UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

VINÍCIUS GARCEZ SCHAURICH

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

Thesis presented in partial fulfillment
of the requirements for the degree of
Master of Computer Science

Advisor: Prof. Dr. Lisandro Zambenedetti
Granville

Porto Alegre
November 2018

CIP — CATALOGING-IN-PUBLICATION

Schaurich, Vinícius Garcez

ISPANN: A policy-based ISP Auditor for Network Neutrality violation detection / Vinícius Garcez Schaurich. – Porto Alegre: PPGC da UFRGS, 2018.

64 f.: il.

Thesis (Master) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR–RS, 2018. Advisor: Lisandro Zambenedetti Granville.

1. Network Neutrality. 2. Policy-based Management. I. Granville, Lisandro Zambenedetti. II. Título.

“One change always leaves the way open
for the establishment of others.”
— NICCOLO MACHIAVELLI

ACKNOWLEDGEMENTS

No one achieves anything alone and I would like to thank everyone that contributed for this accomplishment. Starting with my family, that was as supportive as they could, with every dark hour I passed in these two and a half years. My father Wilmar Flores Schaurich, my stepmother Valquiria Lazari and my grandmother Ivone Correa Flores for the *in loco* support; and my sisters Aleteya Garcez Schaurich and Sabrina Garcez Schaurich Martins being so supportive as well, even from far away.

In addition to my family, my work site partners are a huge part of this accomplishment, as I've been at UFRGS 5 times per week (mostly) in this last two years. Thanks to my Advisor Lisandro Zambenedetti Granville and the PhD Student Márcio Barbosa de Carvalho, that was the co-author in my published paper; I wouldn't really have a paper by now if it wasn't by you. Thanks to the people that have been there in the lab with me this whole time, mixing work with *zoeiras*: Bondan, Anderson, Zaca, Serjo, Bardini, Lucas, Nadal, Bob, Carlos, Ariel, Schimuneck, Tocaio, Riquito, Purinho and any other student that came and gone in this time in labs 210 and 212. I will stop it here to not make this paragraph too large in contrast to the previous one. Hope you all keep publishing (much more then me, obviously).

But not only of hard work a work is done. My time at Porto Alegre made me meet a lot of special people that I will carry with me forever. Thank you Sogev, Vaninha, Freitas, Muratore, Manu and Gui for all the parties, hamburgers at midnight, alcohol we drank and DotA matches we played. Thanks to Gentz, Souto, Mini, Aléxi, Nathan, Rudi, Elissa, Artur, Leti and Alegretti for the shorter but still important time we spent together. Thank you all for your friendship. Hope to see you again soon. And here in Manaus, thanks to my childhood friends, that were with me for almost 20 years already. Thank you Ruan, André, Paulo, Rachel, Vera, Maria and Min.

ABSTRACT

Network Neutrality is a controversial topic that often comes back to the society spotlight when some political shakeup occurs. 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. This thesis considers violations directly from governments legislators Network Neutrality rules. In this sense, *ISPANN* is proposed, a system which takes as input countries' Network Neutrality rules and audits an ISP network, identifying possible Network Neutrality violations. No other work proposes Network Neutrality violation detection in the ISP operator vantage point, to the best of the authors knowledge. In the evaluation conducted, an SDN based ISP network is assumed and Network Neutrality violations are verified based on OpenFlow switches flow tables and network's informations.

Keywords: Network Neutrality. Policy-based Management.

ISPANN: Um auditor de ISP baseado em políticas para detecção de violações de Neutralidade de Rede

RESUMO

Neutralidade de Rede é um tema controverso que costuma voltar à discussão na sociedade quando alguma reviravolta política acontece. Vários trabalhos propõem a detecção de diferenciações de tráfego, medindo recursos de rede no ponto de vista do usuário final, o que esses autores julgam como violações de Neutralidade de rede. Entretanto, esses trabalhos tendem a negligenciar a existência das políticas de Neutralidade de Rede estabelecidas nos países onde são testados. Alguns países consideram casos específicos de diferenciações de tráfico como violações de Neutralidade de Rede que não são tão relacionadas aos trabalhos anteriores. Essa tese considera violações diretamente às regras de Neutralidade de Rede estabelecidas pelos legisladores governamentais. Nesse sentido, *ISPANN* é proposto, um sistema onde são introduzidas as políticas governamentais de Neutralidade de Rede de determinado país e o qual audita uma rede de um ISP, identificando possíveis violações de Neutralidade de Rede. Nenhum trabalho anterior propõe a detecção de violações de Neutralidade de Rede do ponto de vista do ISP, até onde o autor sabe. Para a avaliação do sistema, foi assumido um ISP que se baseia numa rede SDN e as violações de Neutralidade de Rede são verificadas baseando-se nas tabelas de fluxo dos *switches* OpenFlow e em informações da rede em geral.

Palavras-chave: Neutralidade de Rede, Gerenciamento Baseado em Políticas.

LIST OF ABBREVIATIONS AND ACRONYMS

AMJHR	Argentina Ministry of Justice and Human Rights
API	Application Program Interface
BEREC	Body of European Regulators for Electronic Communications
CC	Congress of Colombia
CP	Content Provider
CRB	Chamber of Representatives of Belgium
DoS	Denial of Service
FAS	Federal Antimonopoly Service
FCC	Federal Communications Commission
FNA	French National Assembly
HTTP	HyperText Transfer Protocol
IETF	Internet Engineering Task Force
IP	Internet Protocol
ISP	Internet Service Provider
KN	Kingdom of Netherlands
MCTS	Mexican Communications And Transports Secretary
NCA	Norwegian Communications Authority
NCB	National Congress of Brazil
NCC	National Congress of Chile
NN	Network Neutrality
QoE	Quality of Experience
QoS	Quality of Service
SDN	Software-Defined Networking
SLA	Service Level Agreement

SR	Slovenia Republic
TRAI	Telecom Regulatory Authority of India
URL	Uniform Resource Locator
US	United States
XML	Extensible Markup Language

LIST OF FIGURES

LIST OF TABLES

CONTENTS

1 INTRODUCTION

Network Neutrality (NN) has been the source of huge debate between the academic community, legislators, and society in general. NN is, overall, defined as equality on the Internet access; *i.e.*, traffic of different sources, destinies, or applications, should not be treated differently by ISPs (GARRETT et al., 2018). Accordingly, differentiations provoked by ISPs on their network should be treated as network discriminations. 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 (LING et al., 2010). As an example, a CP might pay ISPs to have priority on their network, thus providing a better service than a second CP and, consequently, having service advantages over it. Therefore, legislators should state which actions are allowed to be taken over the users traffic by ISPs. On the other hand, NN opponents claim that regulations reduce ISPs incentive to enhance their service and make innovative technologies deployment more difficult. So, the market nature of self regulation by competition should be preserved (SCHEWICK; FARBER, 2009).

Regardless of a side taken in this debate, NN is an extremely ambiguous topic (BYUN; LEE, 2013). Given the definition above, some questions can be asked: what is network discrimination, what network configurations implemented by ISPs can be considered discriminations and, if a network configuration is considered a discrimination, how one would detect this discrimination to verify its existence? Moreover, when conceiving the term, Tim Wu raised the question of whether whom would formalize a consensus about NN and what discrimination in the network is: the academia, countries legislators or IETF. By this date, there is still no IETF standard defining guidelines on NN. In contrast, authors and legislators have been proposing new works and policies, regarding network discrimination detection and avoidance, since then.

These policies that countries' legislators establish become rules enforced over ISPs, which restrain their operations (GHARAKHEILI; VISHWANATH; SIVARAMAN, 2016). In this manner, NN rules have substantial technical consequences in ISP's networks, despite being a political-economical oriented subject. For instance, network components must be configured to avoid violating these rules, and, if a misconfiguration is detected in an ISP operation, this ISP may suffer prosecution. In order to try to infer NN violations committed by an ISP and help Internet users suffering with this sort of violations, several studies propose techniques to identify network discrimination (often referred as traffic differentiations) from the end-user vantage point (KAKHKI et al., 2015)(BUSTOS-JIMÉNEZ; FUENZALIDA, 2014)(LI et al., 2015). These violations inferences are based on users traffic statistics measurements, such as packet loss, jitter, and latency.

While the academia focused on these traffic differentiation detection works, the NN ambiguity problem got worse. These studies tend to be disconnected from the NN regulation of the countries they take place and consider traffic differentiations as NN violations. Authors often base their NN definition in their own point of view on the topic and cite an event that has been debated by the society (for example, Comcast's shaping (GHARAKHEILI; VISHWANATH; SIVARAMAN, 2015)) to justify their NN violation detections, as if the usage of a QoS tool like traffic shaping was always a NN violation. Each author assuming a different definition to NN and basing its work on something that may not be a NN violation at all turns the NN research into a dubious topic.

In addition, a Chilean NN violation detection tool called Adkintun (BUSTOS-JIMÉNEZ; FUENZALIDA, 2014) has been reported to support user rights in legal complaints against Chilean ISPs. In the same manner as prior works, Adkintun performs traffic differentiation tests on the user vantage point of the network. However, there is no such a violation detection study to help network operators to verify that their network meets government's NN rules. Network operators that want their network complied to these NN rules must configure their network by themselves. Networks being such complex environments and NN rules not being straightforwardly translated into network devices configurations make violation detections executed solely by humans infeasible.

In this thesis, *ISPANN* is presented, a system that audits whether a network is conformed to the NN rules stated by the country where the ISP network is located. *ISPANN* can be used in two different manners: 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, this thesis brings technical literature more in line to the existing political-economical matter.

ISPANN has two major objectives: reduce the ambiguity generated by NN violation detection works and give a supplementary violation detection tool for network operators. Firstly, *ISPANN* intends to reduce NN violation detection ambiguity by removing the author point of view of the NN definition. As countries legislations are the NN definitions considered in *ISPANN* a layer of assumptions is removed for the author and is demanded from the already defined NN policies. Secondly, by auditing ISP networks, *ISPANN* is designed to advise network operators of any network configuration they may have have applied that breaks a NN legislation. Combining the solutions for these two objectives, *ISPANN* provides a platform for network operators to have a better understanding of side effects their network configurations introduce in the network in terms of NN issues.

Given the introduction above, the rest of this thesis is organized as follows. Chapter 2 explains the concept of NN, presents what are NN legislations and refers related works to *ISPANN*; *i.e.* NN violation detection works. Next, Chapter 3 presents the objectives of the proposed system and *ISPANN*'s architecture, explaining each of the proposed system modules. Then, Chapter 4 shows the environment surrounding *ISPANN* implementation assumed for this thesis, while Chapter 5 presents this implementation *per se*. With the prototype presented in the prior chapter, Chapter 6 describes the evaluation performed over it, gathering performance and violation detection comparison results. Finally, Chapter 7 concludes this thesis with the academic progress this thesis has provided.

2 BACKGROUND AND RELATED WORK

NN has been in the spotlight of the media and sees a huge debate between the academia, legislators and the society in general. This debate raises from one major problem in the Internet: the computational power of the network is limited. Thus, for it to operate properly, network managers, ISPs and backbone providers must configure their networks to satisfy the requirements of the services running over their networks and the users accessing these services; *i.e.* QoS and QoE configurations.

If, on one hand, network managers must configure their network to properly provide an ambient of content and information exchange, on the other hand, network managers can intentionally configure their network to discriminate content and have more economical returns over their infrastructure. As an example, a CP might pay ISPs to have priority on their network, therefore, having service advantages over a second CP. Literature says that this behavior of a given network manager causes an unfair competition and makes the ambient provided by this ISP biased.

With the possibility of unfair competition in the Internet both legislators and the academia started a task force to define what NN violations are and how to detect them. Legislators mainly propose NN rules to be enforced on service providers of the country they legislate, while the academia often cite an event where a given traffic differentiation technique was widely used by network managers to claim that NN violations occur, thus must be detected, and justify their NN violation detection tool.

This chapter presents the state of the art and bases the need of further investigation in NN. Firstly, Section 2.1 gives an overview on NN, showing different definitions of the theme and discussing whether these definitions are feasible in complex networks. Then, Section 2.2 presents a set of NN policies aiming at the discrepancy between different legislations. Finally, Section 2.3 introduces various NN violation detection works, focusing on the gap between NN literature and the policies shown in Section 2.2 and discusses why NN violation detection works should be in line with the policy matter.

2.1 Network Neutrality

NN is a term coined by Tim Wu in 2003 (WU, 2003) that often comes back to the society spotlight when a political turnaround occurs[1] or when an unethical ISP operation

[1] <https://cnnmon.ie/2MC1s81>

gets public[2]. 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 led researchers to study how ISPs were treating applications differently at their network (DISCHINGER et al., 2008). More recently, Netflix has been reported to be the application which occupies most of the downstream traffic at the United States (SANDVINE, 2016), becoming the main target of ISP's throttling and traffic differentiation (KAKHKI et al., 2015).

In his survey, Wu defines NN as *an Internet that does not favor one application over others*, *i.e.*, a neutral network is an end-to-end communication platform where ISPs don't disrupt an arbitrary transmission with no good reason. In this sense, he criticizes the open-access concept that existed in that time, where proponents argued that networks should be neutral as among all applications, and favors *broadband discrimination*, which assumes that there are justified differentiations in applications treatment, as the goal to address NN. The NN problem would lie in finding a consensus on the limit of these differentiations; on the judgment whether they are justified or not.

While creating the concept of NN, Tim Wu concluded his survey with the following question: who should propose a consensus over NN? Should it be an independent group like IETF, academic researchers or country policy legislators? On the one hand, by this date, 15 years later, IETF did not propose an RFC or a Draft over the NN issue. On the other hand, the academic community have been proposing works in several areas over the NN issue (that may vary from the political, economical and technical scope), and many countries legislators defined policies regulating the actions of ISPs in these countries, regarding NN.

In any case, there is no consensus over the NN issue and this openness on the NN definition leads to some ambiguous results. In the regulatory part, legislators may define conflicting policies within each other, making the NN issue diverge in some aspect from country to country. These problems are better explained in the section 2.2. In the academic part, this problem is broader, since the scope of the works are not limited to a political point of view, as in the regulatory part. Technical works, beside having to find a technical solutions to the problems the NN issue create, also have to assume a political point of view to define what the NN definition means and what would be NN

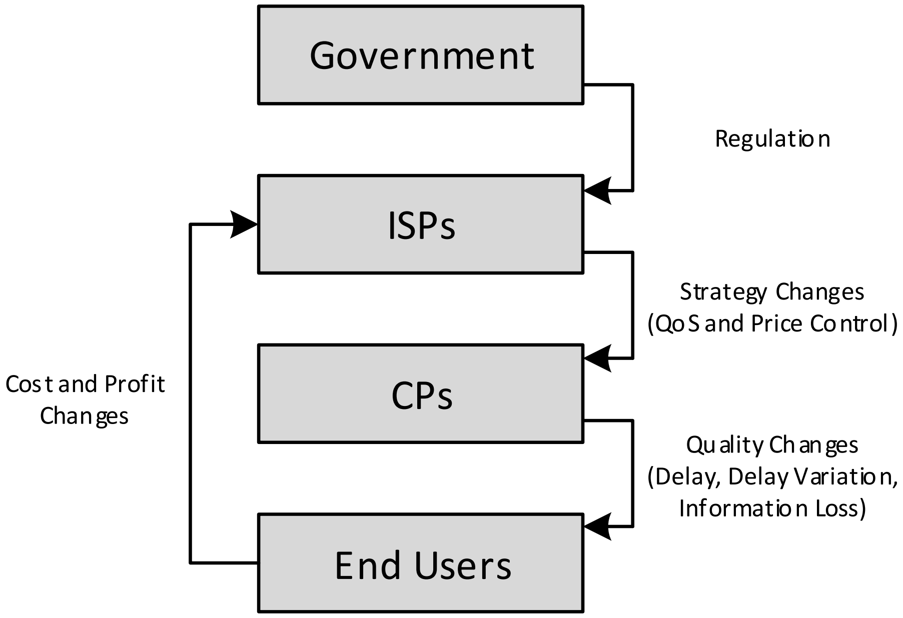
[2] <https://nbcnews.to/2xf0xp4>

violations then. It comes to a point where, with the differences in these assumptions, one cannot compare these works technical results. These technical works and their results are approached in Section 2.3, focusing on NN violation detection works, that is the topic of this thesis.

2.2 Network Neutrality Policies

Governments define policies over NN that ISPs must follow, just as a client's Service Level Agreement (SLA). Figure 2.1 shows how the Internet players interact when under a government NN legislation. Each country's legislator has its own point-of-view and understanding about NN (GHARAKHEILI; VISHWANATH; SIVARAMAN, 2016); and, therefore, policies may diverge in some aspect. For example, zero-rating, the act of not charging the end-user over an specific service, is accepted in countries like Brazil (NCB, 2014), but has limits in Europe with the Body of European Regulator for Electronic Communications' (BEREC) guideline. BEREC allows zero-rating over a group of services of the same type and not only to one service or application specifically. This section overviews the differences on policies from different countries.

Figure 2.1: Players and strategies



Source: (LEE; KIM, 2014)

A large number of countries have defined NN legislations or have discussed the issue and are elaborating one; to cite a few: Argentina (AMJHR, 2014), Belgium (CRB, 2011), Brazil (NCB, 2014), Chile (NCC, 2010), Colombia (CC, 2011), France (FNA, 2011), India (TRAI, 2017), Mexico (MCTS, 2014), Netherlands (KN, 2012), Norway (NCA, 2015), Russia (FAS, 2016), Slovenia (SR, 2012) and US (FCC, 2015). Among these countries, Chile was the first to advocate and propose a NN regulation law, voted unanimously in the National Congress of Chile (NCC) (NCC, 2010). Published in August of 2010, the policy establishes that ISPs *cannot 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, FCC, stated the Open Internet (FCC, 2015), its NN policy. FCC regulates interstate and international communications by radio, television, wire, satellite, and cable in all 50 states, the District of Columbia and U.S. territories. An independent U.S. government agency overseen by Congress, the Commission is the federal agency responsible for implementing and enforcing America's communications law and regulations. The FCC discussion over the Open Internet regulation, that started around 2010 and ended up with its 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 (BEREC, 2016). 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.*

2.3 Network Neutrality Violations Detections

Focusing on 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 (TARIQ et al., 2009), 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 (MARTIN; GLORIOSO, 2008), 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 (LI et al., 2015) 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 (ZHANG; MARA; ARGYRAKI, 2014) developed an algorithm that takes as input a network graph and end-to-end measurements and identifies non-neutral link sequences. Kakhki et. al (KAKHKI et al., 2015) 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 (BUSTOS-JIMÉNEZ; FUENZALIDA, 2014) 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. No study proposes this kind of detection in the network operator vantage point, to assist network operators with the technical issues NN generate. In addition, it is 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.

3 *ISPANN*

In this chapter *ISPANN* is introduced, a system that takes governments NN policies as input and audits an ISP's network, trying to detect NN violations to those rules. Prior to *ISPANN*, no other system was designed to help network managers to detect possible NN violations their network configurations generated in the network or utilized NN policies to perform NN violation detections.

Firstly, Section 3.1 presents and gives a more in depth explanation of the objectives of *ISPANN*, reaffirming prior works ambiguity. Then, the architecture of *ISPANN* is presented in Section 3.2. 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.

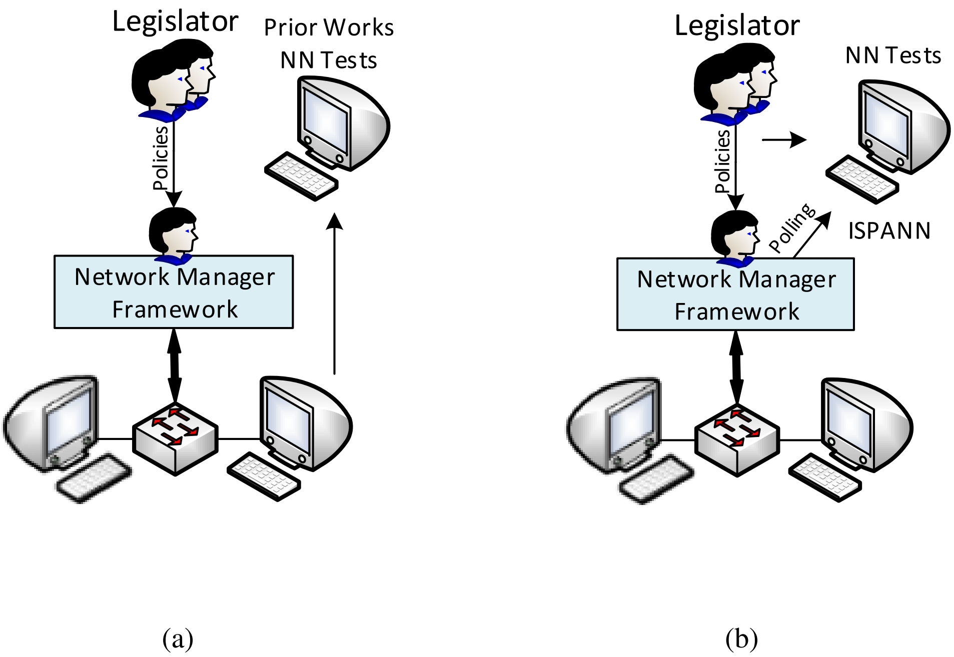
3.1 Objectives

As said in previous chapters, *ISPANN* has two main objectives: reduce prior works on NN violation detection ambiguity and give a system for network operators to test if the configurations they are inputting to their network don't break the NN policies they are subjected to. This section gives a better explanation of why a system focused on these objectives is important while also explaining how *ISPANN* tackles these two issues.

This first objective comes from the fact that, in previous work, authors always assumed a different definition of NN and, consequently, a different definition of NN violations, based on their own point of view of the subject, to be detected in the network. Reducing this ambiguity gives a clearer understanding of the requirements to perform NN violation detections and assists future authors on comparing their works. The second objective revolves around the fact that no other work aims on detecting violations in the network operator vantage point, they all are based on the user vantage point. Network configuration is a recurring network management topic and auditing networks is one the tasks on network configuration, where network managers or systems, as *ISPANN*, seek for network misconfiguration.

Given these two objectives, *ISPANN* intends to reduce NN ambiguity by removing the author point of view over the NN matter. By handing the NN violation definition to legislators policies, authors do not have to assume these definitions and, in this manner, a layer of ambiguity is removed. Thus, legislators policies are assumed as SLAs that are enforced over ISPs: a set of rules that a given ISP network must be configured to be conformed with. NN violation detection can then be compared by performance and accuracy over the same policy, or, in this case, the same SLA, not over different definitions over the NN issue.

Figure 3.1: Comparison between prior works (a) and *ISPANN* (b) NN tests paradigm



(a) (b)

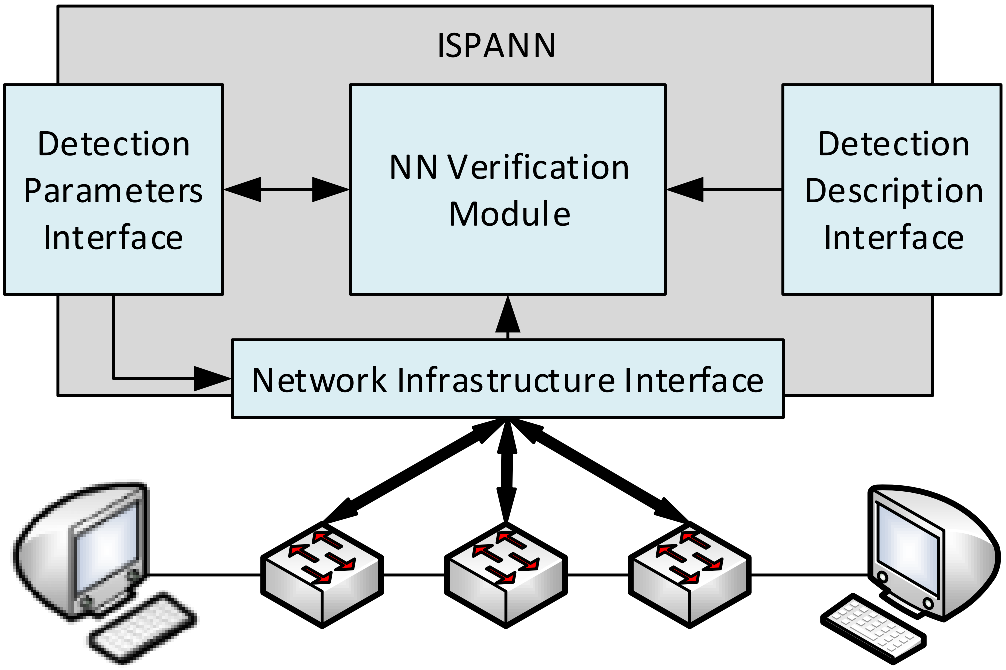
Source: (The Author, 2018)

In addition, *ISPANN* takes advantage of the gap in the academic research, where no other work proposes violation detections in the network operator vantage point, by verifying NN violations in the network operator vantage point. Figure 3.1 shows the difference in paradigms of prior works and *ISPANN*. Although this objective differs from the prior, they both are complementary. As the policies are seen as SLAs for ISPs, ISPs need a way to audit their networks and tell, if they are breaking any rule, how they are doing it. Given the complexity of a network environment, a computational system that assists network managers is needed. *ISPANN* them takes these policies and evaluates if the configurations of the ISP network devices are breaking them.

3.2 Architecture

ISPANN is a NN violation detection tool that aims on auditing ISP networks and verifying if the configurations of the devices of their networks are conformed to the legislations that they are subjected to. It was modeled and implemented to be as generic as possible, fitting to different ISPs' infrastructures and technologies, as well as different NN policies. In this sense, the system is based on four modules: *Detection Parameters Interface*, *Detection Description Interface*, *Network Infrastructure Interface*, and *NN Verification Module*. The Figure 3.2 presents the architecture from *ISPANN*.

Figure 3.2: System architecture


ISPANN
Detection Parameters Interface
NN Verification Module
Detection Description Interface
Network Infrastructure Interface

Source: The Author, 2018

In this design, 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 the information input from network manager and legislators, and tests if any violation to the legislation inputted exists. Each of these modules are better presented below.

3.2.1 Detection Parameters Interface

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. The network management protocol and the country are passed to the *Network Infrastructure Interface*, so it polls network information from the devices properly and gathers relevant data for that country policy; and the country is passed to the *NN Verification Module*, so it knows what NN violation detection algorithm must be executed. A in depth explanation of what are these algorithms is presented in Subsection .

Taking as an example, if an US ISP that bases its network infrastructure in a traditional network, and uses NetConf (ENNS et al., 2011) as its network management protocol, wants to utilize *ISPANN* to test whether its network is breaking NN policies it would input information to the system as follows. The communication protocol informed by the network operator would be NetConf, to be passed to *Network Infrastructure Interface*. In addition, it would input "US" to know if its network is conformed to the legislation it is subjected to. *ISPANN* capabilities to perform different tests, based on different network management protocols and policies, are dependent of the implementations made in the next modules.

3.2.2 Network Infrastructure Interface

Network Infrastructure Interface is the interface between *ISPANN* and the network devices from an ISP infrastructure; and is responsible from polling network information from these devices. As mentioned in Subsection 3.2.1, the *Detection Parameters Interface* 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 queue information, depending on the protocol capabilities, and sends the collected data to the *NN Verification Module*.

3.2.3 NN Verification Module

NN Verification Module is the core part of *ISPANN*, the one that performs NN violation detections, bringing it in line with prior works. It is composed of a set of NN violation detection algorithms and set of links between a country and one or more NN violation detection algorithms. In a best case scenario, legislators would implement the NN violation detection algorithms and link then to the country they are regulating in *ISPANN*, while network operators would just run system based on the country their network operates. These links between a country and a set of NN violation detection algorithms are created in the *Detection Description Interface*, that is presented in the Subsection 3.2.4. So, the same algorithm can be performed as part of the NN violation detection of more than one country, it all depends on the refinement level that operators and legislators require for the violation detections they want performed as well as the depth in which the NN policies are stated in their legislations.

The reuse on NN violation detection algorithm, where a given algorithm is run as part of more than one policy NN violation detection, is a result of the similarity of parts of two NN violation policies. For example, as mentioned in Section 2.2, most NN policies ban communication blocking of legal content. Furthermore, *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.

NN Verification Module receives the network information gathered from the *Network Infrastructure Interface* and the NN policies to be tested from the *Detection Parameters Interface*, defined by country. Given the country received by the *Detection Parameters Interface*, the *NN Verification Module* performs the NN violation detection algorithms related to that country, based on the links explained above.

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 information 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.

3.2.4 Detection Description Interface

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 creates the links between that country and the algorithms that will run in the *NN Verification Module* when the system executes its NN violation detections, as explained in Subsection 3.2.3. So, when a country is input in the *Detection Parameters Interface ISPANN* knows what algorithms to run for that given policy. The country and the algorithms linked to its NN violation detection are stored as part of the *NN Verification Module*.

4 USE CASE ENVIRONMENT

This chapter presents the environment involving *ISPANN* implementation made for this thesis. As it is a generic system, some parameters must be assumed to be input to *ISPANN*; *i.e.*, the network management protocol used a given ISP would use to configure its network devices and the country where the ISP has its network. In this case, it was assumed an ISP that has its network in the United States and bases its network in SDN, consequently, configuring its network with OpenFlow. Also, *ISPANN* being designed to implement NN violation detections directly to legislators policies, three policies were assumed and implemented in this use case, for evaluation purposes.

Given the requirements for an use case assumption to *ISPANN* implementation, this chapter is organized as follows. Section 4.1 presents the network scenario surrounding *ISPANN* in this use case. This section gives an overview of SDN based networks, as well as the emulated ISP network where the system polled the informations for the NN violation detections. Then, Section 4.2 introduces the NN policies assumed for the NN violation detections of *ISPANN*.

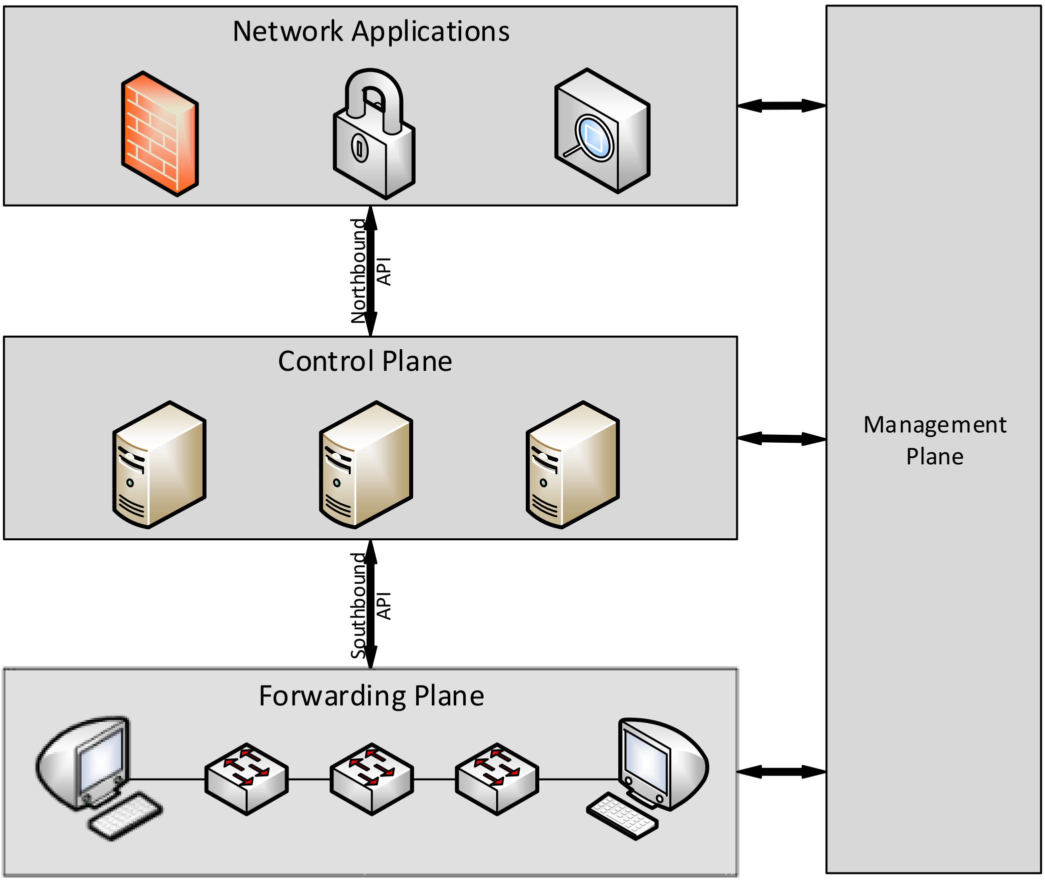
4.1 Network

In this thesis, and for *ISPANN* evaluation, presented in Chapter 6, it is considered an ISP 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 (WICKBOLDT et al., 2015). This approach is achieved by the introduction of a network component called network controller, that coordinates the packet forwarding decisions of the remaining network devices. These sort of decisions were made by the network devices in traditional networks, adding processing time to these devices, thus requiring them to have more processing power.

The control and forwarding plane decoupling makes the network more flexible and facilitates the implementation and development of novel technologies. Developing new technologies in an SDN based network requires a new controller implementation, in contrast to traditional network, where they must be implemented in every network device. This flexibility helps ISPs to improve their network performance, while also reducing complexity and cutting costs of new technologies development, despite the challenges of implementing this paradigm in ISPs networks (BIRK et al., 2016).

The decoupling paradigm of SDN introduces new network planes and interfaces and leads the network to the architecture shown in the Figure 4.1. In addition to the Control and Forwarding planes, SDN based networks can be composed of: Network applications, that are added over the Control Plane, communicating to the network controller; two APIs, one to communicate the Control and the Forwarding planes (Southbound API) and one to communicate the Network applications to the network controller in the Control Plane (Northbound API); and a Management Plane that can manage any of the planes cited previously, depending on the objective of the management tool implemented.

Figure 4.1: SDN architecture



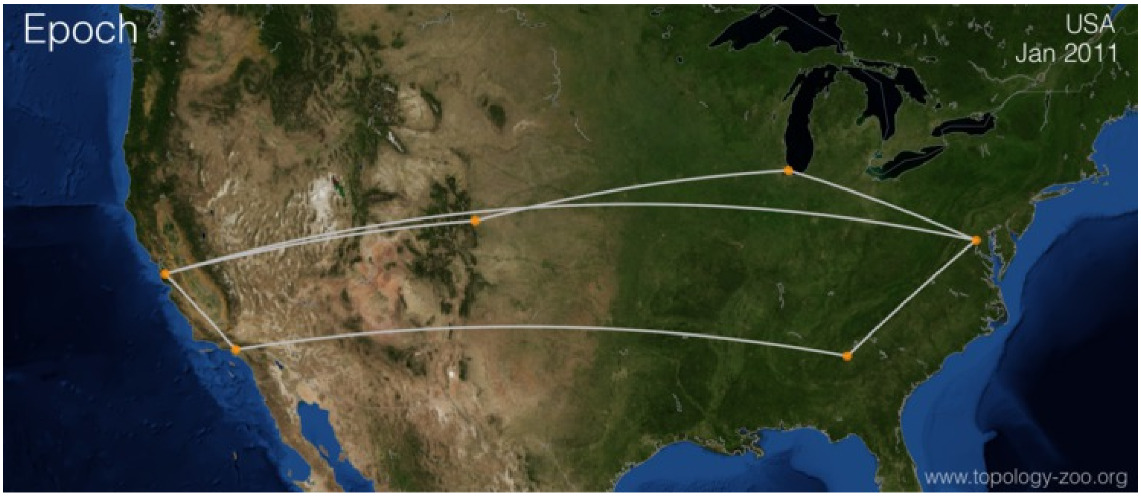
Source: (WICKBOLDT et al., 2015)

For the controller, Floodlight was assumed (FLOODLIGHT, 2012), a network controller implemented in Java which communicates with the network devices via the OpenFlow protocol, its Southbound API. Each OpenFlow based switch is identified by a value named *dpid* and has flow tables, which are table that defines a tuple of four elements: an entry port, a source IP, a destination IP, which are the matching part of the tuple, and an output port. When a packet arrives in an OpenFlow switch, it looks for a matching entry port, source IP and destination IP in its flow table and forwards the packet in the output port of the corresponding match tuple. If no match is found by the switch in its flow table, it requests to the network controller to inform it what should be the port to that packet be forwarded and saves this information on the flow table for further packets input.

Moreover, Floodlight has a Northbound API [1], a Rest API that responds to HTTP requests in the 8080 port of the host where the controller is running. *ISPANN* utilizes Floodlight's Northbound API to acquire the network topology and the OpenFlow switches flow table information in the *Network Infrastructure Interface* that is forwarded to the *NN verification module*, to perform NN violation detections. So, in this scenario, for the system execution, the network operator inputs Northbound API as the protocol that it communicates with its network (in the case of utilizing a Northbound API, the controller needs to be specified as well, that is Floodlight).

In addition to SDN and Floodlight inputs, as the management protocol assumed in this scenario, *ISPANN* needs the country of operation of the ISP, as explained in Subsection 3.2.1. Thus, US was chosen as the country of operation in this scenario, being the country that made the NN topic a worldwide matter, as mentioned in Section 2.2 with its regulatory rules.

Figure 4.2: Epoch's network topology



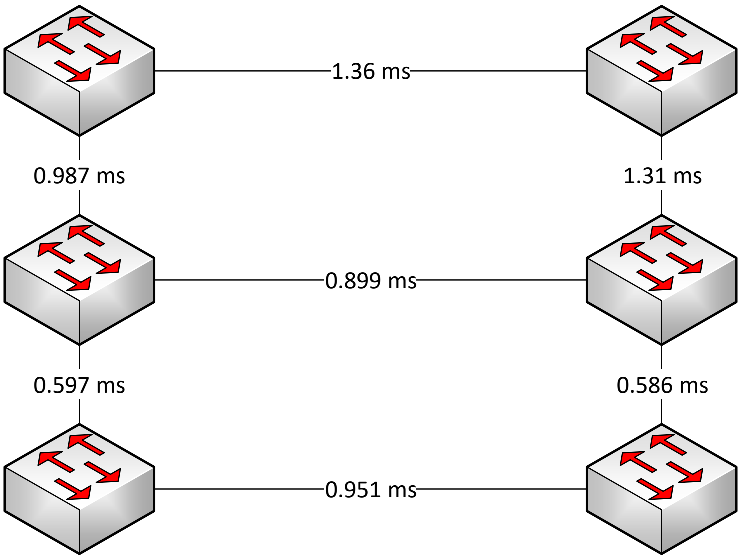
Source: (KNIGHT et al., 2011)

Utilizing a database called topology-zoo (KNIGHT et al., 2011), an Australian project from Adelaide University, which gathers a large set of ISP topologies from around the world, an ISP located in the US was randomly selected. The name of the chosen ISP is Epoch and its topology is presented in the Figure 4.2. This topology was then implemented in Mininet (LANTZ; HELLER, 2011), a SDN network emulator that implements

[1] <https://bit.ly/2NTuGnO>

OpenFlow based network devices. The resulting implementation of Epoch's network in Mininet is shown in Figure 4.3, with the propagating delays from the virtual network links describing each virtual link in Mininet.

Figure 4.3: Epoch's network topology implemented in Mininet and the resulting network virtual links parameters



Source: (The Author, 2018)

Compiling the information in this section, *ISPANN* is implemented over an use case network scenario as follows: a US located ISP, Epoch, runs *ISPANN* to audit their SDN based network for NN violation detections. With this assumption, the system needs to implement algorithms that will detect violations to the US legislation. Section 4.2 overviews the US legislation and how it was understood for *ISPANN* implementation, while further explaining other legislations and their implementations in the system.

4.2 Network Neutrality Policies

A huge part of *ISPANN* is understanding NN policies and implementing NN violation detection algorithms that better describe these policies. The system was projected so legislators implement these algorithms and describe their policies but, for the purpose of evaluation on this thesis, this section presents the policies selected for this evaluation and the understanding over these policies to further describe the algorithms implemented in *ISPANN*, presented in Section 5.

The first NN policy selected to be implemented in *ISPANN* and to perform its evaluation is the FCC's legislation. This NN policy was chosen based on the network scenario presented in Section 4.1, which assumes Epoch as the ISP running the system, an ISP located in the US. As mentioned in Section 2.2, 2015 FCC policy was composed of three bright-line rules: *no blocking, throttling or practice paid prioritization.* These rules are the key of the violation detection that is performed for the FCC policy and, adding the policies cited below, they can be interpreted and translated to NN violation detection algorithms.

Even though it was assumed an US' ISP in the Section 4.1 to base the *ISPANN*'s prototype, it is important to consider more policies in *ISPANN*'s evaluation to verify how different policies produce different NN violation detection results, even in the same network. Thus, two more policies were implemented *ISPANN*: BEREC's guidelines and the NCC's policy, both which were presented in Section 2.2 as well. As discussed, BEREC guidelines states *no blocking, slowing down, alteration, restriction, interference with, degradation and discrimination between specific content, applications or services, or specific categories thereof*, while the NCC policy bans *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.*

Given these three NN policies, some conclusions can be drawn. 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, NCC 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. With these conclusions, four classes of NN violations were conceived: **blocking**, **user discrimination**, **application/service discrimination** and **paid prioritization**. NN violation detection algorithms were implemented in *ISPANN* for each of these classes. These algorithms are presented in Section 5.4.

Blocking violation is the concept of ISP cannot block a legal content in its network. For instance, an ISP blocking a given IP that refers to a content of its market competitors, or that makes anti propaganda of it. **User discrimination** violation means throttling an end-user connection arbitrarily, with no proper reason for it. A recurring form of **user discrimination** done by ISPs is using *traffic shaping* to reduce the bandwidth that a user may utilize for reasons such as usage peak hours on their networks, thus needing to limit their users bandwidth. **Application/Service discrimination** violation is similar to **user discrimination** violation but refers to throttling applications. In this case, *traffic shapping* can be used, but is referring to a determined application that an user is utilizing. This occurred massively in the past with BitTorrent and today is mostly occurring with video streaming services as mentioned in Section 2.1. Lastly, **paid prioritization** violation means the ISP being paid to prioritize a given service in detriment of others. As an example, being paid to give a service a better QoS traffic class or scheduling it in a switch queue with better priority.

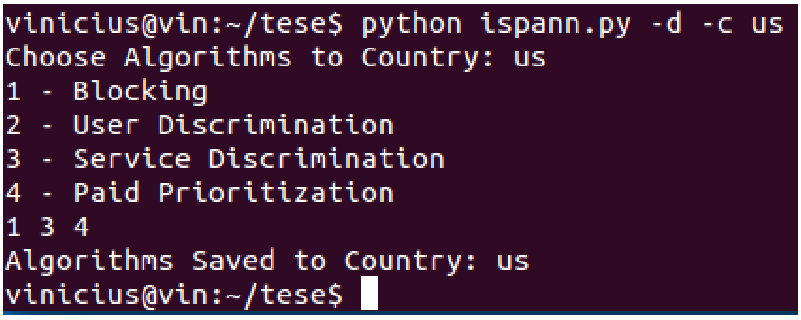
5 *ISPANN* PROTOTYPE

This chapter introduces *ISPANN* Prototype implementation based on the scenario and the policies presented in previous chapter. *ISPANN* Prototype is an application that runs in the Network Application Plane of an SDN-based network, as explained in the Section 4.1. Section 5.1 explains the user interface implemented in the prototype, as well as the parameters permitted and execution modes these parameters run. Then, Section 5.2 presents how the *Network Infrastructure Interface* was implemented and how *ISPANN* Prototype gathers the network information. Finally, sections 5.3 and 5.4 describe the *NN Verification Module*, with Section 5.3 focusing on an overview of the execution of the module, while Section 5.3 shows the algorithms implemented in the module, based on the classes presented in Section 4.2.

5.1 User Interface and Execution Modes

ISPANN Protoype is a command line application implemented in Python. The parameters that can be input in the command line are the following: "-d", "-D" "-c" and "-p". NN violation detection operation is the normal operation mode of the prototype and there is no need to enter a parameter for it, while "-d" and "-D" define secondary system operation modes. "-d" is input when a third party just wants to add a NN violation detection description in the *Detection Parameters Interface* and "-D" is the mode that runs the execution of "-d" and, after this third party describes the NN violation detection, runs the detection described.

Figure 5.1: *ISPANN* Prototype running "-d" mode

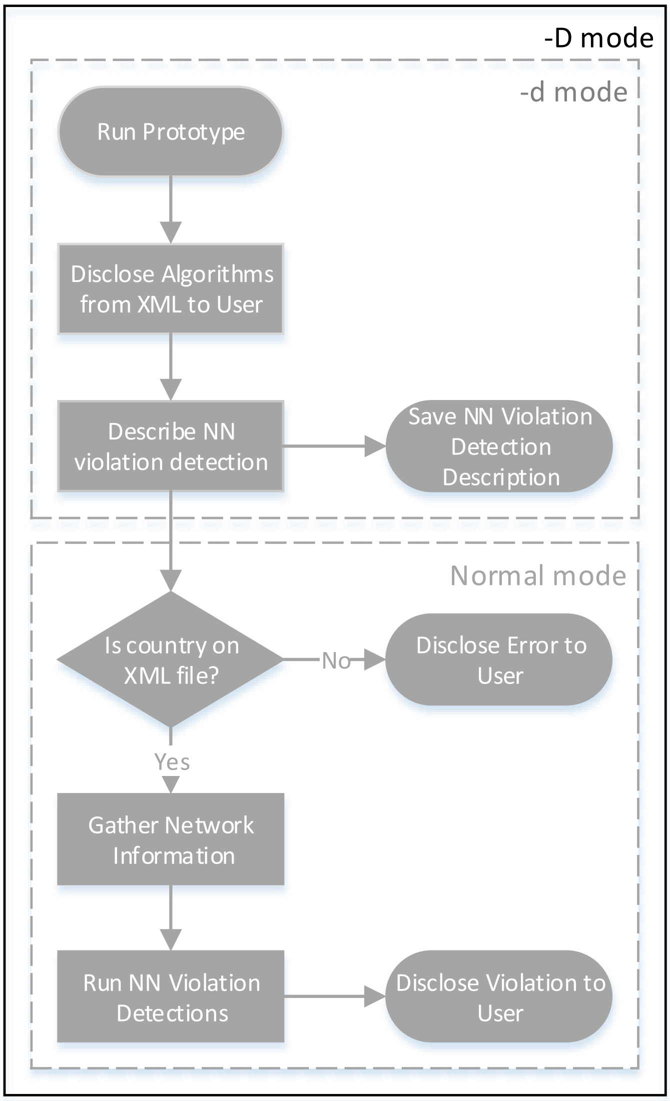


```
vinicius@vin:~/tese$ python ispann.py -d -c us
Choose Algorithms to Country: us
1 - Blocking
2 - User Discrimination
3 - Service Discrimination
4 - Paid Prioritization
1 3 4
Algorithms Saved to Country: us
vinicius@vin:~/tese$
```

Source: (The Author, 2018)

In this context, "-d" and "-D" are void inputs, inputs that are entered without a variable string passed with it, that defines what mode of execution *ISPANN* Prototype

Figure 5.2: *ISPANN* Prototype operation modes


-D mode
-d mode
Run Prototype
Disclose Algorithms from XML to User
Describe NN violation detection
Save NN Violation Detection Description
Normal mode
Is country on XML file?
No
Disclose Error to User
Yes
Gather Network Information
Run NN Violation Detections
Disclose Violation to User

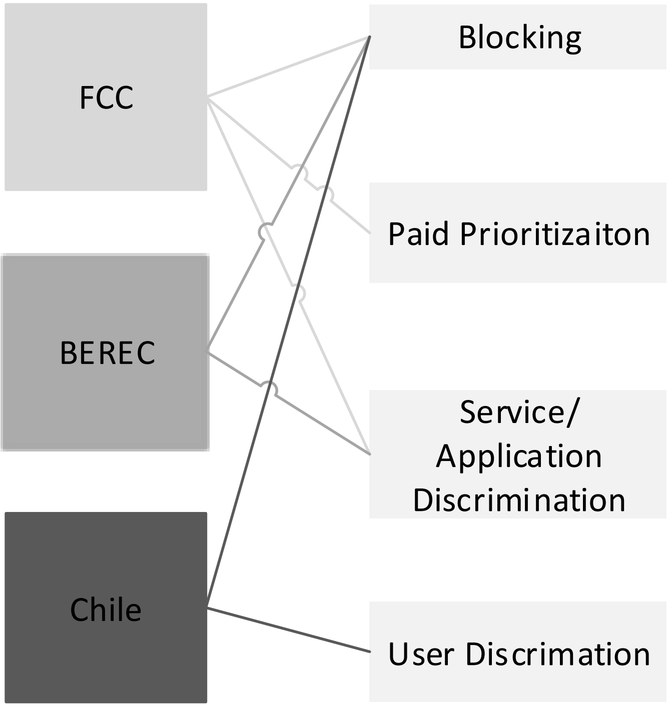
Source: (The Author, 2018)

will run. In addition to them, "-c" and "-p" are inputs that need a variable string entered with them, with "-c" defining the country policy that the prototype will execute and "-p" the type of management protocol it will use to gather the information from the network. As the normal mode needs a country to run NN violation detections and the "-d" is the mode to describe these NN violation detections for a given country, the "-c" parameter is mandatory for any sort of execution, as shown in Figure 5.1. In contrast, the "-p" has no meaning in the "-d" mode, and needs to be input only when *ISPANN* runs NN violation detections. The flow chart in Figure 5.2 presents the three different operation modes of *ISPANN* Prototype.

Thus, the *Detection Parameters Interface* is implemented with the command line and a parser, to execute *ISPANN* Prototype's functions as the users interests, in accordance to the parameters explained above. This parser is implemented with the Python library *argparse* (). In the case where the user enters a description mode ("-d" or "-D"), an UI pops-up, which is the *Detection Description Interface*, listing all algorithms implemented in the prototype. The user then selects the algorithms it wants to run for that given country he entered.

The prototype then saves the selected algorithms in a XML file, which has a list of all countries and their respective algorithms, referring the new country added. The set of algorithms referring to a country in the country list XML file conceive a NN violation detection of that country. The list of all algorithms implemented in the prototype, mentioned above in the description mode case, is another XML file. These XML files are written and are further read in the next modules explained below with the Python library *ElementTree*.

Figure 5.3: Policies, NN violation detection classes and their relations



Source: (The Author, 2018)

As mentioned in the 4.2 four NN violation classes were conceived in the understanding of the NN policies assuming to be implemented in *ISPANN* Prototype. These classes are: **blocking**, **user discrimination**, **application/service discrimination** and **paid prioritization**. The Figure 5.3 shows the connections between the NN violation detection classification and the three policies, which were entered in the "-d" or "-D" mode of the prototype via the *Detection Description Interface*. Each one of these classes, beside user discrimination class that has two corresponding algorithms implemented, has a respective NN violation detection algorithm. The algorithms implemented in *ISPANN* Prototype are shown in section 5.4.

5.2 Gathering Information From the Network

The country and the management protocol string variables are then passed to the *Network Infrastructure Interface*, as explained in section 3.2.2. With the scenario assumed in Section 4.1, the user inputs a tuple *("SDN", "Floodlight")* as the "-p" parameter. Note that for an SDN based network, the system also need the network controller used. Then, *ISPANN* Prototype communicates with the Floodlight controller, which controls Epoch's network, via Floodlight's NorthBound API. With this API the *Network Infrastructure Interface* gathers flow table informations from the OpenFlow Network Devices and network informations such as latency in a given path between two of these devices.

The *Network Infrastructure Interface* uses Floodlight's Northbound API by sending JSON data via HTTP to the host IP, where the network controller is running, in the 8080 port. The HTTP URLs used to gather information are presented in 5.1, 5.2 and 5.3. Each of these URLs, as well as the data they return, are explained below. The Python libraries used for sending HTTP requests is the *requests*, while the data is encapsulated in a JSON format to be sent in the HTTP request by the *json* library.

$$localhost : 8080/wm/core/topology/links/json \tag{5.1}$$

$$localhost : 8080/wm/core/switch/all/flow/json \tag{5.2}$$

$$\begin{split} localhost : 8080/wm/routing/paths/ < src - dpid > / < dst - dpid > \\ / < num - paths > /json \end{split} \tag{5.3}$$

Each of these URLs pools different information from the network. The URL 5.1 gets the link information from the network, defining which switch is connected to each port of each switch. URL 5.2 gets the flow table information of all switches in the network. Finally, URL 5.3 gets the latency between two switches and the path that produces this latency. This last URL is parameterized, where *src-dpid* and *dst-dpid* are the ID from the switches that the prototype wants to know the latency in their communication. *num-paths* is the number of paths that the controller must return for the communication between the *src-dpid* and *dst-dpid*; this last variable is set to 3 for comparison in the violation detections.

The information returned by the network controller in the HTTP response of these URLs are them formated in Python dictionaries and passed to the *NN Verification Module* to be tested in the NN violation detections for the country entered by the user.

5.3 NN Violation Detection and Their Results

With all the information needed to perform NN violation detections, the *NN Verification Module* can run the algorithms for the country the user entered in the *Detection Parameters Interface*. This module reads the XML file with all countries NN violation detection information searching for the country defined by the user. *ISPANN* Prototype then refers the algorithms from the country in this XML file in its implementation and runs each of these algorithms.

$$
\begin{aligned}
&violation\_Detections[\\
&\quad \{source\_IP, destination\_IP\} : [\\
&\quad\quad dpid_x,\\
&\quad\quad dpid_y,\\
&\quad\quad \ldots\\
&\quad],\\
&\quad \ldots\\
&]
\end{aligned}
\tag{5.4}
$$

The *NN Verification Module* then returns the results of the algorithms to the *Detection Parameters Interface* in the format presented in 5.4. Each algorithm has its own *violation_Detection* structure composed by a set of communications that are being violated. The tuple *{source_IP, destination_IP}* define one of these communications in the network. Each of these tuples consist of a set of switch IDs (dpids) where that communication is being violated. For the disclosure of the results, the communications tuples that are possibly being violated are displayed in the command line for the user view. A more detailed log text, with the switch *dpids* where these violation occurs is created, for further analysis by the user.

5.4 NN Violation Detection Algorithms

As mentioned in Section 4.2, the NN violation detection algorithms are an important part of *ISPANN*. This section presents the algorithms that were implemented in this thesis, for evaluation purpose of the *ISPANN* Prototype. Each NN violation detection class has an algorithm implemented to it, with the exception of the **user discrimination** class. This class 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 in the evaluation of *ISPANN* Prototype, presented in Chapter 6.

5.4.1 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 communication 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 queries the URL 5.2, getting the flow table informations of all switches in the network. Then, it looks 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 drop packet rule then
4:             alert a possible NN violation
5:         end if
6:     end for
7: end for
```

In OpenFlow, "drop packet rules" are flow table rules that define an entry port, a source IP and a destination IP of the packets, but does not define for which port that packet must be forwarded. In this sort of flow table rule, OpenFlow switches understand that they must drop the packets that meet the entry port, a source IP and a destination IP requirements.

5.4.2 User Discrimination

User discrimination implies that an user cannot arbitrary be picked to have his/her communication degraded. This sort of discrimination is outlawed in Chile as its NN regulation points that an ISP *cannot interfere or disturb, the rights of any Internet user*.

In *ISPANN* Prototype, 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, which could be used for this communication. 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, utilizing the link information from URL 5.1 and the flow table informations from URL 5.2, and comparing it with alternative network path for this communication. These alternative paths are gathered with the URL 5.3, which gives a set of paths between two switches and their latencies, as explained in section 5.2.

Algorithm 2 iterates over all OpenFlow rules on all switches (lines 1-5) to identify users in the network and their communication paths. This is accomplished by getting the link informations from URL 5.1 and composing then with the forwarding rules from URL 5.2. For each user flow path, is acquired its communication latency, utilizing URL 5.3 (line 6). Then, a network latency threshold is established, 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, the algorithm looks for alternative paths in the network that this user communication could be forwarded to (lines 8-16), again with the information from URL 5.3. In the case where an user communication has more latency than the threshold, the prototype gets alternative 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, the algorithm compares it to the user communication path latency identified before (lines 12-14). It is considered a NN violation if any of the alternative paths have better latency than the path instantiated for the user communi-

Algorithm 2 User Discrimination Detection

```
for all switches do
    for all switch rules do
        identifies flow paths
    end for
end for
latencies = get(latencies of users flow paths)
thresholdLatency = mean(latencies) + std.dv(latencies)
for all latency in user flow paths do
    if latency > thresholdLatency then
        identifies alternative flow paths
        get(latencies of alternative flow paths)
        if latency > alternative flow paths latencies then
            alert a possible NN violation
        end if
    end if
end for
```

cation.

In Algorithm 2, User Discrimination Detection was based in user communication latencies and alternative flow paths with less latency for each user. In the case where a management protocol does not have the capabilities of gathering network topology information, such as from URLs 5.1 and 5.3, *ISPANN* Prototype cannot measure users communication latencies. So, Algorithm 3 was implemented for comparison purposes between cases with and without network topology information. Here it tries to infer it, based on the switch ports the user communication utilizes and their load, information that appear in the results from URL 5.2 requests.

Algorithm 3 User Discrimination Detection Without Network Topology Information

```
for all switches do
    for all switch rules do
        identifies user forwarding port load
    end for
end for
for all user forwarding port load do
    $meanLoad_i = mean(user forwarding port load)_i$
end for
thresholdLoad = mean(meanLoad) + std.dv(meanLoad)
for all load in meanLoad do
    if load > thresholdLoad then
        alert a possible NN violation
    end if
end for
```

To detect an User Discrimination without utilizing 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, which are returned in the URL 5.2 request. Next, for each user communication it is calculated the mean of the loads of all switch ports in its communication path, represented in Algorithm 3 by *meanLoad* (line 7). Then, it is defined 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 Algorithm 3 does not utilizes network topology information, it cannot determine alternative paths for the user communication. Instead of comparing the path of the user communication with alternative paths, Algorithm 3 infers that any *meanLoad* in users communications over the threshold established in a NN violation.

5.4.3 Application/Service Discrimination

Similarly to User Discrimination, Application/Service discrimination means that an application or service cannot arbitrary be degraded. FCC bans application/Service discrimination by preventing ISPs from *throttling* 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 in the switches flow tables.

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 IP, ethernet type, forwarding port]* with the flow tables information from URL 5.2. Then, for each application forwarded to the same destination IP, but in different ports (lines 5-11), *ISPANN* Prototype utilizes the information from URL 5.3, 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

Algorithm 4 Application/Service Discrimination Detection

```
for all switches do
    for all switch rules do
        identifies application forwarding
    end for
    for all application forwarded to same destination do
        get application path latency
        identifies alternative flow paths
        if application latency $>$ alternative flow path latency then
            alert a possible NN violation
        end if
    end for
end for
```

flow path to the destination has better latency than the path instantiated to the application, a possible NN violation is alerted.

5.4.4 Paid Prioritization

Algorithm 5 Paid Prioritization Detection

```
for all switches do
    $maxPriority = 0$
    for all switch rules do
        if $rulePriority > maxPriority$ then
            $maxPriority = rulePriority$
        end if
    end for
    for all switch rules do
        if $maxPriority > rulePriority$ then
            alert a possible NN violation
        end if
    end for
end for
```

Paid prioritization is explicitly proscribed by FCC. OpenFlow has the *priority* field which represents the priority level of a flow entry. Flows with more priority than others are being prioritized. *ISPANN* does not have the information if this prioritization is paid or not, but it alerts this prioritization to the network operator as a possible NN violation, as prior Algorithms.

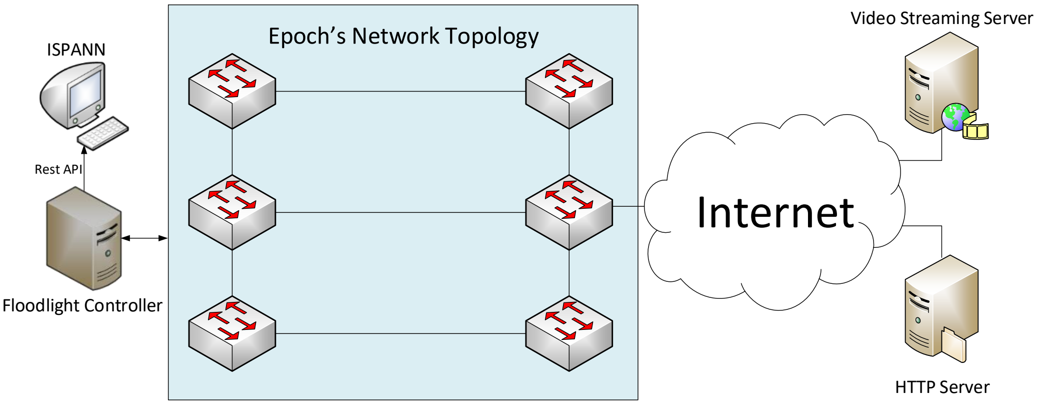
In Algorithm 5, *ISPANN* Prototype looks up all switch rules twice, with the information from the URL 5.2 request. In every switch (lines 1-13), it iterates over each rule, looking for the maximum priority value existing in that switch (lines 2-7). Then, in the second iteration, it verifies what rules have less than that maximum priority value (lines 8-12). Those communications which have underprioritized rules are possibly having their neutrality violated (lines 9-11).

6 EVALUATION

This chapter presents *ISPANN* Prototype evaluation performed for this thesis. This evaluation assessed prototype's scalability, measuring the execution time of the algorithms implemented on it, that were presented in 5.4. Then, the NN violation detection results of the three policies from Section 4.2 were compared, to show how different policies have different NN violation detection results, thus validating the necessity of a system that suits to emerging NN policies. Complementing this test, a comparison between different number of services was executed, to test how the NN violation detection results vary with the number of services in the network. Finally, an accuracy comparison over *ISPANN* Prototype violation detections was performed, comparing the usage of different sets of network information, associating these accuracy results to the prototype execution performance requirements.

As mentioned in Section 4.1, *ISPANN* Prototype was evaluated using Mininet (LANTZ; HELLER, 2011), a network emulator which implements OpenFlow based networks, where Epoch's network was instantiated. Along with the network devices, many virtual hosts were added to emulate end-user communications. Mininet runs in a 4 GB RAM virtual machine with Ubuntu as its operational system, while Floodlight and the prototype both run in another virtual machine with 1 GB RAM and with Ubuntu as well.

Figure 6.1: Epoch's network topology and our evaluation scenario



Source: (The Author, 2018)

As mentioned above, n users were introduced in Epoch's network, choosing randomly which switch to add each host. This n was varied linearly, 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 that function as servers: one streaming a video and another that responds HTTP requests. 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, simulating the current status of Internet traffic. The whole scenario is represented in Figure 6.1.

Then, 10% of the flows in the normal communication stage of the network were selected and changed, to insert NN violations in these flows. For example, assuming a **blocking** violation: it was selected a normal forwarding flow, saving this flow matching values, remove the flow from the switch flow table and adding a "drop packet rule" to the switch with that matching status. Table 6.1 presents the changes made to the network for each NN violation introduced.

Table 6.1: Network changes when introducing NN violations

NN Violation Detection class	Change in the Network
Blocking	Remove a forwarding flow and add a "drop packet rule" with the corresponding matching values
User Discrimination	Remove a forwarding flow and add another forwarding flow with the corresponding matching values but different output port, in a path with more latency
Application/Service Discrimination	Add a flow with the same matching values but different *ethernet_type* value, forwarded in another port
Paid Prioritization	Remove a default *priority* flow and add a flow with max *priority* with the corresponding matching value

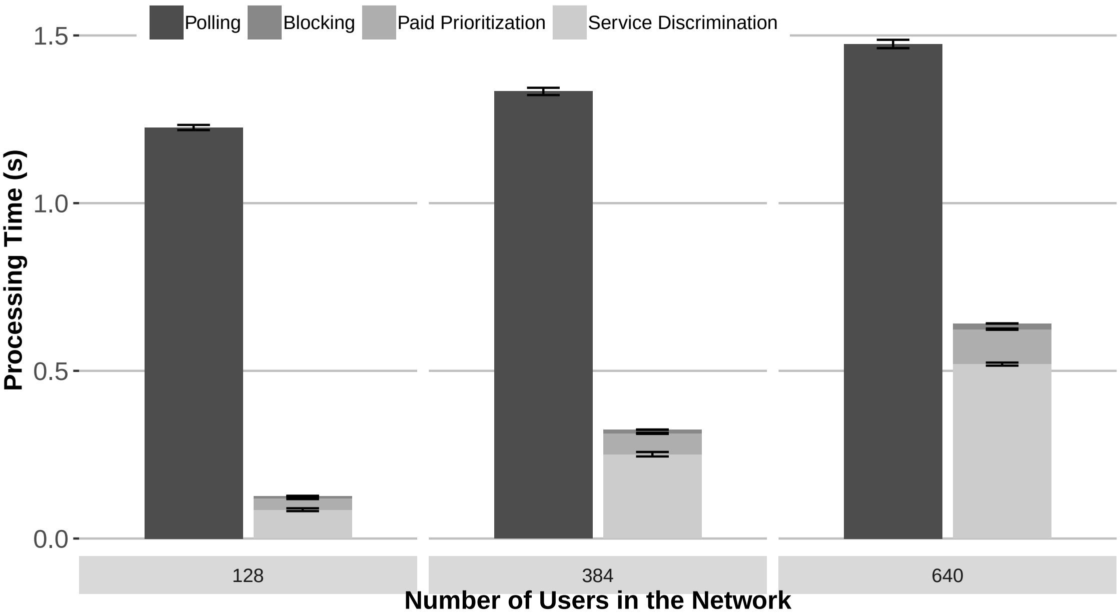
Source: (The Author, 2018)

Finally, the *ISPANN* Prototype was executed and displayed the NN violation detection results of the NN violation detection algorithms implemented. In the end of the prototype execution In each test of this evaluation, 30 samples of the corresponding data were collected. With these results, it was obtained a confidence interval based on a confidence level of 95%.

6.1 Performance Comparison

In this first test, the relation between the number of users in the network and the time *ISPANN* Prototype takes to process network violations was studied. As Epoch is a ISP from US, FCC's policy were utilized in this test. The results for this test are shown in Figure 6.2. As shown, Application/Service Discrimination processing is substantially higher than Blocking Detection and Paid Prioritization detection. For instance, when there is 640 users in the network, Application/Service Discrimination is responsible for 81% of the FCC policy processing time, among the algorithms implemented in *ISPANN* for it.

Figure 6.2: Polling and algorithms time comparison in *ISPANN* Prototype assorted by the number of users in the network

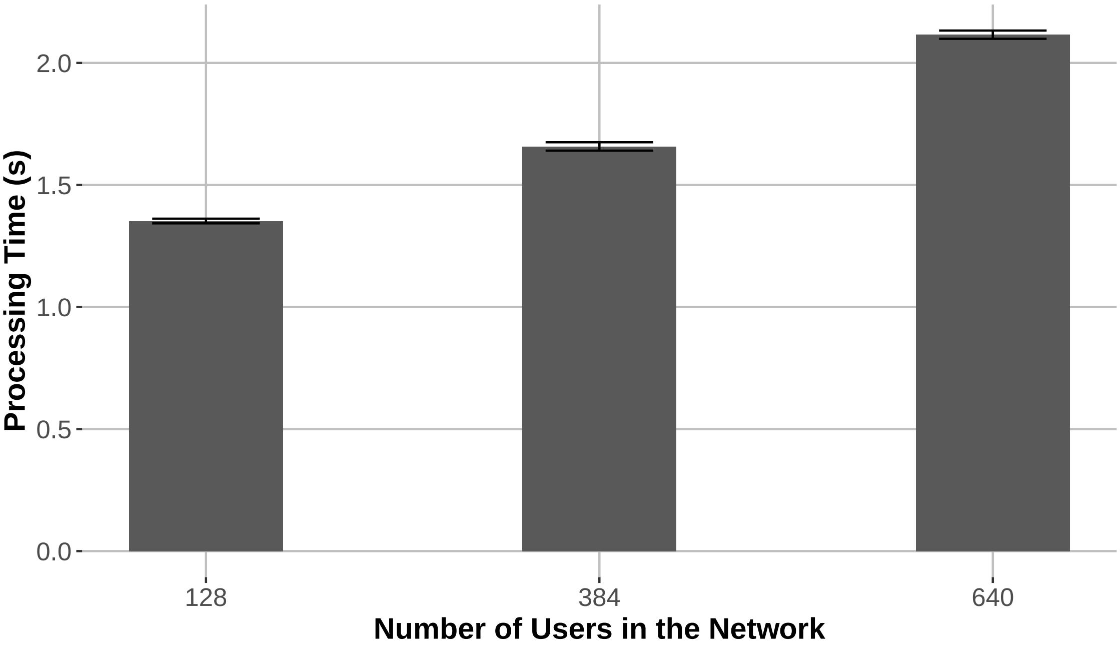


Source: (The Author, 2018)

This huge difference in processing time is due the multiple information Application/Service Discrimination has to process from the network with various path forwarding information, in contrast to Blocking Detection and Paid Prioritization, that only process flow table informations, acquired with URL 5.2 requests, resulting in a large gap of information each algorithm must process. For instance, Blocking Detection executes for 0.07 and 0.017 seconds for 128 and 640 cases consecutively, while Paid Prioritization executes for 0.033 and 0.104 seconds and Application/Service Discrimination executes for 0.086 and 0.52 seconds in the same cases.

Another thing to mention is the difference between the polling time and the network information processing time gap. The time that Floodlight takes to respond the Northbound API responses is not negligible, as noted in Figure 6.2. In fact, in the tests run, most of *ISPANN* Prototype processing is due to the polling part of the system. However, the processing time from the algorithms, even being linear, scales faster than the polling time. By linear regression, it is expected that at 2276 users the algorithms start to take more time to be executed them the polling part. Summarizing the data from Figure 6.2, Figure 6.3 shows the scalability of the total time *ISPANN* Prototype takes detecting

Figure 6.3: *ISPANN* Prototype temporal performance with number of flows variance for FCC policy



Source: (The Author, 2018)

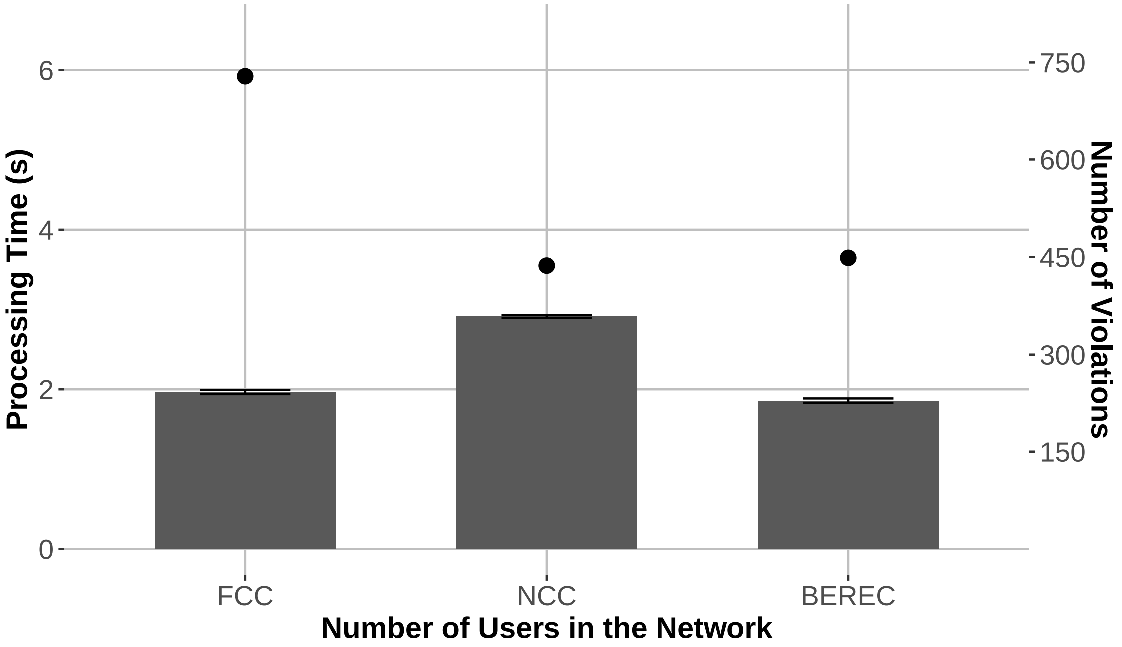
violations to the FCC policy in the network.

6.2 Violation Comparison Per Country

Next test shows how FCC, BEREC and the NCC policies differ when running the algorithms implemented to each one of them in *ISPANN* Prototype. This test has been executed with 512 users in Epoch's network and the results are presented in Figure 6.4. As FCC, BEREC and NCC consider blocking as a NN violation, it is able to correlate the different results with the other violation classes. For example, the difference between FCC and BEREC is that FCC executes Paid Prioritization Detection. So, the 80 violations FCC detects more than BEREC, and the 0.11 second it took for this detection, is due to the Paid Prioritization Detection. In addition, the difference between NCC policy and BEREC is that BEREC realizes Application/Service Discrimination Detection and NCC performs User Discrimination Detection. These differences results in 12 less violations detected by the NCC policy, even though NCC processes for 1.06 seconds more (1.57 times BEREC processing time).

With these results two conclusions can be obtained. First, there is no correlation between the time *ISPANN* Prototype spends running NN violation detection tests and the

Figure 6.4: Comparison of violation detections between countries



Source: (The Author, 2018)

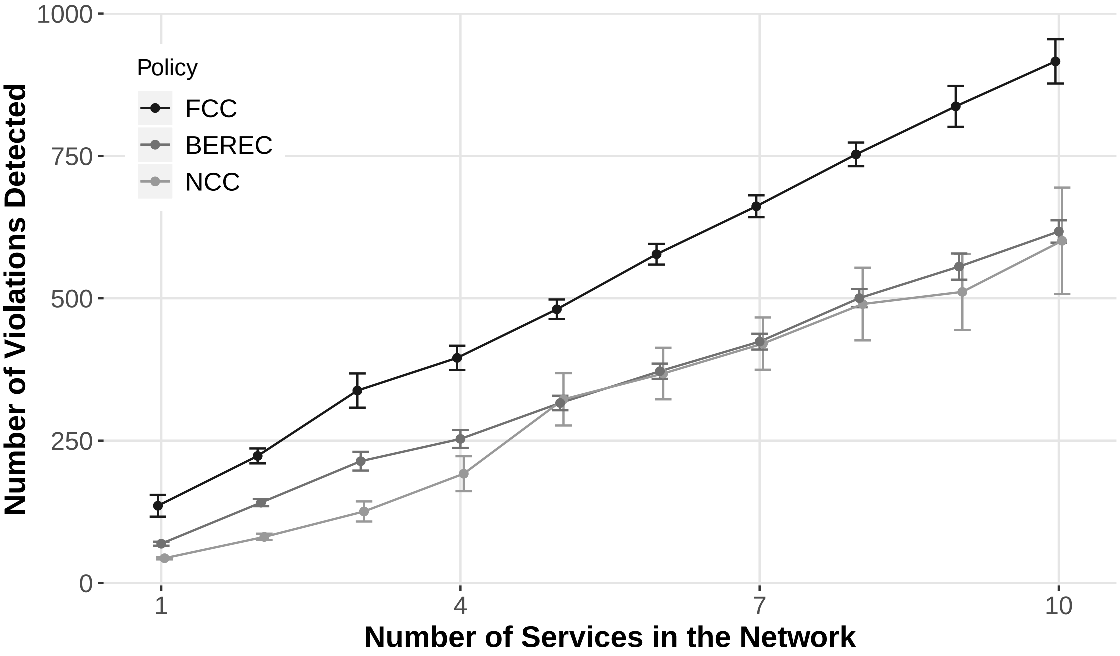
number of violations detected in the network. As Figure 6.4 shows, NCC is the policy that detects the least NN violation detections, but is the one that takes more time for *ISPANN* Prototype to process. Though, there is a hint on the dependency of the number of violations of a given NN violation test: the number of violation classes related to it. It is not clear in the graph, as it is not its objective, but the only test that has more violations detected is the FCC one, which has more detection classes compared to the NCC and BEREC ones.

6.3 Violation Comparison Variating the Number of Services

This test has the intent to show how the NN violation detection results respond to the variation in the number of services in the network, in complement to the previous test, that gave a hint over this issue. Here, the number of users in the network has been kept in 128, while the number of services in the networks has been varied from 2 (base value used in other tests) to 10 services. The result from this test is presented in Figure 6.5.

This result shows that the NN violation detections vary linearly with the number of services in the network and, consequently, with the number of flows in the network devices. So, the number of violations is proportional to the number of flow table entries in the networks, which was expected. Another way to see it is by comparing the results

Figure 6.5: Comparison of violation detections between countries



Source: (The Author, 2018)

from the Subsection 6.2 with this one. With 512 users in the network and two services (prior result), there are 1024 communications in the network. In this test, for 128 users, the case where there are 1024 communications in the network is with 8 services. Table 6.2 compares this case with prior test results and shows their similarity, with prior test results being very close to the (128, 8) point of this test.

Table 6.2: Comparison between the number of violation detection results in prior test *versus* this test, in the 8 service case

	Prior Test Violations	This Test Violations
FCC	729	752
BEREC	449	500
NCC	437	489

Source: (The Author, 2018)

However, the result also shows that the hint from previous test is true: the number of violations in the network also vary with the number of classes describing a NN policy. In Figure 6.5, lines from NCC and BEREC NN violation detection results increase in the same rate, while the FCC test result increase faster. As already commented, the difference between NCC and BEREC NN violation detections is the User Discrimination present in the NCC test and the Application/Service Discrimination present in BEREC. This difference is translated to the graph by the points where the NCC test falls out of the line and its higher confidence interval compared to BEREC. In contrast, the difference between

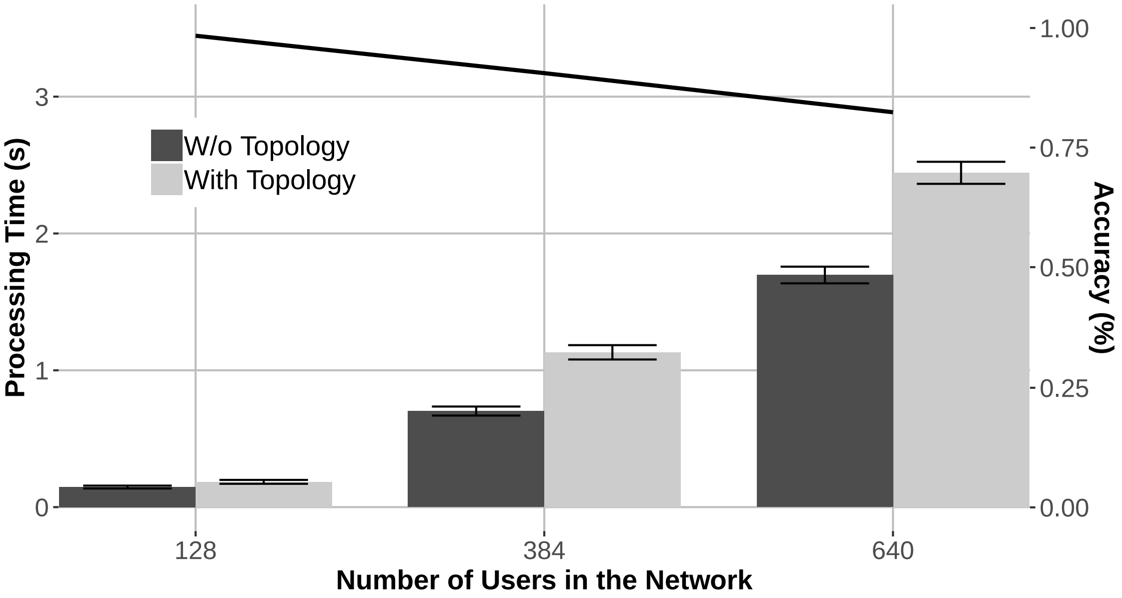
FCC and BEREC is the Paid Prioritization class present in the FCC test. This class alone is responsible for the the higher increasing rate FCC has.

6.4 Accuracy Comparison

Finally, both versions of User Discrimination Detection are run to verify how the number of network informations used in NN violation detection algorithms impact its results. This kind of study must be done, for example, in the case where a network manager have a processing resource limit for *ISPANN* and wants to know how accurate its detection can be with this limitation and if this tradeoff is acceptable.

Algorithm 2, presented in Subsection 5.4, was assumed as a baseline and compared Algorithm 3 in therms of time performance and accuracy results. 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.

Figure 6.6: Temporal and accuracy comparison of User Discrimination Detection using only flow table informations *versus* flow table and topology information



Source: (The Author, 2018)

Figure 6.6 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 0.04 seconds and with 640 users it is 0.75 seconds. So, with 5 times more users in the network, the time performance difference between both algorithms increases 18 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 83%.

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, algorithms should be optimized to use a limited set of network informations.

7 CONCLUSION

NN is a problem that is often revisited by the society for its ambiguity and the frequency that is changes in the course of the years. A single change in the presidency of a country can rework the policies adopted by that country and raise the debate once again. In parallel, the technical academic works, while participating on this debate, apply their point of view on the NN definitions, disregarding the NN policies of the countries they are executing then. One cannot compare NN violation detection detection tools for the bias base that these NN definitions apply to these works.

In contrast, this thesis took no side in the NN definition debate. Instead, it presented *ISPANN*, a system that performs NN violation detections associated with countries NN policies, from the ISP network operator vantage point. *ISPANN* was devised focusing on two main objectives: reduce the ambiguity generated by NN violation detection works and give a supplementary violation detection tool for network operators. To achieve these objectives, *ISPANN* was designed and implemented as modular as possible, so network operators could input the data they need tested, as well as legislators can describe NN violation detection tests they find appropriate for the legislation they specified. Chapter 3 presented the design of *ISPANN* and the data needed to be input by network operators, while Chapter 4 presented its prototype and further use case informations assumed to the implementation of the system.

Based on these assumptions, a series of evaluations were performed and the results obtained were shown in Chapter 6. These results validate that detecting NN violations in the network operator vantage point is possible and, this premise being proven, the need of a general system while considering countries policies as the NN violations to be detected in the network. The tests were performed in an SDN based network, implemented in Mininet, an SDN network emulator. However, as said above, *ISPANN* being a generic system, not only for countries policies but for network protocols as well, it can be used for traditional networks. In this case, different network configuration protocols should be assumed, such as Netconf, but it escapes the scope of this thesis.

From the tests executed for this thesis, firstly, the Performance Measurement verified that the major part of *ISPANN* execution depends on the detection algorithms implemented on it, while the polling part is negligible. So, legislators that would implement algorithms in *ISPANN* should optimize them, in order to minimize its execution time. Secondly, the Country Policy Comparison showed how different policies lead to different violations in the network and, consequently, different NN violation detections. The results from this test also implies that there is no correlation between the number of violations detected in the network and the time *ISPANN* takes to perform NN violation detections.

Following the Policy Comparison test, and complementing it, the Users and Services Comparison test focused on evaluating the correlation between the number of users and services in the network and the results of NN violation detections of a given NN policy. This test demonstrated that there there is no correlation between the focus of a policy (say, an user focused NN violation detection *versus* an application focused one), and its violation detection results. In addition, it showed that what skews the results of the NN violation detections is the number of violations tested for a given policy.

Finally, the last test aimed to compare possible NN violation detection algorithms. The Accuracy Comparison test showed how different algorithms with the same purpose over a given policy have different requirements and results. In the use case assumed for this thesis, the User Discrimination algorithm, which is part of the NCC policy and BEREC's guidelines, was also implemented in an alternative form. This second implementation utilized only flow table informations, as the prior utilized network informations as well. The results of this comparison showed the trade off the accuracy in NN violation detections and the performance of *ISPANN*. This trade off must be in legislators and network managers mind to perform accurate detections, while staying in the performance range that ISPs systems can handle.

ISPANN is the first step towards a better understanding of the political matter in the NN technical literature. Utilizing a policy-based approach for managing ISP networks, that need to be conformed to the NN legislation they are subjected to, has proven being viable and achieved the objectives that the system was designed to with the tests described above. However, it is expected that that legislators and the academic community contribute with NN violation detection algorithms, so *ISPANN* can accurately detect violations according to what legislators expect with the policies they define.

REFERENCES

AMJHR. **Law Number 27078**. 2014. Available at https://bit.ly/2fDLTfS.

BEREC. **BEREC Guidelines on the Implementation by National Regulators of European Net Neutrality Rules**. 2016. Available at https://goo.gl/8NID1G.

BIRK, M. et al. Evolving to an sdn-enabled isp backbone: key technologies and applications. **IEEE Communications Magazine**, v. 54, n. 10, p. 129–135, October 2016. ISSN 0163-6804.

BUSTOS-JIMÉNEZ, J.; FUENZALIDA, C. All packets are equal, but some are more equal than others. In: **Proceedings of the Latin America Networking Conference on LANC 2014**. New York, NY, USA: ACM, 2014. (LANC '14), p. 5:1–5:8. ISBN 978-1-4503-3280-4. Available from Internet: <http://doi.acm.org/10.1145/2684083.2684088>.

BYUN, J. E.; LEE, S. Study on controversial issues related to network neutrality in leading countries, focusing on economic efficiency and user protection. In: **2013 Proceedings of PICMET '13: Technology Management in the IT-Driven Services (PICMET)**. [S.l.: s.n.], 2013. p. 2784–2794. ISSN 2159-5100.

CC. **Law number 1450 of 2011**. 2011. Available at https://bit.ly/1GpKqBc.

CRB. **Amending the Law of 13 June 2005 on Electronic Communications to Ensure the Neutrality of Internet Networks**. 2011. Available at https://bit.ly/2CZJixs.

DISCHINGER, M. et al. Detecting bittorrent blocking. In: **Proceedings of the 8th ACM SIGCOMM Conference on Internet Measurement**. New York, NY, USA: ACM, 2008. (IMC '08), p. 3–8. ISBN 978-1-60558-334-1. Available from Internet: <http://doi.acm.org/10.1145/1452520.1452523>.

ENNS, R. et al. **Network Configuration Protocol (NETCONF)**. 2011. Available at https://bit.ly/2MFgx8W.

FAS. **The Principles of Network Neutrality Began to Operate in Russia**. 2016. Available at https://bit.ly/2xlsCuk.

FCC. **Open Internet**. 2015. Available at https://www.fcc.gov/general/open-internet.

FLOODLIGHT. **Floodlight Controller**. 2012. Available at http://www.projectfloodlight.org/.

FNA. **Information report**. 2011. Available at https://bit.ly/2xhkSu9.

GARRETT, T. et al. Monitoring network neutrality: A survey on traffic differentiation detection. **IEEE Communications Surveys Tutorials**, v. 20, n. 3, p. 2486–2517, thirdquarter 2018. ISSN 1553-877X.

GHARAKHEILI, H. H.; VISHWANATH, A.; SIVARAMAN, V. Pricing user-sanctioned dynamic fast-lanes driven by content providers. In: **2015 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)**. [S.l.: s.n.], 2015. p. 528–533.

GHARAKHEILI, H. H.; VISHWANATH, A.; SIVARAMAN, V. Perspectives on net neutrality and internet fast-lanes. **SIGCOMM Comput. Commun. Rev.**, ACM, New York, NY, USA, v. 46, n. 1, p. 64–69, jan. 2016. ISSN 0146-4833. Available from Internet: <http://doi.acm.org/10.1145/2875951.2875962>.

KAKHKI, A. M. et al. Identifying traffic differentiation in mobile networks. In: **Proceedings of the 2015 ACM Conference on Internet Measurement Conference**. New York, NY, USA: ACM, 2015. (IMC ’15), p. 239–251. ISBN 978-1-4503-3848-6. Available from Internet: <http://doi.acm.org/10.1145/2815675.2815691>.

KN. **Amendment of the Telecomminications Act for the Implementation of the Revised Telecommunications Directives**. 2012. Available at https://bit.ly/2D3iuwh.

KNIGHT, S. et al. The internet topology zoo. **Selected Areas in Communications, IEEE Journal on**, v. 29, n. 9, p. 1765 –1775, october 2011. ISSN 0733-8716.

LANTZ, B.; HELLER, B. **Mininet**. 2011. Available at http://mininet.org/.

LEE, D.; KIM, Y.-H. Empirical evidence of network neutrality – the incentives for discrimination. **Information Economics and Policy**, v. 29, p. 1 – 9, 2014. ISSN 0167-6245. Available from Internet: <http://www.sciencedirect.com/science/article/pii/S0167624514000298>.

LI, D. et al. 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)**. [S.l.: s.n.], 2015. p. 297–304.

LING, F.-Y. et al. Research on the net neutrality: The case of comcast blocking. In: **2010 3rd International Conference on Advanced Computer Theory and Engineering(ICACTE)**. [S.l.: s.n.], 2010. v. 5, p. V5–488–V5–491. ISSN 2154-7505.

MARTIN, J. C. D.; GLORIOSO, A. The neubot project: A collaborative approach to measuring internet neutrality. In: **2008 IEEE International Symposium on Technology and Society**. [S.l.: s.n.], 2008. p. 1–4. ISSN 2158-3404.

MCTS. **Federal Law on Telecommunications and Broadcasting**. 2014. Available at https://bit.ly/2xDL7Je.

NCA. **Net neutrality in Norway**. 2015. Available at https://bit.ly/2QBFRju.

NCB. **Brazilian Civil Rights Framework for the Internet**. 2014. Available at https://bit.ly/1kxaoKm.

NCC. **Chilean Network Neutrality law**. 2010. Available at https://goo.gl/3xgkGv.

SANDVINE. **Global Internet Phenomena Report**. 2016. Available at https://www.sandvine.com/trends/global-internet-phenomena/.

SCHEWICK, B. van; FARBER, D. Point/counterpoint: Network neutrality nuances. **Commun. ACM**, ACM, New York, NY, USA, v. 52, n. 2, p. 31–37, feb. 2009. ISSN 0001-0782. Available from Internet: <http://doi.acm.org/10.1145/1461928.1461942>.

SR. **Electronic Communications Act, Page 12069**. 2012. Available at https://bit.ly/2Mg9zb2.

TARIQ, M. B. et al. Detecting network neutrality violations with causal inference. In: **Proceedings of the 5th International Conference on Emerging Networking Experiments and Technologies**. New York, NY, USA: ACM, 2009. (CoNEXT '09), p. 289–300. ISBN 978-1-60558-636-6. Available from Internet: <http://doi.acm.org/10.1145/1658939.1658972>.

TRAI. **Recommendations on Network Neutrality**. 2017. Available at https://bit.ly/2QxKBXn.

WICKBOLDT, J. A. et al. Software-defined networking: management requirements and challenges. **IEEE Communications Magazine**, v. 53, n. 1, p. 278–285, January 2015. ISSN 0163-6804.

WU, T. Network neutrality, broadband discrimination. **Journal of Telecommunications and High Technology Law**, v. 2, p. 141–180, 2003. Available at SSRN: https://ssrn.com/abstract=388863 or http://dx.doi.org/10.2139/ssrn.388863.

ZHANG, Z.; MARA, O.; ARGYRAKI, K. Network neutrality inference. In: **Proceedings of the 2014 ACM Conference on SIGCOMM**. New York, NY, USA: ACM, 2014. (SIGCOMM '14), p. 63–74. ISBN 978-1-4503-2836-4. Available from Internet: <http://doi.acm.org/10.1145/2619239.2626308>.

APPENDIX A — PUBLISHED PAPER (AINA 2018)

Vinícius Garcez Schaurich, Márcio Barbosa de Carvalho, Lisandro Zambenedetti Granville. **ISPANN: A Policy-Based ISP Auditor for Network Neutrality Violation Detection**. In 2018 IEEE 32nd International Conference on Advanced Information Networking and Applications (AINA), pp. 1081–1088, May 2018. DOI: 10.1109/AINA.2018 .00155.

- **Title**: ISPANN: A Policy-Based ISP Auditor for Network Neutrality Violation Detection
- **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.
- **Status**: Published.
- **Qualis / CAPES**: A2.
- **Conference**: 32nd International Conference on Advanced Information Networking and Applications (AINA).
- **Date**: May 16 – 18, 2018.
- **Local**: Krakow, Poland.
- **URL**: <http://voyager.ce.fit.ac.jp/conf/aina/2018/>.
- **DOI**: <https://doi.org/10.1109/AINA.2018.00155>.

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.

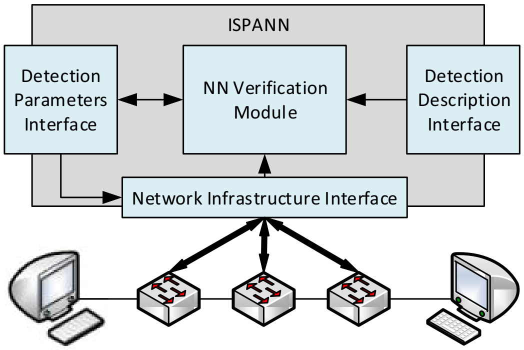

ISPANN
Detection Parameters Interface
NN Verification Module
Detection Description Interface
Network Infrastructure Interface

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.

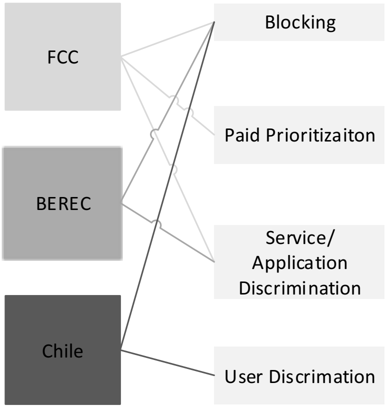


Fig. 2. Policies, NN violation detection classes and their relations

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

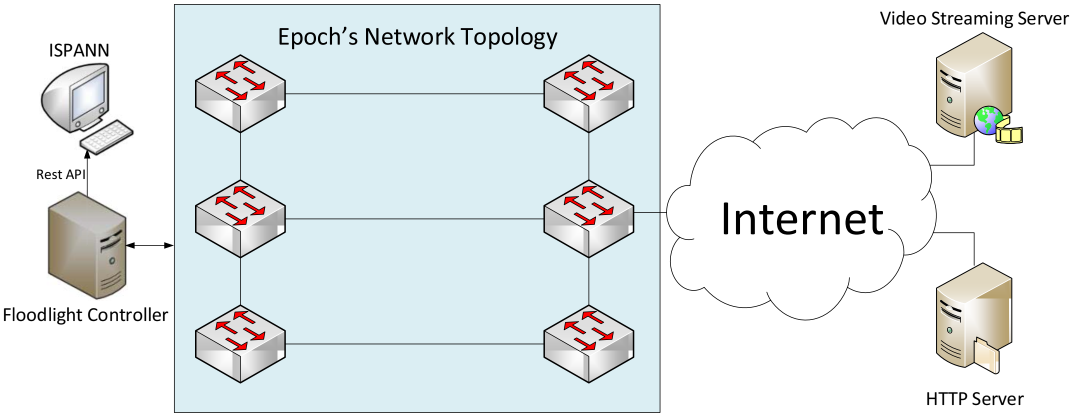


Fig. 3. Epoch's network topology and our evaluation scenario

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.

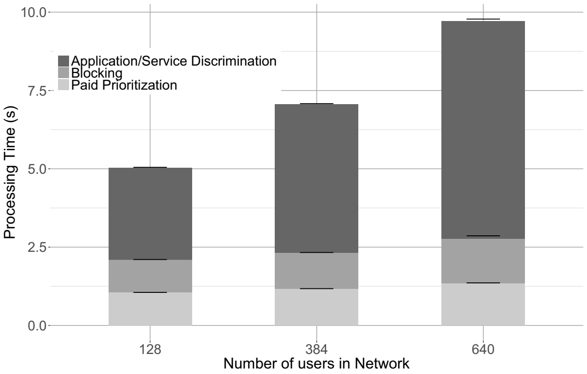


Fig. 4. *ISPANN* temporal performance with number of flows variance for FCC policy

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).

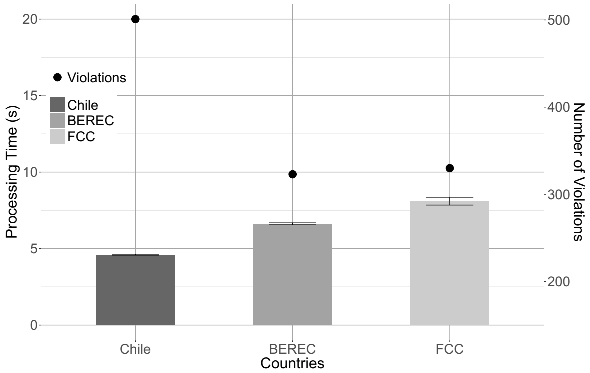


Fig. 5. Comparison of violation detections between countries

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.

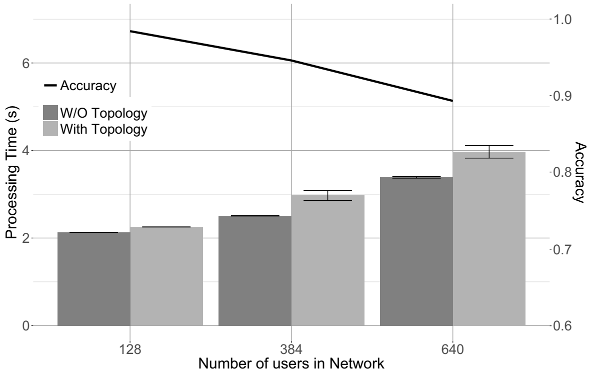


Fig. 6. Temporal and accuracy comparison of User Discrimination Detection using only flow table informations *versus* flow table and topology information

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.