Exploiting AS-level Routing Properties to Locate Traffic Differentiation in the Internet

Thiago Garrett*, Luis C. E. Bona*, and Elias P. Duarte Jr.*
*Federal University of Paraná, Brazil
Emails: {tgarrett,bona,elias}@inf.ufpr.br

***Abstract*—Network Neutrality states that all traffic in the Internet must be treated equally and thus cannot suffer unfair traffic differentiation (TD). Several solutions for detecting the presence of TD in the Internet have been proposed. However, locating where in the network TD is happening is still an open problem. In this work, we propose a strategy to locate Autonomous Systems (ASes) that are differentiating traffic. The proposed strategy takes advantage of AS-level routing properties to identify valid AS-level paths between end-hosts. It is then possible to select measurement points between which the AS-level paths traverse suspect ASes. Probes are sent from the measurement points and processed using end-to-end TD detectors based on statistical inference. The main idea is to check suspect ASes until only the AS that is actually discriminating traffic is filtered out. We first present results of experiments executed to validate the routing properties employed. Then the efficiency of the proposal for locating TD is evaluated using simulation. The results show that the proposed strategy is effective and efficient.**

***Index Terms*—Network Neutrality, Traffic Differentiation, Valley-free Paths, AS-level Paths, Internet Routing**

I. Introduction

Network Neutrality (NN) states that all traffic in the Internet must be treated equally, regardless of its origin, destination and/or content, i.e., traffic differentiation (TD) is not allowed [1]. The main motivation for NN is to ensure the Internet continues to be an open environment for innovation, fair competition, and consumer's freedom of choice [2]. On the other side, Internet Service Providers (ISPs) may employ discriminatory traffic management techniques to handle congestion, to increase revenue under commercial agreements, or even to benefit their own services, for example. NN regulations have been implemented around the world. But regulations alone may not be enough to ensure ISP compliance. Furthermore, regardless of regulations, transparent traffic management practices may contribute to a more competitive market.

Multiple solutions for detecting TD have been proposed [1], each employing different measurement and inference techniques. Nevertheless, there are still only a few solutions for locating where in the network TD is happening [3]–[6]. We argue that locating instead of just detecting TD is important both to help enforce regulations and/or to increase transparency and empower consumers.

In this work, we propose a strategy for locating which Autonomous Systems (AS) is employing TD in the Internet. The rationale is that if a particular AS is in all possible paths between two end-hosts, then TD detection probes are guaranteed to traverse that AS, and thus its behavior can be assessed. Our strategy investigate suspect ASes until only the AS that is actually discriminating traffic is filtered out. We take advantage of AS-level routing properties to select well-positioned measurement points – end-hosts from which TD detection probes are issued. TD is then located by combining the probes, taking into account the AS-level paths between the measurement points. AS-level paths in the Internet follow a set of routing policies based on the relationship between ASes [7], i.e. how they exchange traffic. In the proposed strategy, measurement points are selected in a way that AS-level paths between them traverse the ASes that are suspected of having discriminatory behavior. We argue that our proposal presents an innovative use of AS-level routing properties.

We conducted experiments on a global testbed [8] to validate our assumptions regarding the properties of AS-level paths. Then several simulations for assessing the efficiency of our proposal for locating TD were executed. These simulations also evaluated different metrics for selecting measurement points. The results obtained on the testbed experiment confirm that the majority of AS-level paths observed complied with the AS-level routing properties assumed. These experiments also show that some techniques employed by other existing solutions may not be reliable. The simulations show that our proposal is capable of locating TD between several different pairs of ASes. Furthermore, issuing measurements from a few core Internet ASes achieves similar results to issuing measurements from a large number of ASes on the edge. Results also show which metrics are better for selecting measurement points.

The rest of this paper is organized as follows. Section II presents related work. Next, an overview of AS-level routing in the Internet is presented in Section III. Our proposal for locating TD is then described in Section IV. Experiments for validating the AS-level routing properties assumed are described in Section V. Next, simulations for evaluating our strategy for locating TD are presented in Section VI. We conclude the paper in Section VII.

II. Related Work

Several solutions for detecting TD have been proposed in the last decade. A survey describing such proposals can be

978-1-7281-8086-1/20/$31.00

found in [1]. These solutions are based in network measurements and statistical inference. The idea is to issue probes from one or more end-hosts, employing multiple types of traffic, or passively measure ongoing traffic. The obtained measurements are then compared in order to check if there were any significant differences between measurements taken for different types of traffic. However, these proposals only detect if TD happened, but do not locate where TD occurred.

There are few proposals about locating TD. In [3]–[5] path discovery techniques, such as the *traceroute* tool [9], are employed to obtain the exact host-level path between end-hosts. Unfortunately, these techniques may not succeed in obtaining the exact path between end-hosts, which may turn those proposals to locate TD ineffective. Furthermore, application traffic may traverse a different path than *traceroute* probes. We evaluate the limitations of such techniques in Section V. In this work, we propose a strategy that does not rely on path discovering techniques: we consider all possible paths the traffic may take, making inferences only about ASes that were surely traversed.

Another proposal [6], based on network tomography, combines measurements from several different end-hosts to infer in which host TD occurred. Complete knowledge of the network host-level topology is assumed, as well as knowledge of the exact path traffic takes between end-hosts. These assumptions are not realistic for running the solution on the Internet since the host-level topology is not only hard to obtain, it is also constantly and rapidly changing. In this work, we assume knowledge of the AS-level topology, which is feasible to be obtained [10]. Furthermore, we do not assume which exact path traffic actually traverses. We also evaluate metrics for choosing good measurement points.

III. AS-level Routing

As is well known, the Internet is the interconnection of multiple administrative domains, the so-called Autonomous Systems (ASes). Each AS is assigned a set of IP prefixes and can be connected to other ASes. In this section, we present an overview of the AS-level routing properties we assume.

Traffic from one end-host in the Internet to another may traverse several ASes. The sequence of traversed ASes is called an AS-level path, which in this work we simply call a *path*. ASes decide to which neighbor AS to forward packets as they arrive. This decision depends on the packet final destination and on the traffic exchange agreements the AS has with its neighbors.

The relationships between ASes can be abstracted into three types [10]: (i) customer-to-provider ($c2p$), or provider-to-customer ($p2c$) in the opposite direction; (ii) peer-to-peer ($p2p$); and (iii) sibling-to-sibling ($s2s$). An AS connects to another AS in order to gain access to other parts of the Internet. In a $c2p$ relationship, a customer AS pays a provider AS for transit services, i.e., for access to part of the Internet. In a $p2p$ relationship, ASes mutually exchange traffic without payments, but only between the two ASes themselves and their customers. In a $s2s$ relationship, the two ASes belong to the same organization, thus exchange traffic freely.

The Gao-Rexford model is widely accepted for describing paths in the Internet [7]. According to this model, a path between two ASes is defined as a sequence of ASes in which for every AS providing transit (a transit provider), there is a customer AS adjacent to the transit provider. Therefore, there is always an AS paying for the transit service. Thus a path must have the following pattern: zero or more $c2p$ links, followed by zero or one $p2p$ link, followed by zero or more $p2c$ links. Moreover, any number of $s2s$ links may appear in the path. This pattern corresponds to the so-called *valley-free* property. A path that follows this property is a *valley-free path*, and a path that does not follow the property is a *valley path*.

There may exist several possible valley-free paths between two ASes. Any of the possible paths may be the actual path traversed by traffic [11]. Furthermore, the actual traversed path may change over time [12].

IV. Strategy for Locating TD

In this section, we propose a strategy for locating which AS is employing TD. Our proposal takes advantage of the valley-free property in order to select measurement points in a way that AS-level paths between them traverse the ASes suspected of employing TD. If a suspect AS is in all possible paths between two measurement points, then the TD detection probes issued between them are guaranteed to have traversed that AS, and thus its behavior can be assessed. From an initial set of suspects, the main idea is to rule out the ASes that are not employing TD until a remaining AS is left that can be identified as the responsible for TD.

Our proposal relies on five assumptions. We assume that the AS-level topology of the Internet is known, along with the relationships between ASes. Several datasets that infer AS-level topology are available, mainly based on BGP routing tables. We also assume that the valley-free property is valid, which is considered a fundamental BGP routing policy [13].

The availability of an end-to-end TD detector for checking the presence of TD between two end-hosts is also assumed. Several solutions exist as we described in Section II. Another assumption is that we are able to execute the TD detector on some set of ASes – the so-called *measurement ASes* or measurement points. This can be done by having access to end-hosts connected to those ASes, for example.

Finally, we assume that if an AS discriminates some type of traffic, this discrimination will occur regardless of the origin and/or destination of the traffic. Therefore, if an AS discriminates a specific application, all traffic from that application will be affected, regardless of where it is coming from or going to. Note that in this work we consider only TD based on content, not TD based on origin/destination of the traffic.

In this work, we refer to the concepts of *discriminatory* and *neutral* AS pairs, as well as *discriminatory*, *neutral*, and *suspect* ASes. A *discriminatory* AS pair is a pair of ASes between which TD was detected by an end-to-end TD detector, while between a *neutral* AS pair no TD was detected. An AS

that was found to be employing TD is a *discriminatory* AS, while an AS that was found to employ no TD is a *neutral* AS. A *suspect* AS may be *discriminatory* or *neutral*, but there is not enough data to infer its behavior.

The proposed strategy relies on checking valley-free paths between ASes. In order to search for these paths, we model the AS-level topology of the Internet as a directed graph in which the vertices correspond to the ASes, and the edges correspond to the relationship between the respective ASes. The search is then performed employing a modified breadth-first search, which discards paths that contain "valleys". To keep the search feasible, parameter σ is employed to establish a limit of the maximum path size with respect to the corresponding shortest valley-free path. In other words, we always search for valley-free paths with sizes that are at most σ links larger than the shortest valley-free path. Note that in the Internet real paths employed are often larger than the shortest possible path, as we observed in the experiments described in Section V.

Our proposal follows 5 steps: Initialization, AS Pair Selection, TD Detection, Inference, and Completion. An overview of our strategy is shown in Figure 1. In the Initialization, our solution receives the input and creates a set of suspects (the ASes suspected to be *discriminatory*). In the next step, AS Pair Selection, the pairs of measurement ASes from which probes will be issued are selected. The probes are effectively issued in the TD Detection step, and the outcomes of these probes are examined in the Inference step. The TD locating process returns then to the AS Pair Selection step, or finishes in the Completion step if an ending condition is met. We further describe each step below.

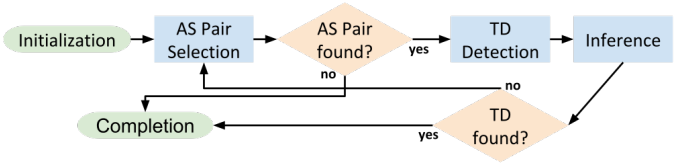


Figure 1: Overview of the proposed strategy for locating TD.

1) Initialization: The proposed strategy receives as input: (i) the AS-level topology of the Internet; (ii) a pair of initial ASes between which TD will be located; and (iii) a set of ASes available to perform measurements from. In this step, the set of suspects is initialized containing all the ASes present in the valley-free paths between the initial pair. If an AS is doing TD, it is one of these ASes.

2) AS Pair Selection: This step selects a pair of measurement ASes that will help infer the behavior of a suspect. Thus this step starts by selecting one *suspect* AS to be investigated. Then, we search for a pair of measurement ASes between which *all* valley-free paths traverse the selected *suspect* AS. The first time this step is performed, if the initial pair of ASes is available for measurement, then it is selected.

The *suspect* that appears less times in the paths between *discriminatory* AS pairs is selected to be investigated. The rationale is that such suspects are less likely to be *discriminatory*, and the idea is to identify and eliminate neutral ASes first. If no *discriminatory* pair has been found yet, the first suspect in the set is selected. We then search for a pair of measurement ASes that has not been selected previously and satisfy the criterion above (all paths traverse the suspect). We limit this search with parameter δ, which sets the maximum valley-free distance from the selected suspect up to which measurement ASes are checked on the graph. Therefore, the proposed strategy tries to form an AS pair starting from measurement ASes closer to the suspect, up to measurement ASes that are at distance δ to the suspect. The valley-free property makes this search computationally feasible since it limits the possible paths between ASes. Is there are no more pairs to investigate a suspect, the next suspect is chosen.

3) TD Detection: In this step, an end-to-end TD detector is executed to detect the presence of TD between the AS pair selected in the previous step.

4) Inference: In this step, the outcomes of the TD detection measurements, issued in the TD Detection step, are combined. The idea is to filter the suspects, eliminating *neutral* ASes until only the *discriminatory* AS remains. The rationale is that while there are two or more *suspect* ASes in the same set of paths between a *discriminatory* pair, it is not possible to infer which one is practicing TD, since we do not know which of them were actually traversed by the TD detection traffic. Inference is done in two parts. First, the *neutral* pairs of ASes are examined to search for *neutral* ASes. Then, the *discriminatory* pairs of ASes are examined to search for *discriminatory* ASes.

In the first part, for each *neutral* AS pair, we search for the set of ASes that are present in all valley-free paths between the ASes in the pair. The ASes in this set are guaranteed to have been traversed by the TD detection probes since they are in all possible paths. The ASes in this set are thus classified as *neutral* and are no longer suspects – this includes at least the ASes in the *neutral* pair, and the suspect for which the pair was selected on the AS Pair Selection step.

Then, for each *discriminatory* AS pair, we take all valley-free paths between the pair and remove the *neutral* ASes from such paths. If there is a single *suspect* AS left in all non-empty paths, then such AS is classified as *discriminatory*. The rationale is that all other suspects were found to be *neutral*, so the remaining AS is the only that could have been the responsible for TD. If there is more than one AS left in the paths, they remain as suspects.

5) Completion: The TD locating process may complete under three conditions: (i) a *discriminatory* AS between the initial pair is found; (ii) all ASes between the initial pair are classified as *neutral*; or (iii) there are no more measurement AS pairs available. In the first two cases, the process is considered to have finished successfully, while in the last case the process did not succeed. The output consists of three sets: *neutral* ASes, *discriminatory* ASes, and *suspect* ASes.

V. AS-level Paths in the Internet: Validation

In this section, we present an experiment executed on the PlanetLab global testbed. The goal is to validate our assumptions regarding AS-level routing properties. First, we briefly

describe the AS-level topology graph and the dataset from which the graph was built, employed both in this experiment and in the simulations presented in the next section. Then we describe the experiment and the results.

The AS-level topology graph employed in our evaluations was built from the dataset published by CAIDA within their AS Rank project[1]. This dataset contains the relationship between numerous ASes, inferred based on BGP data [10]. However, some ASes in the dataset have no relationship with other ASes. We thus ignored those ASes in our evaluations. The dataset we employed was obtained in October 2018. It contains 86622 different ASes, from which 24815 have no inferred relationships with other ASes: 61807 ASes are thus considered in our evaluations.

We measured the AS-level paths between 29 PlanetLab hosts and a large amount of Internet IP prefixes. We employed the list of Internet prefixes and corresponding ASes published in May 2018 by CAIDA[2]. Several ASes control more than one prefix. In these cases, one prefix was chosen for each AS. Furthermore, some ASes from the prefix list were not present in our AS-level topology graph, and were thus discarded. The resulting list contained 60578 prefixes/ASes.

From each PlanetLab host, we continually measured the paths to all prefixes from our list using the *traceroute* tool. The experiment took place from January 10, 2019 to February 1, 2019 (22 days of measurements). For each measurement obtained, we mapped the IP addresses to the corresponding ASes, using the list of prefixes from CAIDA. Thus we converted the host-level paths acquired by *traceroute* to AS-level paths. However, it is common for some hosts not to reply *traceroute* probes, or to reply with an invalid IP address. In such cases, we can not know the corresponding AS is in the path, unless another host within the same network replies to another probe during the same measurement.

We then classified all paths measured as *valley*, *valley-free*, or *unknown*. Paths that follow the valley-free property in the graph are classified as *valley-free*, otherwise they are classified as *valley*. A portion of the measured paths presented measurement errors, as described above. These errors resulted in incomplete paths: for some hosts of these paths the corresponding ASes were missing. Whenever ignoring these errors caused the resulting path to be valley-free, then it was classified as *valley-free*: in those cases, we considered that another host of the same AS replied correctly. Otherwise, paths are classified as *unknown*, since we failed to obtain the complete set of ASes and thus cannot know the actual classification. We excluded from our results the paths that contained links not in the graph.

A total of 75597104 *traceroute* measurements were issued, but 1801089 were excluded due to missing links (2.38%). From the remaining 73796015 measurements, 55.34% (40837151, more than half) resulted in *unknown* paths, which clearly shows the limitation of measuring paths with the *traceroute* tool. 44.31% (32703036) of the measurements resulted in *valley-free* paths, the vast majority of measurements that were not *unknown*, while 0.35% (255828) of the measurements resulted in *valley* paths. The *valley-free* paths reached 48283 different ASes (79.7% of all prefixes measured).

We also evaluated the sizes of the measured valley-free paths, taking into account the parameter σ. We compared the size of measured valley-free paths with the size of the corresponding shortest paths in the graph. In our experiment, 55.78% of valley-free measurements corresponded to shortest paths, while 31.87% traversed paths with one more edge in comparison with the corresponding shortest paths, and 10.34% were two edges larger.

VI. Locating TD: Simulations

In this section, we present simulation results for evaluating the proposed strategy to locate TD. A large number of simulations under different conditions were executed. The goals of these simulations are to evaluate if the proposed strategy is capable of locating TD and to identify which ASes are better measurement points. We employ three main criteria for comparing sets of measurement ASes: the success rate, which is the portion of the simulations in which TD was successfully located; the average number of probes; and the number of ASes available for measurement. The optimal set of measurement ASes is the one that achieves the largest success rate, issuing the least amount of probes, and containing the least amount of ASes available for measurement. The rationale is that it may not be feasible to have access to a large number of different ASes. Furthermore, issuing a large number of probes presents an overhead to the network.

In our simulations, we employed the same AS-level topology graph described above in Section V. We executed several groups of simulations. All groups employed the same set of initial pairs (the ASes between which TD should be located), but each employed a different set of measurement ASes (the ASes available for measurement). In each group of simulations, for each initial pair, we take all ASes present in the valley-free paths between that pair. For each of these ASes, we then execute one simulation, fixing that AS as the *discriminatory* AS for that simulation. The simulation is successful if TD is located in that AS. We also execute a simulation considering that no AS is employing TD, in which case the simulation is successful if all ASes between the initial pair are classified as *neutral*.

The sets of measurement ASes employed in our simulations were built based on metrics extracted from the graph, as well as on the classification of ASes available on the PeeringDB website [14]. PeeringDB is an online database where operators contribute information about their networks. The metrics employed for listing ASes from the graph are degree, betweenness centrality, and valley-free betweenness centrality.

Betweenness centrality measures to which extent a vertex is present in paths between all other vertices. The betweenness of a vertex is the sum of the fractions of shortest paths between all other pairs of vertices in which the vertex is present

[1]http://as-rank.caida.org/

[2]http://www.caida.org/data/routing/routeviews-prefix2as.xml

[15]. The *valley-free betweenness centrality* is a variation that takes into account only the shortest *valley-free* paths. The rationale is that, since our proposal searches for measurement ASes that are in paths traversing certain ASes (the suspects), the betweenness centrality may be a good indicator of how effective an AS is to be used for measurements – ASes with higher betweenness belong to more paths, therefore are more likely to be selected as a measurement AS.

Table I shows the sets of measurement ASes selected. The columns of the table indicate for each set: name, description, and number of ASes. The first three sets were taken from the PeeringDB website, on June 20th, 2019. The last three sets consist of the n ASes with the largest values for the corresponding metrics. The values of n we employed were: 10, 50, 100, 500, and 1000.

Table I: Sets of Measurement ASes

Name	Description	Size
pdb-access	Access providers from PeeringDB	5263
pdb-transit	Transit providers from PeeringDB	2293
degree-le-2	ASes with degree ≤ 2 in the graph	41247
degree-top-n	ASes with the largest degree	n
vfbet-top-n	ASes with the largest valley-free betweenness	n
bet-top-n	ASes with the largest betweenness	n

The set of initial pairs employed on the simulations presented in this work was built using the ASes from the set of measurement ASes *pdb-access*. It contains 1000 AS pairs randomly selected from *pdb-access*, i.e., from all the possible pairs between access providers (from PeeringDB), we randomly picked 1000 pairs. This set represents a common situation in the Internet: two end-hosts, connected to access providers, communicating with each other. Using this set of initial pairs, each group of simulations resulted in 7818 simulations. 18 groups of simulations were executed (one for each set of measurement ASes), thus 140724 simulations were executed in total.

In each simulation, we assume that the ASes in the initial pair are also available for measurement. Furthermore, to run a simulation, an end-to-end TD detector is required. We simulate the TD detector with an "oracle" detector instead of generating real measurement traffic. The oracle detector receives as input two ASes, between which the presence of TD is to be checked. The oracle works by checking if the AS responsible for TD in the current simulation is in any valley-free path between the two input ASes. If it is not present in any path, then there is no TD and the given AS pair is *neutral*, since traffic does not traverse the *discriminatory* AS for that simulation and thus can not be discriminated. Otherwise, the oracle assumes the worst case, which corresponds to traffic traversing the path containing the *discriminatory* AS, and thus the given AS pair is *discriminatory*. In the case of simulations with no TD, the oracle always returns *neutral*.

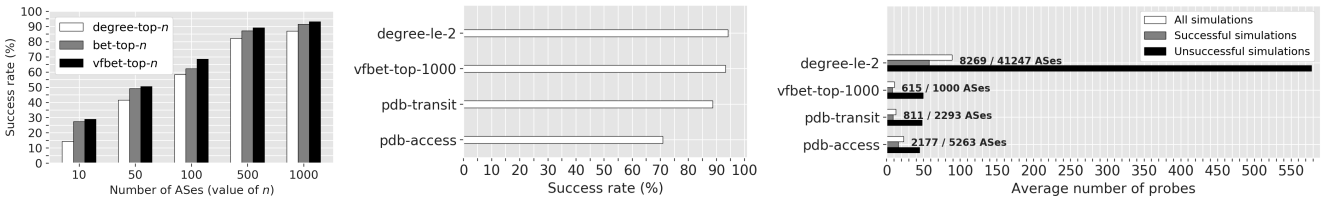
We employed two extra parameters for selecting AS pairs, mp and mt, in addition to δ and σ. Parameter mp is the maximum number of AS pairs that may be selected to investigate a suspect. If mp AS pairs have already been checked to investigate a suspect, that suspect will no longer be investigated. mt is the maximum number of times our strategy tries to form a pair a given measurement AS. If for mt times the paths between pairs containing the same AS do not all traverse the suspect, we no longer try to form measurement pairs using that AS for the suspect under investigation. These parameters limit the search space for AS pairs, making it feasible to execute a large number of simulations.

The following values for the parameters were employed. Parameter $\delta = 2$, thus only measurement ASes up to 2 hops away from the suspects are considered. Parameter $mp = 40$, thus up to 40 AS pairs are selected for each suspect, and $mt = 20$, thus we discard a measurement AS after 20 attempts when searching for AS pairs for each suspect. Larger values for these parameters significantly increase the search space and execution times, but achieve similar results. Parameter $\sigma = 0$, thus we examine only the shortest valley-free paths between ASes. Larger values for σ resulted in similar success rates and more probes, but the same conclusions are drawn.

We now present the results. First, we compare the metrics degree, betweenness and valley-free betweenness centrality. Figure 2a shows the success rates achieved by the sets of measurements ASes *degree-top-*n, *bet-top-*n and *vfbet-top-*n – $n \in \{10, 50, 100, 500, 1000\}$. For all sizes, sets *degree-top-*n achieved the smallest success rates, while *vfbet-top-*n achieved the highest values, ranging from 29% for *vfbet-top-10* to 93% for *vfbet-top-1000*. Since the *vfbet-top-1000* set achieved the best success rate, we will show no more results for the other sets in this work. ASes in the *vfbet-top-*n sets are generally closer to the suspects when compared to ASes in *degree-top-*n and *bet-top-*n sets. There are usually less possible paths and less ASes between the selected AS pairs from *vfbet-top-1000*, and thus the *discriminatory* AS in each simulation appears less often in these paths, making it easier to filter the *neutral* ASes.

Next, we compare the following sets of measurement ASes: *degree-le-2, pdb-access, pdb-transit*, and *vfbet-top-1000*. Figure 2b shows the success rates achieved by each set, while Figure 2c shows the average number of probes for all simulations, including those that were successful and those that did not succeed. The values beside each set of bars in the Figure indicates the number of different ASes selected for measurement across all simulations for the corresponding set of measurement ASes, as well as the total number of measurement ASes available.

Results show that the *degree-le-2* and *vfbet-top-1000* sets achieved the best success rates, 94% and 93%, respectively. However, *degree-le-2* employed a significantly larger number of probes on average. This happens because the ASes selected for measurement from *degree-le-2* are usually farther from each other than the AS pairs selected from *vfbet-top-1000*. ASes in *degree-le-2* are on the edge of the Internet, while ASes in *vfbet-top-1000* are on the core. Therefore, more AS pairs were needed in simulations with *degree-le-2*, since several of the selected pairs do not help to filter the suspects in the inference step of our strategy. 8269 different ASes (from a total of 41247) were selected for measurement from the *degree-le-2*



(a) (b) (c)

Figure 2: Results: comparing multiple sets of measurement ASes.

set, and 615 (from a total of 1000) from *vfbet-top-1000*.

Furthermore, *pdb-transit* achieved a slightly smaller success rate than *vfbet-top-1000* (88%), with a similar amount of probes. However, more ASes were employed on measurements (811 from a total of 2293). The set *pdb-access* achieved the smallest success rates, 71%. For all sets of measurement ASes, the average number of probes employed in unsuccessful simulations is significantly larger than those of successful simulations. This is due to the termination conditions we adopted: all the possible measurement AS pairs for all suspects are selected in every simulation that does not succeed.

The results presented in this section show that the proposed strategy is capable of locating TD under the assumptions made. Furthermore, we show that the valley-free betweenness centrality is a good metric for selecting measurement points. Having few measurement ASes (1000 from the *vfbet-top-1000* set) on the core of the network led to similar success rates as having a larger number of measurement ASes on the edge (41247 from the *degree-le-2* set). Moreover, using ASes on the core resulted in much fewer probes. ASes on the core are generally closer to a larger portion of the network, while ASes on the edge are often farther away. Therefore, to achieve good success rates using ASes on the edge, there should be a much larger number of them available for measurement at several parts of the network, in order to "cover" several vantage points.

In the wild, it is possible that TD is mistakenly detected or traffic traverses valley paths. It is also possible that an AS treat traffic differently depending on which portion of its network the traffic is traversing. Furthermore, the inferred AS-level topology may be incomplete. In all these scenarios, false-positives or false-negatives might happen when running the proposed strategy. On the other hand, the oracle detector always assumed the worst case, i.e. it considered the traffic would always follow the path containing the *discriminatory* AS in each simulation. However, if the actual path may not traverse the *discriminatory* AS, fewer probes might be necessary to locate TD, since suspects might be filtered earlier.

VII. Conclusion

In this work, we proposed a strategy for locating which AS between two end-hosts is employing TD. The proposed strategy investigates several suspect ASes until only the discriminatory AS remains. We take advantage of the valley-free property of AS-level paths to select measurement points between which valley-free paths traverse the suspects. End-to-end measurements between the measurement points then help infer the behavior of the suspects. We argue that our proposal presents an innovative use of AS-level routing properties. To evaluate our proposals, we first conducted an experiment on PlanetLab that validated the AS-level routing properties assumed. We then executed a large number of simulations that show the efficiency of our strategy to locate TD. The simulations also show that the valley-free betweenness centrality is a good metric for selecting measurement ASes.

Future work includes evaluating our strategy under more different scenarios, varying the parameters and inputs, such as employing different sets of initial pairs. Another idea is to create a system that, after locating which AS is discriminating traffic, deviates traffic through a path known to be fully neutral, circumventing the discriminatory AS.

References

[1] T. Garrett, L. E. Setenareski, L. M. Peres, L. C. E. Bona, and E. P. Duarte, "Monitoring Network Neutrality: A Survey on Traffic Differentiation Detection," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 3, 2018.

[2] H. Schulzrinne, "Network Neutrality Is About Money, Not Packets," *IEEE Internet Comput.*, vol. 22, no. 6, 2018.

[3] Y. Zhang, Z. M. Mao, and M. Zhang, "Detecting Traffic Differentiation in Backbone ISPs with NetPolice," in *ACM SIGCOMM IMC*, 2009.

[4] 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)*, 2015.

[5] E. Gregori, V. Luconi, and A. Vecchio, "Studying Forwarding Differences in European Mobile Broadband with a Net Neutrality Perspective," in *European Wireless Conference*, 2018.

[6] Z. Zhang, O. Mara, and K. Argyraki, "Network Neutrality Inference," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 4, 2014.

[7] P. Gill, M. Schapira, and S. Goldberg, "A Survey of Interdomain Routing Policies," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 1, 2013.

[8] B. Chun, D. Culler, T. Roscoe, A. Bavier, L. Peterson, M. Wawrzoniak, and M. Bowman, "PlanetLab: An Overlay Testbed for Broad-Coverage Services," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 33, no. 3, 2003.

[9] B. Augustin, X. Cuvellier, B. Orgogozo, F. Viger, T. Friedman, M. Latapy, C. Magnien, and R. Teixeira, "Avoiding Traceroute Anomalies with Paris Traceroute," in *ACM SIGCOMM IMC*, 2006.

[10] M. Luckie, B. Huffaker, A. Dhamdhere, V. Giotsas, and k. claffy, "AS Relationships, Customer Cones, and Validation," in *ACM SIGCOMM IMC*, 2013.

[11] M. E. Tozal, "Enumerating single destination, policy-preferred paths in AS-level Internet topology maps," in *IEEE Sarnoff Symposium*, 2016.

[12] S. Cho, R. Nithyanand, A. Razaghpanah, and P. Gill, "A Churn for the Better: Localizing Censorship Using Network-level Path Churn and Network Tomography," in *ACM CoNEXT*, 2017.

[13] V. Giotsas and S. Zhou, "Valley-free violation in Internet routing – Analysis based on BGP Community data," in *IEEE International Conference on Communications (ICC)*, 2012.

[14] A. Lodhi, N. Larson, A. Dhamdhere, C. Dovrolis, and k. claffy, "Using peeringDB to Understand the Peering Ecosystem," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 2, 2014.

[15] U. Brandes, "A faster algorithm for betweenness centrality," *The Journal of Mathematical Sociology*, vol. 25, no. 2, 2001.