2018 IEEE 32nd International Conference on Advanced Information Networking and Applications

ISPANN: A policy-based ISP Auditor for Network Neutrality violation detection

Vinícius Garcez Schaurich, Márcio Barbosa de Carvalho, Lisandro Zambenedetti Granville
Institute of Informatics – Federal University of Rio Grande do Sul
Av. Bento Gonçalves, 9500 – Porto Alegre, Brazil
Email: {vgschaurich,mbcarvalho,granville}@inf.ufrgs.br

***Abstract*—Network Neutrality is a controversial and full of ambiguity topic. Several works measure network features in the end-user vantage point to detect traffic differentiations, which are judged as Network Neutrality violations. However, these works neglected that each country has their own Network Neutrality rules. Some countries consider specific cases of traffic differentiations as Network Neutrality violations, and are not as general as previous works believed. In this work we consider violations directly from governments legislators Network Neutrality rules. In this sense, we propose ISPANN, a system which takes as input countries' Network Neutrality rules and audits an ISP network, identifying Network Neutrality violations. No other work proposes Network Neutrality violation detection in the ISP operator vantage point, to the best of our knowledge. We conducted an evaluation that assumes an SDN based ISP network to verify Network Neutrality violations based on OpenFlow switches flow tables and network's informations.**

***Index Terms*—Network Neutrality, Policy Based Management**

I. Introduction

Network Neutrality (NN) has been the source of huge debate between the academic community, legislators, and society in general. NN means equality on the Internet access; *i.e.*, traffic of different sources, destinies, or applications, should not be treated differently by ISPs. On the one hand, NN proponents point that ISPs have incentives to discriminate Content Providers (CPs) and end-users traffics in order to have economical advantages. As an example, a CP might pay ISPs to have priority on their network, thus having service advantages over a second CP. Therefore, legislators should state which actions are allowed to be taken over the users traffic by ISPs [1]. On the other hand, NN opponents claim that regulations reduce ISPs incentive to enhance their service and make innovative technologies deployment more difficult.

Ultimately, the NN debate leads countries' legislators to establish rules that ISPs must follow [2]. Despite being a political-economical oriented subject, NN rules have substantial technical consequences in ISP's networks. For instance, network components must be configured to avoid violating these rules. In order to try to infer NN violations committed by an ISP, several studies propose techniques to identify traffic differentiations from the end-user vantage point [3][4][5]. These violations inferences are based on users traffic statistics measurements, such as packet loss, jitter, and latency.

In a particular case, a Chilean NN violation detection tool called Adkintun [4] has been reported to support user rights in legal complaints against Chilean ISPs. However, to the best of our knowledge, there is no such a violation detection study to help network operators to verify that their network meets government's NN rules. As networks are complex and its configuration is not straightforwardly comparable to NN rules, it remains infeasible for an human to check whether the configuration of a whole network violates NN rules. In addition, besides Adkintun, NN violation detection studies tend to be disconnected from the NN regulation of the countries they take place. Authors often cite an event that has been debated by the society (for example, Comcast's shaping [6]) to justify their NN violation detections, disregarding countries' NN policies.

In this work, we present *ISPANN*, a system that audits whether a network is conformed to the NN rules stated in the ISP country. *ISPANN* can be used by network operators to perform a self-assessment of their networks, or can be used by third parties (a legislator, a regulation body, or a policy specialist) to describe the NN violation detections that should be performed by ISPs of a given country. The assessment of the network is a huge and recurring task because network configuration usually is based on a large number of rules that can be complex spread on multiple devices. Besides that, they are constantly changed by administrators along the network operation. Providing a system that is aware of NN legislation stated in the countries, we aim to bring technical literature more in line to the existing political-economical matter.

In this sense, *ISPANN* takes as input the country the ISP resides to determine the NN rules it must follow. Each NN rule is associated to a detection algorithm that can be introduced in the system by the network operator itself or by a third party. The network operator provides the necessary information to communicate with its infrastructure devices, which are used by the *ISPANN* to collect the network information needed by the algorithms. Analyzing these informations, the system is able to point whether the NN rules are being violated and which configuration are violating them. The system does not expose the user traffic since the algorithms are performed in the ISP vantage-point.

For our system evaluation, we assume an ISP that utilizes an SDN based network topology. By gathering the ISP's traffic statistics and flow tables of it's OpenFlow switches, our system searches and identifies violations according to those rules input. Our results show that different government's politics and sources of network information (*i.e.*, only flow tables versus

1550-445X/18/$31.00 ©2018 IEEE
DOI 10.1109/AINA.2018.00155

flow tables and traffic statistics) produce different detection results. These results were gathered from a series of NN violation detection algorithms we implemented in *ISPANN*. We emphasize that these algorithms are not our focus in this paper. Instead, we made *ISPANN* easily adaptable to new violation detection algorithms, thus policy specialists, regulators or network operators can work on their own algorithms as they judge pertinent to their countries NN regulations.

The rest of this paper is organized as follows: we discuss NN and the existing related literature in Section II. Then, in Section III, we made a summary of NN policies aiming at those we utilized in *ISPANN* evaluation. In Section IV we describe our approach to the NN violation detection problem, as well as the system architecture and algorithms that were implemented to detect NN violations. Our evaluation setup and results are presented in Section V. Finally, in Section VI, we present our final remarks and proposed future works.

II. NN Violation Detection

End-user traffic differentiation, and the consequent NN debate, is associated with the emergence of bandwidth demanding applications. As one of these applications is deployed and overloads an ISP network with a large amount of traffic, that ISP may end up throttling it. In the past, BitTorrent and other peer-to-peer content (usually illegal) sharing were the targeted applications to be throttled. BitTorrent blocking and shaping events lead researchers to study how ISPs were treating applications differently at their network [7]. More recently, Netflix has been reported to be the application which occupies most of the downstream traffic at the United States [8], becoming the main target of ISP's throttling and traffic differentiation [3]. In general, these traffic differentiations have been considered by researchers as NN violations.

Focusing in end-user traffic differentiations has lead the community to perform various studies on the detection of NN violations. Such studies can be categorized in two main groups, according to the type of traffic measurement and evaluation they perform: passive and active. The first group aims to passively collect and analyze end-user traffic to detect differentiations. In contrast, the second group is composed of works that generate traffic to make their differentiation tests. For the first group, Tariq *et. al* deployed NANO [9], a system that infers whether a performance degradation relates to ISPs policies. NANO collects clients and network features (*e.g.*, IP addresses, TCP retransmits, TCP duplicate ACKs), identifies equal confounding factors, and compares services among multiple ISPs to reach traffic differentiation inference.

The literature presents a larger amount of studies from the second group. Martin and Glorioso deployed Neubot [10], which is a network feature measurement application that runs in end-user's devices and actively generates traffic tests against a server or in peer-to-peer mode and centralizes these statistics in a Database Server which allowed authors to further verify NN violations. Li et. al [5] utilized Neubot and, basing their work in a transformation of the Mathis model, focused on packet-loss statistics to detect traffic differentiations. Zhang et. al [11] developed an algorithm that takes as input a network graph and end-to-end measurements and identifies non-neutral link sequences. Kakhki et. al [3] studied traffic differentiations in mobile networks where they recorded user traffic and replayed it to a test server with and without VPNs, which are an usual countermeasure adopted by end-users to bypass blocking and traffic differentiation, to compare communication statistics and infer NN violations.

Finally, Bustos and Jimenez [4] implemented Adkintun, an end-user application that tests many network features against a test server, similar to Neubot. Using a different approach, Bustos and Jimenez consider the Chilean government NN rules for their violation detection work. However, these studies detect traffic differentiations only in the end-user vantage point and, to the best of our knowledge, no study proposes this kind of detection in the network operator vantage point. In addition, the only work that bases its NN violation detection in governments' NN rules, Adkintun, considers specifically Chilean rules and, therefore, a more generic approach should be applied to this topic.

In our system, that is presented in Section IV, we consider governments' NN rules as the main hint of traffic differentiation to be detected on an ISP's network. The NN rules of the country were the ISP is located are taken as input in our system, making it generic to any country that has a NN regulation. Further, instead of end-user vantage point measurements, we base our NN violation detection on the network operator vantage point using communication measurements, network topology information, and devices configurations.

III. Policies

Governments define policies over NN that ISPs must follow, just as a client's Service Level Agreement (SLA). Each country's legislator has its own point-of-view and understanding about NN and, therefore, policies may diverge in some aspect [2]. For example, zero-rating, the act of not charging the end-user over an specific service, is accepted in countries like Brazil, but has limits in Europe with the Body of European Regulator for Electronic Communications' (BEREC) regulation. BEREC defines that zero-rating policies must act over a group of services of the same type, i.e. all message exchangers, and not only to one message exchanger specifically. This section presents an overview about the differences on policies from different countries. The policies below were used as use cases in *ISPANN* evaluation.

Chile was the first country to advocate and propose a NN regulation law [12]. Published in August of 2010, the policy establishes that ISPs *can't arbitrarily block, interfere, disturb or restrict the rights of any Internet user to utilize, send, receive or offer any legal content, application or service in the Internet*.

Then, in 2015, the US' legislator, Federal Communication Commission (FCC), stated the Open Internet [13], it's NN policy. The FCC discussion over the Open Internet regulation, that started around 2010 and ended up with it's legislation, made the NN problem a worldwide topic. Open Internet is

based in three clear points: no *blocking* or *throttling* legal services or applications and no *paid prioritization*, which states that broadband providers may not favor some lawful Internet traffic over other lawful traffic in exchange for consideration of any kind.

In 2016, BEREC proposed a guideline to European countries to define their NN policies [14]. These guidelines present seven principles which should be used by legislators when assessing ISPs' practices: *no blocking, slowing down, alteration, restriction, interference with, degradation and discrimination between specific content, applications or services, or specific categories thereof.*

Given these three NN policies, we can draw some conclusions. For example, Blocking is unaccepted in these three policies, so it is a common violation detection to be made in Chile, US and Europe. In contrast, Chile identifies user discrimination in the network, while both FCC and BEREC regulations focus on services and applications on the Internet. Furthermore, FCC explicitly describes paid prioritization as a NN violation, while the other two do not, thus, detections of economical advantages must be considered in the US. These differences are taken in account in the violation detection implemented in *ISPANN*, which is presented in Section IV.

IV. ISPANN

In this Section we will introduce *ISPANN*, a system that takes governments NN policies and audits an ISP's network, trying to detect NN violations to those rules. Firstly, we present the architecture of the system in the Subsection IV-A. The system architecture was modeled to be as generic as possible, both for different NN policies and for different network protocols used by ISPs for network configuration. Then, we detail the use case assumed for the evaluation of the system in the Subsection IV-B. This second subsection presents the scenario we assumed to evaluate *ISPANN* and a series of algorithms we implemented in it. As said, these algorithms are not the main focus of the paper, being *ISPANN* our major contribution. Instead, we implemented these algorithms to verify the differences in the policies cited in Section III and show how different NN neutrality policies need different sorts of NN violation detection requirements.

A. Architecture

ISPANN was modeled and implemented to be as generic as possible, fitting to different ISPs' infrastructures and technologies. In this sense, our system is based on four modules: *Detection Parameters Interface*, *Detection Description Interface*, *Network Infrastructure Interface*, and *NN Verification Module*. The Figure 1 presents the architecture from *ISPANN*.

Network operators input the protocols which their topology is based and the country where their infrastructure is located in the *Detection Parameters Interface*. In the *Detection Description Interface* a third party (whom can either be legislators or the network operator itself) input what detection should be executed for a given country. The *Network Infrastructure Interface* collects network data based on the legislation and the protocols inputed in the *Detection Parameters Interface*. The *NN Verification Module* centralizes the information gathered from the network and tests if any violation to the legislation input exists.

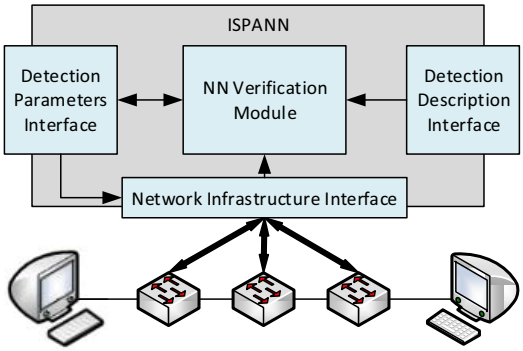

Fig. 1. System architecture

In the *Detection Parameters Interface*, the network operator informs the management protocol used to communicate with its network infrastructure and the country where the ISP is located and provides its service. As an example, if the ISP's infrastructure is based in a traditional network, the communication protocol informed by the network operator could be NetConf, for switches configuration polling. In addition, the country information makes *ISPANN* aware of the policies that must be tested in the network.

The *Detection Parameters Interface* then informs the *Network Infrastructure Interface* the communication protocol that must be used to collect network informations and the country input by the network operator, so the *Network Infrastructure Interface* knows which data it must collect from the network. Given said protocol, the *Network Infrastructure Interface* polls the relevant network data, such as flow paths, switch port queues informations, depending on the protocol capabilities, and sends the collected data to the *NN Verification Module*.

The *NN Verification Module* receives the network information and the NN policies to be tested, defined by country, and performs the NN violation detection algorithms related to this data input. Each NN policy has a set of violation detection algorithms linked to it. Hence, the NN violation detection algorithms implemented in *ISPANN*, as an use case for this work, are presented in Subsection IV-B. *ISPANN* modularity makes so that NN violation detection algorithms can be easily interchanged. It allows the community and legislators to apply their understanding about countries NN violations and, consequently, develop new NN violation detection algorithms.

After investigating and verifying the existence of NN violations, the *NN Verification Module* returns the flows suffering from these violations to the *Detection Parameters Interface* for operators visualization. These informations behave as alerts, so that network operators are cognizant and can validate that these NN violations are not premeditated or are the result

of another management mechanism (for example, a blocked IP associated to a DoS). *ISPANN* does not change network devices configuration states.

Finally, in the *Detection Description Interface* a legislator or the network operator itself may describe the NN violation detection that must be performed for a certain policy. This third party agent inputs a country to *ISPANN* and identifies the algorithms that must run when this given country is input in the *Detection Parameters Interface*. The country and the algorithms linked to its NN violation detection are stored in the *NN Verification Module*.

B. Use case

In this work, and for *ISPANN* evaluation, presented in Section V, we considered an ISP located in the US that bases its infrastructure in an SDN network. SDN is a paradigm that proposes the separation of the network forwarding and the control plane, which are both coupled in traditional network devices [15]. This approach is achieved by the introduction of a network component called controller, that coordinates the packet forwarding decisions of the remaining network devices. The control and forwarding plane decoupling makes the network more flexible and facilitates the implementation and development of novel technologies.

For the controller, Floodlight was assumed [16], a network controller implemented in Java which communicates with the network devices via the OpenFlow protocol. The system utilizes Floodlight's Northbound API to acquire the network information that is forwarded to the NN verification module. So, in this scenario, for the system execution, the network operator inputs OpenFlow as the protocol that it communicates with its network (in the case of utilizing OpenFlow, Floodlight need to be specified as the network controller as well) and US as the country of its operation.

Even though we assume an US' ISP, it is important to consider the different policies discussed in Section III, to show how these differences can cause different results on NN violation detections. From the three policies considered, we conceived four NN violation detection classifications and linked it to the corresponding policy with those characteristics.

The four NN violation classes are: **blocking**, **user discrimination**, **application/service discrimination** and **paid prioritization**. The Figure 2 shows the connections between the NN violation detection classification and the three policies. These connections are input to the system via the *Detection Description Interface*. We explain our interpretation of the NN policies that lead us to devise such classes below. Also, we present NN violation detection algorithms based on these interpretations. Each one of these classes, beside user discrimination class, has a respective NN violation detection algorithm.

User discrimination has two algorithms implemented in this work: one which uses topology information, such as switches connections and user communication latency, and another that uses only flow table informations. The second algorithm was implemented to show how having different sets of network information impact NN violation detection.

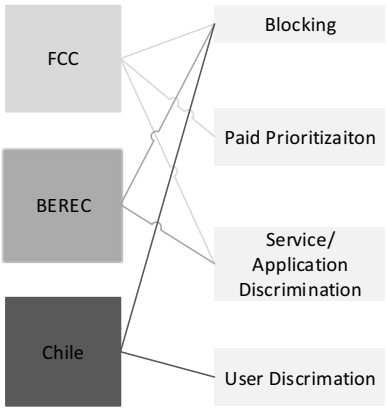

Fig. 2. Policies, NN violation detection classes and their relations

1) Blocking: Communication blocking prohibition is a tendency in most NN violations and occurs in the three policies utilized in this work. A service or user is considered blocked when a switch has an OpenFlow drop packet rule referencing its IP. To detect this kind of NN violation, the Algorithm 1 look up all rules of all switches (lines 1-7) checking whether rules are "drop packet rules" (lines 3-5).

Algorithm 1 Blocking Detection

```
1: for all switches do
2:     for all switch rules do
3:         if is packet drop rule then
4:             alert a possible NN violation
5:         end if
6:     end for
7: end for
```

2) User Discrimination: User discrimination implies that an user can't arbitrary be picked to have his/her communication degraded. This sort of discrimination is outlawed in Chile as it's NN regulation points that an ISP *can't interfere or disturb, the rights of any Internet user*.

In our implementation, an user is being discriminated if the latency of its communication is higher than the other users latencies in the network and there is a path between the user and its destination with better latency. An user is given by an unique IP in the network. This is achieved by building the current communication path of the user with its destination and using an FloodLight Northbound API which gives a set of paths between two switches and their latencies.

Algorithm 2 iterates over all OpenFlow rules on all switches (lines 1-5) to identify users in the network and their communication paths. For each user flow path, we acquire its communication latency, utilizing Floodlight's Rest API (line 6). Then, we establish a network latency threshold, which is the sum of the mean of all users communications latency

with its standard deviation (line 7). If an user communication has more latency than the threshold, we look for alternative paths in the network that this user communication could be forwarded to (lines 8-16). Next, utilizing Floodlight's Rest API, we get other possible paths between the user and its destination (line 10) and the latency of this alternative path (line 11). Then, for each of the alternative paths in the Rest API response, we compare it to the user communication path latency identified before (lines 12-14). We consider a NN violation if any of the alternative paths have better latency then the path instantiated for the user communication.

Algorithm 2 User Discrimination Detection

```
1: for all switches do
2:     for all switch rules do
3:         identifies flow paths
4:     end for
5: end for
6: latencies = get(latencies of users flow paths)
7: thresholdLatency = mean(latencies) + std.dv(latencies)
8: for all latency in user flow paths do
9:     if latency > thresholdLatency then
10:        identifies alternative flow paths
11:        get(latencies of alternative flow paths)
12:        if latency > alternative flow paths latencies then
13:            alert a possible NN violation
14:        end if
15:    end if
16: end for
```

Algorithm 3 User Discrimination Detection Without Network Topology Information

```
1: for all switches do
2:     for all switch rules do
3:         identifies user forwarding port load
4:     end for
5: end for
6: for all user forwarding port load do
7:     meanLoad_i = mean(user forwarding port load)_i
8: end for
9: thresholdLoad = mean(meanLoad) + std.dv(meanLoad)
10: for all load in meanLoad do
11:    if load > thresholdLoad then
12:        alert a possible NN violation
13:    end if
14: end for
```

In Algorithm 2 we based User Discrimination detection in user communication latencies and alternative flow paths with less latency for each user. Without having network topology information, we can't measure users communication latencies, so, we try to infer it, based on the switch ports the user communication utilizes and their load.

To detect an User Discrimination without topology information, Algorithm 3 iterates over all OpenFlow rules on all switches (lines 1-5) to identify the load in the switch ports of the user communication, based only in the information provided by the network flow tables. Next, for each user communication we calculate the mean of the load on its communication path, represented in Algorithm 3 by *meanLoad* (line 7). We then define the reference threshold in this algorithm as the mean of all *meanLoad* items plus its standard deviation, as in Algorithm 2 (line 9). Again, as we don't have network topology information in this algorithm, we can't determine alternative paths for the user communication, so, we infer that any load in users communications over the threshold established in a NN violation.

3) Application/Service Discrimination: Similarly to User Discrimination, Application/Service discrimination means that an application or service can't arbitrary be degraded. FCC bans application/Service discrimination by preventing ISPs *throttling* of legal services or applications, while BEREC forbids it by stating that ISPs must not *slow down, alter, restrict, interfere with, degrade and discrimine between specific content, applications or services*.

An application or a service is being discriminated if two applications/services destined to the same user are being forwarded through different paths in the same switch and one of these paths has a worse latency compared to the overall communication latency of the network and there is a path between the application source and its destination with better latency. An unique application/service is given by an unique ethernet type OpenFlow field.

Algorithm 4 Application/Service Discrimination Detection

```
1: for all switches do
2:     for all switch rules do
3:         identifies application forwarding
4:     end for
5:     for all application forwarded to same destination do
6:         get application path latency
7:         identifies alternative flow paths
8:         if application latency >
             alternative flow path latency then
9:             alert a possible NN violation
10:        end if
11:    end for
12: end for
```

Algorithm 4 iterates over all OpenFlow rules (lines 2-3) on all switches (lines 1-12) to identify the forwarding paths of all applications to a destination user in the network (line 3). This is a achieved by creating a triplet *[destination, ethernet type, forwarding port]*. Then, for each application forwarded to the same destination (lines 5-11), we utilize Floodlight's Rest API, as in Algorithm 2, to get the latency of the path in which that application is being forwarded (line 6), based on the *forwarding port* part of the triplet and the latency of alternative paths to the destination (line 7). If any alternative

flow path to the destination has better latency then the path instantiated to the application, there is a NN violation.

4) Paid Prioritization: Paid prioritization is explicitly proscribed in FCC's NN regulation. OpenFlow also has the *priority* field which represents the priority level of a flow entry. Flows with more priority than others are being prioritized. As *ISPANN* does not have the information if this prioritization is paid or not, *ISPANN* just discloses this prioritization to the network operator as an alarm.

Algorithm 5 Paid Prioritization Detection

```
1:  for all switches do
2:      maxPriority = 0
3:      for all switch rules do
4:          if rulePriority > maxPriority then
5:              maxPriority = rulePriority
6:          end if
7:      end for
8:      for all switch rules do
9:          if maxPriority > rulePriority then
10:             alert a possible NN violation
11:         end if
12:     end for
13: end for
```

In Algorithm 5, we look up all switch rules twice. In every switch (lines 1-13), we iterate over each rule, looking the maximum priority value existing in that switch (lines 2-7). Then, in the second iteration, we verify what rules have less than that maximum priority value (lines 8-12). Those communications which have underprioritized rules are having their neutrality violated (lines 9-11).

V. Evaluation

This section presents *ISPANN* evaluation we performed for this work. We assessed its scalability, measuring the execution time of the algorithms implemented. Then, we compared the three policies from Section III to show NN violation detection differences between them, thus validating the necessity of a system that suits to emerging NN policies. Finally, we performed an accuracy comparison over *ISPANN* violation detections when using different sets of network informations, associating these accuracy results to *ISPANN* execution performance requirements.

ISPANN was evaluated using Mininet [17], a network emulator which implements OpenFlow based networks. At Mininet we instantiated Epoch's network topology, an US' ISP. Epoch was chosen for its simple topology, enabling the virtual machine Mininet was running to handle more users in the network and, consequently, to have a larger evaluation scenario. Epoch's network topology was acquired from a database called topology-zoo [18], an Australian project from Adelaide University, which gathers a large set of ISP topologies from around the world. Mininet runs in a 4 GB RAM virtual machine with Ubuntu as it's operational system. Floodlight and *ISPANN* both run in another virtual machine with 1 GB RAM and with Ubuntu as well.

We introduced n users in Epoch's network, choosing randomly which switch to add each host. We varied n in bases of 2, from 128 to 640, that was the maximum number of users Mininet virtual machine could handle. In addition, outside Epoch's network were added two hosts, one streaming a video and another containing a HTTP server. Hosts in Epoch's network access one of the servers in a 3:7 proportion, making 30% of the network traffic be HTTP and 70% be video streaming. The whole scenario is represented in Figure 3. Then, we randomly selected 10% of the flows and changed them, to insert NN violations in these flows (for example, change a forwarding flow to a drop one). For all tests we collected 30 samples of each data and obtained a confidence interval based on a confidence level of 95%.

Firstly, we studied the relation between the number of users in the network and the time *ISPANN* takes to process network violations. As Epoch is a ISP from US, we utilized FCC's policy in this test. The results for this test are shown in Figure

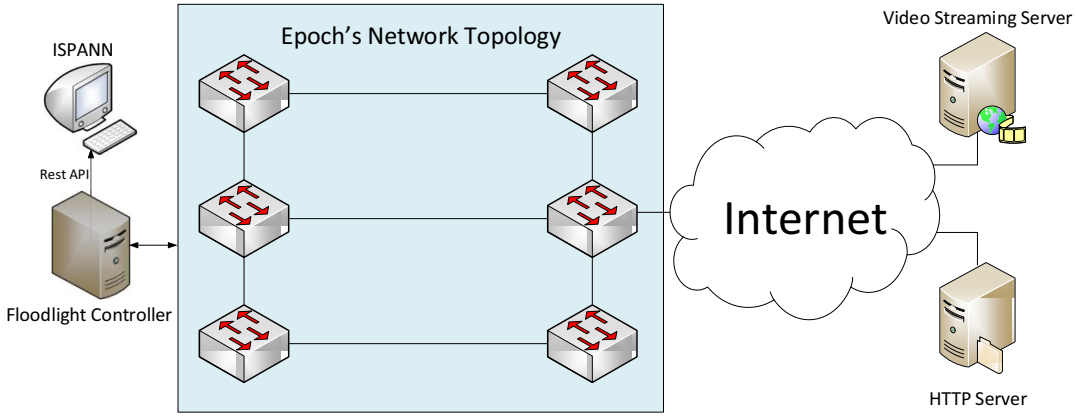

Fig. 3. Epoch's network topology and our evaluation scenario

4. As we can observe, Application/Service Discrimination processing is substantially higher then Blocking Detection and Paid Prioritization detection. For instance, when there is 640 users in the network, Application/Service Discrimination is responsible for 71% of *ISPANN* processing time. This huge difference is due the multiple information Application/Service Discrimination has to get from the network by the Rest API, in contrast to Blocking Detection and Paid Prioritization, that only need flow table informations. For instance, Blocking Detection executes for 1.04 and 1.42 seconds for 128 and 640 cases consecutively, while Paid Prioritization executes for 1.05 and 1.17 seconds and Application/Service Discrimination executes for 2.93 and 6.95 seconds in the same cases.

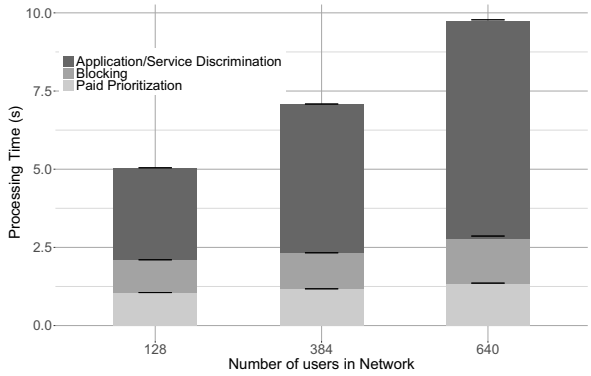

Fig. 4. *ISPANN* temporal performance with number of flows variance for FCC policy

Our next test shows how FCC, BEREC and the Chilean policies differ on *ISPANN*. This test has been executed with 512 users in Epoch's network and the results are presented in Figure 5. As FCC, BEREC and Chile consider blocking as a NN violation, we can correlate the results with the other violation classes. For example, the difference between FCC and BEREC is that FCC executes Paid Prioritization Detection. So, the 7 violations FCC detects more than BEREC, and the 1.5 second it took for this detection, is due to the Paid Prioritization Detection. In addition, the difference between Chilean policy and BEREC is that BEREC realizes Application/Service Discrimination Detection and Chile performs User Discrimination Detection. These differences results in 178 more violations detected by the chilean policy, even though BEREC processes for 2 seconds more.

These differences in each algorithm detection is explained by the scenario chosen to this evaluation. As there is 512 users in the network and only 2 applications, detections are skewed to user-based detections. On one hand, User Discrimination Detection, which focus on user communications, detected 186 violations, and Blocking Detection, which is general for both users and applications, detected 315 violations. On the other hand, Paid Prioritization and Application/Service Discrimination, which focus on applications, detected almost no violations (7 and 8 violations respectively).

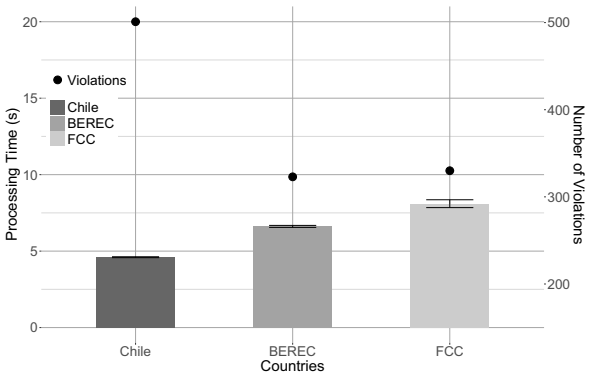

Fig. 5. Comparison of violation detections between countries

Finally, we run both versions of User Discrimination Detection to verify how the number of network informations used impacts NN violation detection. We assumed Algorithm 2 presented in Section IV as a baseline and compared Algorithm 3 time performance and accuracy results to it. In this test, accuracy means the violations that Algorithm 3 detected equally to Algorithm 2. So, in the users communications that Algorithm 2 and 3 detected NN violations there is a true positive, and when both algorithms do not detect violations there is a true negative. When Algorithm 2 detects a NN violation and Algorithm 3 do not, there is a false positive. Moreover, when Algorithm 2 do not detect a NN violation but Algorithm 3 detects one, there is a false positive. Accuracy is calculated summing true positives and true negatives and dividing this sum by the number of user communications.

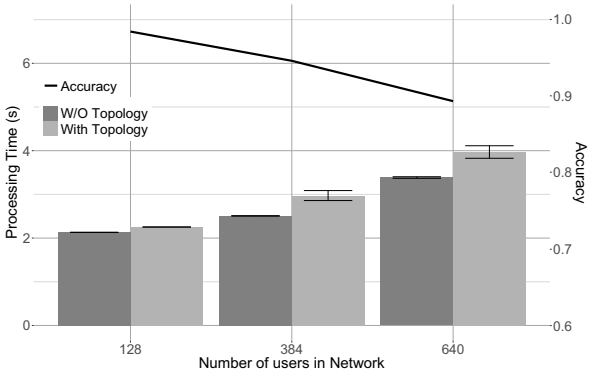

Fig. 6. Temporal and accuracy comparison of User Discrimination Detection using only flow table informations *versus* flow table and topology information

Figure 6 shows that, on the one hand, utilizing more network informations, in this case topology informations, increases NN violation detections processing requirements. With 128 users in the network, the time difference between both algorithms is 0.08 seconds and with 640 users it is 0.6 seconds. So, with 4 times more users in the network, the time performance difference between both algorithms increases 7.5 times. On the other hand, as the network grows, utilizing less network informations decreases detections accuracy. Again, at 128

users in the network Algorithm 3 accuracy is 98% and with 640 users its accuracy drops down to 89%.

Having more network informations makes NN violation detection algorithms less susceptible to false positives and false negatives. Ideally, algorithms implemented in *ISPANN* should test the maximum number of network informations it can. This number is limited by the managing protocol the ISP uses and the frequency the network operator would execute *ISPANN*. For example, if a network operator often applies patches to its devices configurations, *ISPANN* should be executed in the same frequency. If this frequency is high enough, the algorithms implemented should be optimized to use a limited set of network informations.

With these evaluations presented, we conclude this section by commenting the results obtained. The first test, the execution time measurement, indicates us that NN violation detection algorithms implemented in *ISPANN* should be optimized, as they are executed in large scenarios, with many users and applications. Our second test demonstrates the need to take NN policies into account when detecting NN violations. Even at the same scenario, different NN policies produces different violation detection results. Our final test indicates how NN violation detection algorithms should be implemented, in terms of network informations used in them, in *ISPANN*. NN violations detections should test more or less network informations, and consequently be more or less accurate, depending on network operators performance requirements.

VI. Conclusion and Future Work

In this work we presented *ISPANN*, a system that audits an ISP network and detects violations to NN legislations of a given country. The system was modeled and implemented to be generic and easily adaptable to different government existing policies. Our major contribution with *ISPANN* is to bring NN violation detection literature more in line with the political-economical matter. Differently from prior works, that assume any sort of traffic differentiation as NN violations, we assume that NN violations are infringements to countries NN policies. In addition, to the best of our knowledge, our work is the first one which explores NN violation detection at the network operator vantage point.

Our results show that, as different countries have different NN policies, detecting violations to these policies depend on different network features. In this sense, this detections have different time demands and the number of violations in a network may vary. Another important point to raise is that, the accuracy in a violation detection depends on the network features an operator can manage. On the one hand, our system has a worse accuracy when dealing with OpenFlow switch forwarding tables only, when compared to forwarding tables and network topology informations. On the other hand, the more network features the system has to deal with, the more time consuming the detection becomes.

In a future work, we will propose a policy language for NN violation definition and change the *Detection Description Interface* to handle this language. The objective of this policy language is to help non-network specialist legislators with these violation definitions.

Acknowledgment

We would like to thank the Coordination of Higher Education and Graduate Training (CAPES) for the financial support that made this research possible.

References

[1] B. van Schewick and D. Farber, "Point/counterpoint: Network neutrality nuances," *Commun. ACM*, vol. 52, no. 2, pp. 31–37, Feb. 2009. [Online]. Available: http://doi.acm.org/10.1145/1461928.1461942

[2] H. Habibi Gharakheili, A. Vishwanath, and V. Sivaraman, "Perspectives on net neutrality and internet fast-lanes," *SIGCOMM Comput. Commun. Rev.*, vol. 46, no. 1, pp. 64–69, Jan. 2016. [Online]. Available: http://doi.acm.org/10.1145/2875951.2875962

[3] 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 *Proceedings of the 2015 ACM Conference on Internet Measurement Conference*, ser. IMC '15. New York, NY, USA: ACM, 2015, pp. 239–251. [Online]. Available: http://doi.acm.org/10.1145/2815675.2815691

[4] J. Bustos-Jiménez and C. Fuenzalida, "All packets are equal, but some are more equal than others," in *Proceedings of the Latin America Networking Conference on LANC 2014*, ser. LANC '14. New York, NY, USA: ACM, 2014, pp. 5:1–5:8. [Online]. Available: http://doi.acm.org/10.1145/2684083.2684088

[5] D. Li, F. Tian, M. Zhu, L. Wang, and L. Sun, "A novel framework for analysis of global network neutrality based on packet loss rate," in *2015 International Conference on Cloud Computing and Big Data (CCBD)*, Nov 2015, pp. 297–304.

[6] H. H. Gharakheili, A. Vishwanath, and V. Sivaraman, "Pricing user-sanctioned dynamic fast-lanes driven by content providers," in *2015 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, April 2015, pp. 528–533.

[7] M. Dischinger, A. Mislove, A. Haeberlen, and K. P. Gummadi, "Detecting bittorrent blocking," in *Proceedings of the 8th ACM SIGCOMM Conference on Internet Measurement*, ser. IMC '08. New York, NY, USA: ACM, 2008, pp. 3–8. [Online]. Available: http://doi.acm.org/10.1145/1452520.1452523

[8] Sandvine. Global internet phenomena report. Available at https://www.sandvine.com/trends/global-internet-phenomena/.

[9] M. B. Tariq, M. Motiwala, N. Feamster, and M. Ammar, "Detecting network neutrality violations with causal inference," in *Proceedings of the 5th International Conference on Emerging Networking Experiments and Technologies*, ser. CoNEXT '09. New York, NY, USA: ACM, 2009, pp. 289–300. [Online]. Available: http://doi.acm.org/10.1145/1658939.1658972

[10] J. C. D. Martin and A. Glorioso, "The neubot project: A collaborative approach to measuring internet neutrality," in *2008 IEEE International Symposium on Technology and Society*, June 2008, pp. 1–4.

[11] Z. Zhang, O. Mara, and K. Argyraki, "Network neutrality inference," in *Proceedings of the 2014 ACM Conference on SIGCOMM*, ser. SIGCOMM '14. New York, NY, USA: ACM, 2014, pp. 63–74. [Online]. Available: http://doi.acm.org/10.1145/2619239.2626308

[12] Congreso Nacional de Chile. Chilean network neutrality law. Available at https://goo.gl/3xgkGv.

[13] Federal Communications Commission. Open internet. Available at https://www.fcc.gov/general/open-internet.

[14] Body of European Regulators for Electronic Communications. BEREC guidelines on the implementation by national regulators of european net neutrality rules. Available at https://goo.gl/8NID1G.

[15] J. A. Wickboldt, W. P. D. Jesus, P. H. Isolani, C. B. Both, J. Rochol, and L. Z. Granville, "Software-defined networking: management requirements and challenges," *IEEE Communications Magazine*, vol. 53, no. 1, pp. 278–285, January 2015.

[16] Floodlight Controller. Available at http://www.projectfloodlight.org/.

[17] B. Lantz and B. Heller. Mininet. Available at http://mininet.org/.

[18] S. Knight, H. Nguyen, N. Falkner, R. Bowden, and M. Roughan, "The internet topology zoo," *Selected Areas in Communications, IEEE Journal on*, vol. 29, no. 9, pp. 1765 –1775, october 2011.