88] N. P. Jewell, *Statistics for epidemiology*. Chapman & Hall/CRC, 2004.

[89] G. E. Noether, *Introduction to statistics: the nonparametric way*. Springer Science & Business Media, 2012.

[90] M. Dischinger, A. Mislove, A. Haeberlen, and K. P. Gummadi, "Detecting Bittorrent Blocking," in *Internet Measurement Conference*. ACM, 2008, pp. 3–8.

[91] V. Bashko, N. Melnikov, A. Sehgal, and J. Schönwälder, "BonaFide: A traffic shaping detection tool for mobile networks," in *2013 IFIP/IEEE International Symposium on Integrated Network Management (IM 2013)*, May 2013, pp. 328–335.

[92] D. R. W. H. B. Mann, "On a Test of Whether one of Two Random Variables is Stochastically Larger than the Other," *The Annals of Mathematical Statistics*, vol. 18, no. 1, 1947.

[93] A. Coates, A. O. H. III, R. Nowak, and B. Yu, "Internet tomography," *IEEE Signal Processing Magazine*, vol. 19, no. 3, 2002.

[94] R. Ravaioli, C. Barakat, and G. Urvoy-Keller, "Chkdiff: Checking Traffic Differentiation at Internet Access," in *ACM Conference on CoNEXT Student Workshop*. ACM, 2012.

[95] M. Carbone and L. Rizzo, "Dummynet Revisited," *SIGCOMM Computer Communication Review*, vol. 40, no. 2, 2010.

[96] I. Trestian, R. Potharaju, and A. Kuzmanovic, "WindRider - A Mobile Network Neutrality Monitoring System," accessed in October 16, 2017. [Online]. Available: http://networks.cs.northwestern.edu/mobile-neutrality

[97] D. Miorandi, I. Carreras, E. Gregori, I. Graham, and J. Stewart, "Measuring net neutrality in mobile Internet: Towards a crowdsensing-based citizen observatory," in *IEEE International Conference on Communications Workshops (ICC)*, June 2013, pp. 199–203.

[98] M. Yuksel, K. K. Ramakrishnan, S. Kalyanaraman, J. D. Houle, and R. Sadhvani, "Quantifying Overprovisioning vs. Class-of-Service: Informing the Net Neutrality Debate," in *International Conference on Computer Communications and Networks*, August 2010, pp. 1–8.

[99] G. Hourton, G. D. Canto, J. Bustos, and F. Lalanne, "Crowd-measuring: Assessing the quality of mobile Internet from end-terminals," in *International Conference on Network Games, Control and Optimization (NetGCooP)*, November 2012, pp. 145–148.

[100] Z. S. Bischof, J. S. Otto, and F. E. Bustamante, "Up, Down and Around the Stack: ISP Characterization from Network Intensive Applications," *SIGCOMM Computer Communication Review*, vol. 42, no. 4, 2012.

[101] M. A. Sánchez, J. S. Otto, Z. S. Bischof, and F. E. Bustamante, "Dasu - ISP Characterization from the Edge: A BitTorrent Implementation," *SIGCOMM Computer Communication Review*, vol. 41, no. 4, 2011.

[102] M. Aida, N. Miyoshi, and K. Ishibashi, "A scalable and lightweight QoS monitoring technique combining passive and active approaches," in *IEEE INFOCOM*, vol. 1, 2003.

[103] Ookla, "The world standard in Internet metrics," accessed in October 16, 2017. [Online]. Available: https://www.ookla.com

[104] TestMy.net, "Broadband Internet Speed Test," accessed in October 16, 2017. [Online]. Available: http://testmy.net

[105] Broadband Speed Checker, "The UK's No.1 Broadband Speed Test," accessed in October 16, 2017. [Online]. Available: http://www.broadbandspeedchecker.co.uk

[106] NIC.br, "Sistema de Medição de Tráfego Internet (SIMET)," accessed in October 16, 2017. [Online]. Available: http://simet.nic.br

[107] J. Sommers, P. Barford, N. Duffield, and A. Ron, "Accurate and Efficient SLA Compliance Monitoring," *SIGCOMM Computer Communication Review*, vol. 37, no. 4, 2007.

[108] ——, "Multiobjective Monitoring for SLA Compliance," *IEEE/ACM Transactions on Networking (TON)*, vol. 18, no. 2, 2010.

[109] X. Ta and G. Mao, "Online End-to-End Quality of Service Monitoring for Service Level Agreement Verification," in *IEEE International Conference on Networks*, vol. 2, 2006.

[110] R. Serral-Gracia, Y. Labit, J. Domingo-Pascual, and P. Owezarski, "Towards an Efficient Service Level Agreement Assessment," in *IEEE INFOCOM*, 2009.

[111] T. Qiu, J. Ni, H. Wang, N. Hua, Y. R. Yang, and J. J. Xu, "Packet Doppler: Network Monitoring Using Packet Shift Detection," in *ACM CoNEXT Conference*. ACM, 2008.

[112] R. Serral-Gracià, M. Yannuzzi, Y. Labit, P. Owezarski, and X. Masip-Bruin, "An efficient and lightweight method for Service Level Agreement assessment," *Computer Networks*, vol. 54, no. 17, 2010.
[113] Z. Jukic, M. Weber, V. Svedek, M. Vukovic, D. Katusic, and G. Jezic, "Technical aspects of network neutrality," in *International Conference on Telecommunications (ConTEL)*, 2011.
[114] J. Bustos-Jiménez, G. Del Canto, S. Pereira, F. Lalanne, J. Piquer, G. Hourton, A. Cádiz, and V. Ramiro, "How AdkintunMobile Measured the World," in *ACM Conference on Pervasive and Ubiquitous Computing Adjunct Publication (UbiComp)*. ACM, 2013.
[115] F. Lalanne, N. Aguilera, A. Graves, and J. Bustos, "Adkintun Mobile: Towards using personal and device context in assessing mobile QoS," in *International Wireless Communications and Mobile Computing Conference (IWCMC)*, 2015.
[116] T. Hwang, "Threat Modeling: Herdict: A Distributed Model for Threats Online," *Network Security*, vol. 2007, no. 8, 2007.
[117] Network of Excellence in InterNet Science, "MorFEO: MOnitoRing network connections to assess Freedom of Expression Online," accessed in October 16, 2017. [Online]. Available: http://www.internet-science.eu/open-call-projects/morfeo
[118] "Net neutrality monitor," accessed in October 16, 2017. [Online]. Available: http://www.neumon.org
[119] G. Aceto and A. Pescapé, "Internet Censorship detection: A survey," *Computer Networks*, vol. 83, 2015.
[120] Electronic Frontier Foundation, "Switzerland Network Testing Tool," accessed in October 16, 2017. [Online]. Available: https://www.eff.org/pages/switzerland-network-testing-tool
[121] Network Neutrality Squad, "NNSquad Network Measurement Agent (NNMA)," accessed in October 16, 2017. [Online]. Available: https://www.nnsquad.org/agent.html
[122] N. Weaver, R. Sommer, and V. Paxson, "Detecting Forged TCP Reset Packets," in *Network and Distributed System Security Symposium (NDSS)*, 2009.
[123] M. Dhawan, J. Samuel, R. Teixeira, C. Kreibich, M. Allman, N. Weaver, and V. Paxson, "Fathom: A Browser-based Network Measurement Platform," in *ACM Conference on Internet Measurement Conference (IMC)*. ACM, 2012.
[124] M. Dischinger, A. Haeberlen, K. P. Gummadi, and S. Saroiu, "Characterizing Residential Broadband Networks," in *SIGCOMM Conference on Internet Measurement (IMC)*. ACM, 2007.
[125] R. Mahajan, M. Zhang, L. Poole, and V. Pai, "Uncovering Performance Differences Among Backbone ISPs with Netdiff," in *USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 2008.
[126] Z. S. Bischof, J. S. Otto, M. A. Sánchez, J. P. Rula, D. R. Choffnes, and F. E. Bustamante, "Crowdsourcing ISP Characterization to the Network Edge," in *SIGCOMM Workshop on Measurements Up the Stack (W-MUST)*. ACM, 2011.
[127] M. A. Sánchez, J. S. Otto, Z. S. Bischof, D. R. Choffnes, F. E. Bustamante, B. Krishnamurthy, and W. Willinger, "Dasu: Pushing Experiments to the Internet's Edge," in *USENIX Conference on Networked Systems Design and Implementation*. USENIX Association, 2013, pp. 487–500.
[128] SamKnows, "The global platform for internet measurement," accessed in October 16, 2017. [Online]. Available: https://www.samknows.com
[129] D. Antoniades, E. P. Markatos, and C. Dovrolis, *MOR: Monitoring and Measurements through the Onion Router*. Springer Berlin Heidelberg, 2010, pp. 131–140.
[130] V. Bajpai and J. Schönwälder, "A Survey on Internet Performance Measurement Platforms and Related Standardization Efforts," *IEEE Communications Surveys Tutorials*, vol. 17, no. 3, 2015.
[131] K. V. Vishwanath and A. Vahdat, "Swing: Realistic and Responsive Network Traffic Generation," *IEEE/ACM Transactions on Networking*, vol. 17, no. 3, 2009.
[132] F. Michaut and F. Lepage, "Application-oriented network metrology: metrics and active measurement tools," *IEEE Communications Surveys Tutorials*, vol. 7, no. 2, 2005.
[133] S. Basso, M. Meo, and J. C. De Martin, "Strengthening Measurements from the Edges: Application-level Packet Loss Rate Estimation," *SIGCOMM Computer Communication Review*, vol. 43, no. 3, 2013.
[134] P. Kanuparthy and C. Dovrolis, "ShaperProbe: End-to-end Detection of ISP Traffic Shaping Using Active Methods," in *SIGCOMM Conference on Internet Measurement Conference (IM)*. ACM, 2011.
[135] J. C. D. Martin and A. Glorioso, "The Neubot project: A collaborative approach to measuring internet neutrality," in *IEEE International Symposium on Technology and Society*, 2008.
[136] S. Basso, A. Servetti, and J. C. D. Martin, "The network neutrality bot architecture: A preliminary approach for self-monitoring of Internet access QoS," in *IEEE Symposium on Computers and Communications (ISCC)*, 2011.
[137] C. Kreibich, N. Weaver, B. Nechaev, and V. Paxson, "Netalyzr: Illuminating the Edge Network," in *SIGCOMM Conference on Internet Measurement (IMC)*. ACM, 2010.
[138] D. Batory, "Feature Models, Grammars, and Propositional Formulas," in *International Conference on Software Product Lines*. Springer-Verlag, 2005, pp. 7–20.
[139] Y.-C. Cheng, U. Hölzle, N. Cardwell, S. Savage, and G. M. Voelker, "Monkey See, Monkey Do: A Tool for TCP Tracing and Replaying," in *USENIX Annual Technical Conference (ATEC)*, 2004.
[140] A. Botta, A. Dainotti, and A. Pescapé, "A Tool for the Generation of Realistic Network Workload for Emerging Networking Scenarios," *Computer Networks*, vol. 56, no. 15, 2012.
[141] T. Garrett, S. Dustdar, L. C. E. Bona, and E. P. Duarte, "Ensuring Network Neutrality for Future Distributed Systems," in *International Conference on Distributed Computing Systems (ICDCS)*, June 2017, pp. 1780–1786.
[142] A. Maltinsky, R. Giladi, and Y. Shavitt, "On Network Neutrality Measurements," *ACM Transactions on Intelligent Systems and Technology*, vol. 8, no. 4, pp. 56:1–56:22, May 2017.

Thiago Garrett is a Ph.D. student in Computer Science at Federal University of Parana, Curitiba, Brazil, from where he also received the MSc and BSc degrees in Computer Science in 2011 and 2008, respectively. His research interests include Computer Networks and Distributed Systems, along with related topics such as the Internet of Things and Network Neutrality. He is currently a student member of the Computer Networks and Distributed Systems Lab (LaRSis), Curitiba, Brazil.

Ligia E. Setenareski is a Ph.D. student in Computer Science at Federal University of Paraná (UFPR), in Curitiba, Brazil, where she is Vice-Director of the Library System. She has a Master degree in Public Policy (2013) also from UFPR. Among her professional activities, she held the position of President of the Regional Council of Librarianship (9th Region) from 1988 to 1990. She was the representative of the South Region, Brazilian Commission of University Libraries, from 1998 to 2006. She was Director of the Library System of UFPR, from 1998 to 2014, during this time she coordinated several relevant projects on Digital Information Management, and the Construction and Management of Digital Repositories. Ligia's main research interests are on Network Neutrality.

Leticia M. Peres is an Adjunct Professor at Federal University of Parana, Curitiba, Brazil, where she is a member of the Applications and Fundamentals of Software Engineering Lab (FAES). She has a Ph.D. degree in Computer Science from Federal University of Parana, 2010, and carried out her post-doctoral studies at the Department of Services and Information Systems Engineering at the Polytechnic University of Catalonia, 2016. Her main research interests are on applications and fundamentals of systems and software engineering, with emphasis on verification and validation of systems and software.

Luis C. E. Bona is an Associate Professor at Federal University of Parana, Curitiba, Brazil, where he is member of the Computer Networks and Distributed Systems Lab (LaRSis). He obtained a Ph.D. degree in Electric Engineering at Federal University of Technology - Parana, 2006, and carried out his post-doctoral studies at the Barcelona Supercomputing Center (BSC), 2013. His research interests include Operating Systems, Computer Networks and Distributed Systems. He acted as coordinator of several research, technological and development projects, both national and international. He also served as chair of the Department of Computer Science of Federal University of Parana from 2008 to 2012.

Elias P. Duarte Jr. is a Full Professor at Federal University of Parana, Curitiba, Brazil, where he is the leader of the Computer Networks and Distributed Systems Lab (LaRSis). His research interests include Computer Networks and Distributed Systems, their Dependability, Management, and Algorithms. He has published nearly 200 peer-reviewer papers and has supervised nearly 130 students, both on the graduate and undergraduate levels. Prof. Duarte has served as chair of more than 20 conferences and workshops in his fields of interest. He received a Ph.D. degree in Computer Science from Tokyo Institute of Technology, Japan, 1997, M.Sc. degree in Telecommunications from the Polytechnical University of Madrid, Spain, 1991, and both BSc and MSc degrees in Computer Science from Federal University of Minas Gerais, Brazil, 1987 and 1991, respectively. He chaired the Special Interest Group on Fault Tolerant Computing of the Brazilian Computing Society (2005-2007); the Graduate Program in Computer Science of UFPR (2006-2008); and the Brazilian National Laboratory on Computer Networks (2012-2016). He is a member of the Brazilian Computer Society and a Senior Member of the IEEE.