Computer Networks 205 (2022) 108738

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

JurisNN: Judging traffic differentiations as network neutrality violations according to the regulation

Márcio Barbosa de Carvalho *, Lisandro Zambenedetti Granville

Federal University of Rio Grande do Sul, Informatics Institute, Av. Bento Gonçalves, 9500 Porto Alegre, RS, Brazil

ARTICLE INFO

Keywords:
Network Neutrality
Internet Regulation
Jurisdiction
Traffic Differentiation

ABSTRACT

Network Neutrality (NN) is a principle that is not taken for granted on the Internet. Instead, it must be enforced by regulation that defines which Traffic Differentiation (TD) practices are allowed or prohibited from being adopted by ISPs. Thus, the regulation is the proper source of NN definitions to design solutions for detecting NN violations. However, the regulation set by legislators is valid within a geographical area named *jurisdiction*. Therefore, as an end-to-end network path may traverse multiple jurisdictions, distinct NN definitions along with this path must be considered. Thus, to be effective, solutions based on the regulation must consider where in the end-to-end network path the TD was deployed to evaluate the proper NN definitions stated there. We propose a solution named JurisNN that implements the functionalities required to judge TDs as NN violations considering the regulation. To evaluate the JurisNN judgments, we use TD information collected by a state-of-the-art solution to assess whether this information is enough to judge TDs as NN violations by analyzing the conclusiveness of the results (the TD is an NN violation or not). The results show that to help the signaling of NN violations according to the regulation solutions need both to collect the network paths traversed by tests and to pinpoint where the TD was deployed with better accuracy.

1. Introduction

Network Neutrality (NN) is a principle that is not taken for granted on the Internet. Instead, it must be enforced by regulations that define which TD practices are allowed or prohibited from being adopted by Internet Service Providers (ISPs). Therefore, the definitions from the regulations may be the proper source of information to design solutions for the detection of violations of the NN principle. In this case, an NN violation would be the adoption of traffic management practices that were prohibited by the regulators (legislators and regulatory agencies). Indeed, previous work found that under certain circumstances, up to 48% of the detected TDs cannot be signaled as NN violations when the regulation is considered [1], exposing that the regulatory aspect of the Internet cannot be ignored. Being based on the regulation, the solutions may provide legally actionable evidence to support customer complaints against ISPs in the competent judicial authority [2].

The regulations are set by regulators whose acts are valid within a geographical area named *jurisdiction*. Therefore, the NN definitions provided by regulation may vary across different regions of the Internet, given the distinct intention of the regulators responsible for each area. An end-to-end network path connecting a user to an application server may traverse multiple jurisdictions. Thus, an end-to-end communication may be under distinct NN definitions along with this path, as depicted in Fig. 1. However, a TD can be deployed anywhere in the communication infrastructure. Therefore, a solution to signal NN violations based on the regulation, to be effective, must consider where in the end-to-end network path the TD was deployed to evaluate the proper NN definitions stated in that place.

The solutions from the state-of-the-art for detection of TDs usually calculate a metric related to a statistical method, detecting the TD when this metric surpasses a threshold. The detected TDs are then signaled as NN violations. Thus, the NN violations are signaled by adopting definitions intrinsically related to the detection method. In turn, a few solutions apply definitions from the regulation to signal the detected TDs as NN violations. However, they adopt only the regulation definitions from the jurisdiction of their proponents [2]. Therefore, they ignore that multiple definitions for NN may be found along with an end-to-end network path connecting users to application servers.

In this article, we address the problem of the lack of solutions for the detection of NN violations that follow the definitions stated on regulations and also consider the multiple definitions that may be found along with an end-to-end network path. We propose a solution

* Corresponding author.
E-mail address: mbcarvalho@inf.ufrgs.br (M.B. de Carvalho).

[1] This step is named Jurisdiction Assessment in that work.

https://doi.org/10.1016/j.comnet.2021.108738
Received 26 August 2021; Received in revised form 10 December 2021; Accepted 19 December 2021
Available online 7 January 2022

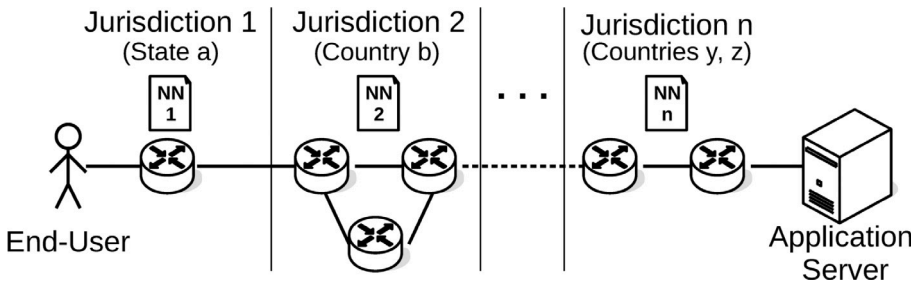


Fig. 1. End-to-end network path traversing multiple jurisdictions [1].

named JurisNN that implements the extra Regulation Assessment step[1] already discussed in previous work [3]. The Regulation Assessment step encompasses the management of information related to the NN definitions stated on regulations around the world, the processing of information about the detected TD to compare it to these definitions, and the signaling of NN violation when the TD violates these definitions. Therefore, it may be challenging to implement such functionalities in every solution committed to detection of NN violations. Given this, JurisNN provides the Regulation Assessment step as a service to be consulted by existing and future TD detection solutions. JurisNN relies on information models designed using the findings identified in previous work [1]. For the processing of TD information and the signaling of the NN violations, JurisNN relies on *Judgment Algorithms* that accommodate nuances in how jurisdictions are established on the Internet and distinct ways of processing the TD information.

A JurisNN prototype was designed to evaluate whether state-of-the-art solutions provide enough information to signal NN violations using definitions from the regulation. The dataset of TDs detected by Wehe [4] was used to provide information about detected TDs. As Wehe does not collect the network paths traversed by tests, which JurisNN requires, the Réseaux IP Européens (RIPE) Atlas platform [5] was used to collect network paths connecting the client to the measurement server of each test. This information is processed by Judgment Algorithms that return the verdict of the analyzed TD to be an NN violation according to the regulation and a metric that reflects the verdict conclusiveness (similarity index). The evaluation analyzes the conclusiveness of the judgments achieved by JurisNN (the TD is an NN violation or not), discussing whether the available information is enough to signal NN violations using definitions from the regulation properly.

The remainder of this article is organized as follows. In Section 2, background information about Internet regulation and jurisdiction issues are discussed, along with the presentation of a few representative examples from the state-of-the-art for detection of TDs and NN violations. In Section 3, the concepts and requirements for a solution devoted to performing the Regulation Assessment step are presented, identified in previous work [1], along with the information models that were designed following these findings. In Section 4, the JurisNN architecture is briefly presented along with details of the Judgment Algorithms. In Section 5, information about the Wehe dataset, the collection of network paths using the RIPE Atlas platform, the regulation definitions considered, the achieved judgment results, and the analysis caveats are presented. In Section 6, the article is concluded by providing final remarks and future work opportunities.

2. Background information and related work

In this section, background information about Internet regulation and how jurisdictions are established on the Internet are presented. A few examples of solutions from the state-of-the-art for the detection of TDs and NN violations are also presented.

2.1. Internet regulation

In this subsection, Internet regulation concepts that influence how to signal NN violations using definitions from regulation are presented.

There was a debate around whether the Internet should be regulated and how the regulation should be established [6]. One discussed approach is self-regulation, in which the Internet actors elaborate the rules that must be followed, which could increase the legitimacy of the rules. However, such an approach is impacted by the difficulty of imposing punishments due to the private nature of the Internet actors (providers, companies, and organizations) who lack the power to enforce the rules. Another discussed approach is the regulation by national laws imposed by regulators (*e.g.*, legislators, regulatory agencies), which could use the state power to enforce the rules. However, such an approach is impacted by the Internet characteristics (*e.g.*, decentralized structure, lack of territorial limits) that hinder the regulator legitimacy in the whole network. Another discussed approach is the regulation by international treaties, in which countries could agree about the harmonization of rules. This approach could increase regulation legitimacy and enforceability. However, this approach is impacted by the consensus required to define the rules, given that many countries should sign the treaty to be effective. However, countries may not agree about some issues (*e.g.*, freedom of speech, censorship) hindering the treaty establishment and its coverage.

Nowadays, Internet regulation is a mix of self-regulation, national-scope regulation, and international treaties. Self-regulation is applied in technical aspects of the Internet by organizations like Internet Corporation for Assigned Names and Numbers (ICANN), Internet Engineering Task Force (IETF), and World Wide Web Consortium (W3C). In this sense, self-regulation helps in the legitimacy of the technical recommendations stated by these bodies because they have significant representation from the technicians. National laws and policies are being established by many countries, regulating the economics and social aspects of the Internet, such as auditing, civil rights, and network neutrality. In this sense, national laws help to achieve the required enforceability since the state has the power to impose punishments. International treaties are rarer due to the difficulty of their establishment. However, even though the European Union (EU) has established a single Internet regulation for the region. However, regulation enforcement is delegated to the state members. This mix of regulations increases the legitimacy and enforceability of Internet regulation [6].

The approaches mentioned above are examples of regulation based on rules defined *a priori* through soft or hard laws [7], which is named an *ex-ante* approach. However, in some countries, the supposed harmful conducts performed by ISPs are evaluated *after* their occurrence, which is named an *ex-post* approach. In this approach, these conducts are claimed in the regular judicial system or a specialized bureau. The claims are judged according to jurisprudence, *i.e.*, the previous understanding of certain conduct and the specific harm caused. For instance, this regulation approach has been adopted in the United States of America (US) since mid-2018 for the NN regulation [8]. This lack of federal regulation in the US opened the opportunity for US states to establish their NN regulations within their jurisdictions.

The development of these multiple models for Internet regulation is intrinsically related to the discussion of who has the power to establish the Internet regulation and where such regulation can be valid, which is named the regulation *jurisdiction* [9]. Therefore, the jurisdiction concept is closely dependent on geographical information because it must assess who is competent in an area. However, due to this localization dependence, the jurisdictions are hard to establish on the Internet because of its characteristics (*e.g.*, decentralized structure, lack of territorial limits). Beyond, there are many components (*e.g.*, servers, routers, endpoints) and actors (*e.g.*, users, ISPs, Content Providers (CPs)) to track the localization that made this task even harder. For instance, Chertoff et al. [10] propose four ways to establish the jurisdiction of a case related to data based on: data creator citizenship, data subject citizenship, data holder citizenship, and location of the harm, exposing the complexity of the matter.

In order to establish the jurisdictions, the courts usually adopt tests that assess characteristics of the case to decide whether they have competence over it, which is referred to as admissibility control. The straightforward way to establish jurisdiction is by assessing where the harmful action took place. Therefore, if the harmful action happened within the court's jurisdiction, the court can judge the case. Courts around the world commonly use this test. In turn, US courts adopt, among others, the *Targeting test* [11] that consists in assessing the place where harmful action has *effects*. For instance, a company established in jurisdiction A committing harmful actions against its clients in jurisdiction B. Then, the clients can claim against the company in jurisdiction B because the effects of its harmful actions were targeted to that jurisdiction. In turn, German courts adopt both tests: where the harmful action happened and where the injury incurs [12]. These tests must be considered when establishing the jurisdiction of a TD.

2.2. Solutions for detection of TDs and NN violations

In this subsection, a few examples from the state-of-the-art for detecting TDs and NN violations are presented, focusing on aspects related to this article, such as the criteria adopted to signal NN violations.

Adkintun [2] is a platform composed of Test servers and probes designed to monitor the fulfillment of the Chilean NN regulation. Therefore, it is among the few solutions that use regulation definitions to signal NN violations. The probes may be deployed as a client application or an instrumented wireless router. The probes measure a set of indicators based on the regulation (latency, jitter, bandwidth, Internet Protocol (IP) changes, availability, and packet loss) to characterize the broadband Internet quality. End-users may access the results in a portal through aggregated measures from their ISP, their service level, and the quality perceived by other users.

ISPANN [13] is a tool designed to detect NN violations (blocking, throttling, and prioritization), by auditing the network devices looking for configurations that may violate the NN principle, such as the blocking of applications or routing packets through links of higher latency. The auditing process considers the rules established on the regulation of the country where the ISP operates selected by the ISP network administrator or an auditor. For each rule, the tool performs an algorithm over the collected configuration to detect NN violations. The audited configuration is collected directly from devices through management interfaces (*e.g.*, OpenFlow). It is important to note that this is the first work that recognizes the importance of accounting for the multiple definitions for the NN principle found in different countries. However, its goal is to audit network configurations looking for instructions that may impose NN violations. Therefore, it is not a solution for detecting end-to-end NN violations.

NeutMon [14] is a system designed to detect violations of the NN principle (blocking, throttling, prioritization, and degradation) in mobile networks. The system detects differentiations that limit the application bandwidth or route packets through slower or more congested links. In order to detect bandwidth-based differentiation, the system compares the performance (speed test) achieved by synthetic BitTorrent traffic and control traffic. The control traffic mimics the BitTorrent protocol behavior (same payload sizes and inter-packet times) but filled with random bytes. In order to detect routing-based differentiation, the system identifies the hops along with the path between the client and the server using an approach similar to the traceroute (Time to Live (TTL) manipulation). However, in contrast to existing traceroute approaches that do not establish the connection at the transport level, the system avails the connection used in the speed test phase to perform the path identification. After that, the mechanism identifies the hops that the application traffic traversed because it uses the same connection the application uses. The routing differences are identified hop-by-hop, *i.e.*, the system builds sets of addresses achieved by the application and control traffics in the ith hop in multiple test repetitions to reduce noise. Then, the system computes the cardinality of the sets of addresses achieved by the application, excluding the addresses achieved by the control traffic and vice-versa. High set cardinalities indicate that the ISP routes packets using information from the transport and application levels due to routing-based differentiation.

Wehe [4] is a system designed to detect NN violations (throttling). Its architecture consists of a smartphone app and a replay server. The system can replay traffic of many applications, such as Netflix, YouTube, and Facebook. The application traffic performance is compared to control traffic performance, which has the payload filled with inverted bits from the application traffic. The traffic is replayed in both directions: upload and download. In order to detect throttling, the Kolmogorov–Smirnov (K-S) statistical test is adopted, comparing its results using the Jackknife resampling method. If the average throughputs from regular and bit inverted traffic differ by 10% and the result from the whole sample and resampling are similar, Wehe detects the throttling. Wehe also performs aggregated data analysis, which the authors point out to solve a few confound factors that may affect tests (*e.g.*, bandwidth volatility). The data is grouped by ISP and application. This aggregated analysis also allows detecting the rate limit adopted by the ISPs performing a Kernel Density Estimation (KDE) analysis, which identifies the throughputs most present in ISP-app results.

NeutMon and Wehe are examples of solutions that detect NN violations adopting NN definitions intrinsically related to the detection mechanism. To detect throttling, NeutMon compares the Cumulative Distribution Functions (CDFs) for the throughput of application and control traffics. To detect route-based differentiation, it computes the cardinality of the set of addresses traversed by the application traffic, excluding addresses traversed by the control traffic. Wehe also compares the performance of control and application traffics using a statistical test, detecting throttling when the performance differs by 10%. Although Adkintun and ISPANN consider the definitions stated on the regulation, they are not a general approach to tackling the problem of signaling NN based on the regulation. ISPANN is a network auditor designed to help administrators assess their configurations according to the regulation. The jurisdiction issue is solved by the administrator choosing the correct regulation to assess the network configuration according to the ISP's country. In turn, for Adkintun, the solution only considers the Chilean regulation. Therefore, it is impossible to signal NN violations based on regulations stated in other jurisdictions.

In the next section, the background information discussed in this section is used to build the information model designed to support the solution to signal NN violations using definitions from the regulation.

3. Modeling

In this section, the conceptual and modeling information for a solution to address the problem of signaling NN violations following the regulation are presented. In Section 3.1, the concepts that are relevant to the issue, the assumptions made for the modeling, and the solution objectives that impacted modeling decisions are presented. In Section 3.2, the information model designed to support the concepts and stated objectives is presented.

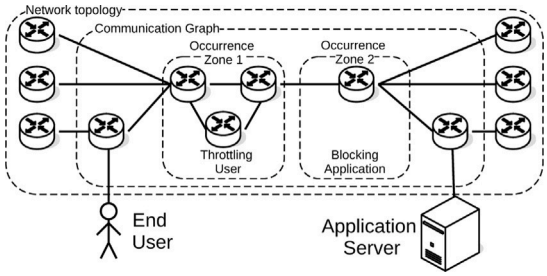


Fig. 2. Network topology, communication graph, and occurrence zones relationships.

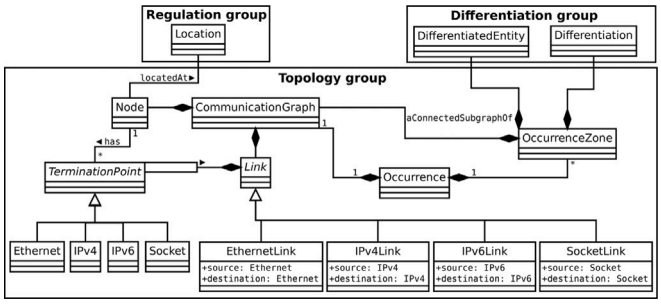


Fig. 3. Information model — Topology classes.

3.1. Concepts, assumptions, and solution objectives

In this subsection, the modeling's concepts, assumptions, objectives, and requirements are presented. This information guided the design of the information model to support a solution that can judge TDs as NN violations considering the definitions from the regulation.

The proposed solution is designed to complement TD detection solutions providing the functionalities of judging TDs as NN violations considering the regulation. In this sense, the detection solution provides the information used to feed the information model. Therefore, the model should be technologically agnostic and easily extendable to accommodate new information because each TD detection solution can identify distinct and specific information.

Jurisdictions are the geographical area where regulators can establish regulations. The side effect is that each regulation is valid only on the jurisdiction of its regulator. This fact leads to the situation depicted in Fig. 1, exposing that the communication can be under distinct NN regulations and NN definitions along with an end-to-end network path. Thus, the modeling must consider the place where the regulations are established as long as the place where the TDs are deployed.

It is common to say that links are non-neutral [15]. However, the non-neutral behaviors attributed to links are introduced by nodes (*e.g.*, router, switch), where traffic management mechanisms dictate how the links shall perform. Therefore, the node is where the violation may happen and can be seen as the criminal responsible for violating the NN regulation.[2] There are a few possibilities to determine the jurisdiction of a case, as discussed in Section 2.1. In this sense, *the place where the node is deployed* is one of the factors considered to establish the jurisdiction.

There is a need to represent the regulations within the solution. In this sense, the regulatory instructions generally have two types of regulatory commands: prohibition or permission. Each command may establish exceptional situations, which also should be represented. As discussed in Section 2.1, states are smaller jurisdictions with their regulations, but these regulations are under a broader regulation (*e.g.*, the national one). Therefore, there is a hierarchy between them: state regulations are under national regulations, national regulations are under regional regulations. There are also global regulation efforts (*e.g.*, Digital Constitutionalism) or first-order constitutional principles (*e.g.*, *"everything which is not forbidden is allowed", "everything which is not allowed is forbidden"* principles) [16] that organize the legal systems. Therefore, this regulation hierarchy should be represented.

There is the need to represent regulatory *interpretations* within the solution because regulatory commands usually refer to classes of traffic. For instance, a possible command could be "traffic blocking is prohibited, except for unlawful content". However, it is not enumerated which traffic is considered "unlawful content". Besides, what is considered "unlawful content" in one jurisdiction may not be considered the same in another. Therefore, the model must represent the interpretation of classes of traffic for distinct regulations.

[2] Of course, nodes are configured by administrators that are the responsible.

There is also the need to represent the topology involved in the communication to map the jurisdictions traversed by the traffic. Therefore, the model must represent nodes (*e.g.*, routers, switches) and links. Additionally, the model must be agnostic to the kind of topology or level of information that TD detection solutions may collect. In this sense, multiple types of connectivity links should be represented: data link level (*e.g.*, Ethernet), network-level (*e.g.*, IP version 4 (IPv4), IP version 6 (IPv6)), or transport/application level connectivity (*e.g.*, Virtual Private Network (VPN) tunnels).

A few solutions try to point wherein the network path the TD is deployed [17]. These solutions lack precision, usually referring to a set of links and nodes where the TD may have been deployed. In order to represent this, it was introduced the concept of *Occurrence Zone*, as depicted in Fig. 2. The Occurrence Zone is the set of links and nodes (a connected subset of the communication graph, which is the part of the topology involved in the communication), the differentiation (*e.g.*, throttling, blocking), and the entity that was differentiated (*e.g.*, user, application). As multiple differentiations in distinct parts of the communication graph may be found, the model must represent several Occurrence Zones within one communication graph.

In the next subsection, the designed information model is presented.

3.2. Information model

In this subsection, the information model designed to comply with the requirements and objectives discussed in the previous subsection is presented. In order to help the presentation of the information model, it was divided into three groups: topology, differentiation, and regulation. However, the designed information model is the composition of these groups. The information model was represented by Class diagrams from the Unified Modeling Language (UML) [18], and most of the attributes were omitted for sake of simplicity.

In Fig. 3, the topology group of the information model is presented. The *Occurrence* class is composed of one *CommunicationGraph* class and many *OccurrenceZone* classes. The *CommunicationGraph* class is responsible for representing the topology involved in the communication. It is composed of *Nodes* and *Links*. A Node may have several *TerminationPoints*, which may be multiple TerminationPoints from the same Open System Interconnection (OSI) layer (*e.g.*, IPv4), such as routers with multiple IP addresses, or maybe TerminationPoints of different layers (*e.g.*, Ethernet and IPv4) for devices that connect with different connectivity technologies, such as Cable Modems or a VPN server. Nodes are located at one *Location* class, presented in the Regulation group. A Link is a self-relation of two distinct TerminationPoints of the same type (*e.g.*, IPv4). As the proposed solution is designed to complement the functionality of state-of-the-art solutions, the type of technology that such solutions may identify may vary. Thus, the information model allows being expanded through the generalizations of TerminationPoints and their associated Link types.

The *OccurenceZone* class is composed of a connected sub-graph of the CommunicationGraph. This sub-graph represents the granularity

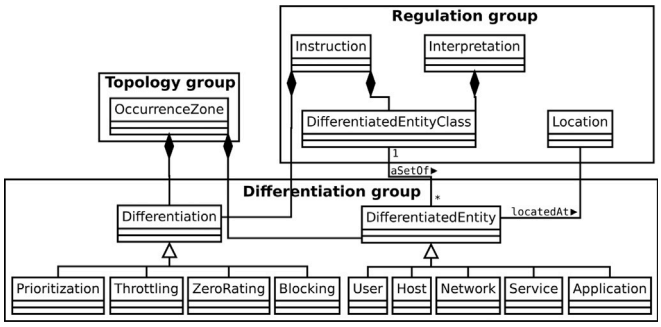


Fig. 4. Information model — Differentiation classes.

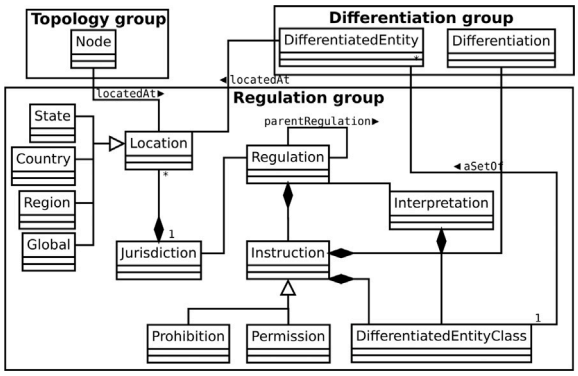


Fig. 5. Information model — Regulation classes.

and precision of the placement of the TD performed by the state-of-the-art solutions. Therefore, it may be composed of several Nodes and Links or just a single Node. The OccurrenceZone class is also composed of the DifferentiatedEntity and Differentiation classes presented in the Differentiation group, representing who is differentiated and what kind of differentiation is facing. Therefore, the OccurrenceZone is a portion of the topology identified by the state-of-the-art solution where a DifferentiatedEntity faces Differentiation.

In Fig. 4, the Differentiation group of the information model is presented. The *Differentiation* class represents all sorts of TDs that may affect communications. Through generalizations, it is possible to include new types of TDs. The Differentiation is part (composition) of the OccurrenceZone class from the Topology group. It is also part of the Instruction class, presented in the Regulation group. The *DifferentiatedEntity* class represents all sorts of entities that may have the communication affected. Through generalizations, it is also possible to include new types of DifferentiatedEntities. It is part of the OccurrenceZone class from the Topology group.

In Fig. 5, the Regulation group of the information model is presented. The *Location* class represents the geographical areas where jurisdictions may be established. There is an implicit hierarchy among these locations: Global, Region, Country, and State. Although there are no enforced regulations of Global scope nowadays, it could represent approaches such as Digital Constitutionalism [19]. However, even these Global regulation efforts may require agreements from countries, which is hard to achieve Global coverage in practice. However, the *Global* class is also used to model first-order constitutional principles (*e.g.*, *"everything which is not forbidden is allowed", "everything which is not allowed is forbidden"* principles) that are used to organize legal systems. The *Jurisdiction* class is a composition of Location classes.

The *Regulation* class is related to one Jurisdiction class. The Regulation class has a self-relation (parentRegulation) to express the regulation hierarchy: state regulations are related to country regulations, country regulations may be related to regional or global regulations, and global regulations are at the top of the hierarchy (they relate to themselves). The Regulation class is composed of *Instruction* classes that express the individual regulatory instructions that compose the regulation. These Instructions can be of two types: *Prohibition* or *Permission*. They are related to DifferentiatedEntityClass and a Differentiation. For instance, the regulatory instruction may express: it is "prohibited" to "blocking" "lawful content", which are the Instruction, Differentiation, and DifferentiatedEntityClass, respectively. The *Interpretation* class represents which DifferentiatedEntities are interpreted/considered of one DifferentiatedEntityClass for one regulation. For instance, what DifferentiatedEntities are considered "lawful content" in the example.

In the next section, the proposed solution that uses this information model to signal NN violations according to the regulation is presented.

4. JurisNN architecture and judgment algorithms

In this section, the proposed solution for the detection of NN violations following definitions from the NN regulation, named JurisNN, is presented. In Section 4.1, the conceptual architecture of JurisNN is presented. In Section 4.2, a few examples of Judgment Algorithms to judge TDs as NN violations are presented.

4.1. JurisNN architecture

In this subsection, the JurisNN architecture is presented. The architecture is depicted in Fig. 6, reflecting the presented modeling goals (Section 3.1). Therefore, the module descriptions presented next are short because they were already discussed earlier.

The Application Programming Interface (API) module is responsible for all external interactions with JurisNN, including TD detection and positioning solutions and the wrappers that may be developed to collect information from these solutions and submit it to JurisNN for the Regulation Assessment step. The API is also used by the User Interface module that users may use to interact with the system. Therefore, the API needs to interact with other modules to expose their functionalities.

The Topology module is responsible for keeping the information about the nodes, the links, and their relationships. The Occurrence module is responsible for the information about the TDs, which includes the kind of detected TD, the entity that is suffering the TD, the communication graph that references nodes and links from the Topology, and occurrence zones that reference nodes and links from the communication graph. The Regulation module is responsible for the information about the regulations registered in the system, including their jurisdictions, instructions, and interpretations.

The Judgment module is responsible for the regulation assessment processing using information from the Topology, Occurrence, and Regulation modules. The regulation assessment depends on definitions about how jurisdictions are established, as discussed in Section 2.1. When the TD positioning is not precise, the occurrence zone may span multiple jurisdictions without pointing to the exact one where the TD is deployed. In such a case, the system may not signal precisely whether the TD may be considered an NN violation. Therefore, there is a level of uncertainty in the regulation assessment processing. In order to accommodate these nuances, the regulation assessment is delegated to Judgment Algorithms detailed in the following subsection.

4.2. Judgment algorithms

In this subsection, a few examples of Judgment Algorithms are detailed. These algorithms are responsible for accommodating the subjectivity and uncertainty related to the judgment of TDs as NN violations according to the regulations. One source of subjectivity and uncertainty

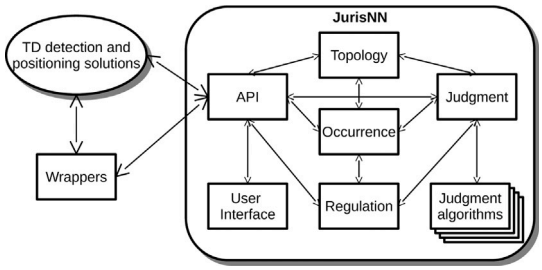


Fig. 6. JurisNN architecture.

is the jurisdiction establishment, as discussed in Section 2.1. The jurisdiction may be assessed by observing several factors, such as where the effects of a TD were felt or where the harmful action took place. Thus, the suitability of each algorithm concerning jurisdictional issues needs to be assessed before applying its judgment. Therefore, the first step of one Judgment Algorithm is assessing the TD occurrence to decide whether it is suitable to be judged by the algorithm, which is named *admissibility control*. Another source of subjectivity and uncertainty is the judgment itself, which may be affected by several factors, such as the inaccuracy of the TD placement or interpretations of the regulatory instructions. The assessment of such factors is the central part of the Judgment Algorithm, whose step is named *judgment*. It is important to note that jurisdictional issues affect admissibility control and judgment steps. Therefore, the concerns about jurisdictions span the whole algorithm. It is also important to point out that distinct Judgment Algorithms may have similar processing to evaluate the criterion used to cope with jurisdictional issues. Next, a few auxiliary functions that Judgment Algorithms may use to perform the processing are presented.

4.2.1. Jurisdiction establishment criteria and auxiliary functions

The Judgment Algorithms are divided into two steps: admissibility control and judgment. These steps are very dependent on the criterion used to establish jurisdiction, such as where the effects of a TD were felt or where the harmful action took place. However, different Judgment Algorithms may adopt the same criterion, requiring similar processing to judge TDs as NN violations. Next, a few auxiliary functions that implement these criteria are presented. Judgment Algorithms may use such functions to perform the admissibility control and judgment steps following the above criteria.

Procedure 1 EffectsFeltWithinLocalityCountry

```
Input: Occurrence, OccurrenceZone, Locality
 1: countries ← ∅
 2: differentiated_entity ← getDifferentiatedEntity(Occurrence, OccurrenceZone)
 3: if differentiated_entity.type ∈ {User, Host} then
 4:     countries ← countries ∪ {getCountry(differentiated_entity.located_at)}
 5: end if
 6: if differentiated_entity.type ∈ {Network, Service, Application} then
 7:     communication_graph ← getCommunicationGraph(Occurrence)
 8:     source ← getSource(communication_graph)
 9:     destination ← getDestination(communication_graph)
10:     countries ← countries ∪ {getCountry(source.located_at)}
11:     countries ← countries ∪ {getCountry(destination.located_at)}
12: end if
13: if getCountry(Locality) ∈ countries then
14:     return true
15: else
16:     return false
17: end if
```

The admissibility control step needs to evaluate whether the jurisdiction of a TD can be attributed to a specific locality according to the jurisdiction establishment criteria (the place of the injury or the place of the harmful action). By the place of the injury criterion, the place where the effects of the TD is considered, as presented in the EffectsFeltWithinLocalityCountry function (Procedure 1). The place where the effects of a TD are felt depends on the type of differentiated entity. When users and hosts are differentiated, the effects of the TD are felt by themselves. Therefore, only their locations are considered (lines 3–5). However, when networks, services, or applications are differentiated, the effects of the TD are felt by the endpoints of the communication. For instance, when an application is differentiated, its provider and users are affected. Therefore, the locations of the communication endpoints (source and destination) are considered (lines 6–12). As the jurisdictional rules are established at the country level (*e.g.*, the Targeting test is adopted in the US, the German test is adopted in Germany (DE)), the locations are considered at the country level, even when the locality is a state. If the country of the Locality is in the set of countries (line 13), then it is true that the effects of the TD are felt within the Location (line 14). By the place of the harmful action criterion, the place of the nodes in the occurrence zone is considered. Such processing is done by the HarmfulActionWithinLocalityCountry function, which was omitted for the sake of simplicity. Indeed, its processing is similar to Procedure 1 but building a set of countries of the nodes within the occurrence zone and testing whether the Locality country is within the set of countries.

Procedure 2 IsViolationByRegulation

```
Input: TD, regulation
 1: is_prohibited ← false; is_permitted ← false; is_defined ← false;
 2: instructions ← getInstructions(regulation)
 3: differentiated_entity_classes ← getEntityClasses(TD, regulation)
 4: for all instruction ∈ instructions do
 5:     if instruction.regulation_class ∈ differentiated_entity_classes then
 6:         if instruction.type = Prohibition then
 7:             is_prohibited ← true; is_defined ← true;
 8:         end if
 9:         if instruction.type = Permission then
10:             is_permitted ← true; is_defined ← true;
11:         end if
12:     end if
13: end for
14: result ← is_prohibited and not is_permitted
15: return result, is_defined
```

The judgment step needs to evaluate whether the TD is considered an NN violation according to the regulation, which is helped by a two auxiliary functions presented next. The first auxiliary function is IsViolationByRegulation (Procedure 2), which assesses whether the TD is considered an NN violation considering what is established in one specific regulation. This assessment considers the interpretations stated for the regulation. For instance, one regulation may point out that "blocking" is prohibited, but the "blocking" of "unlawful content" is allowed, and therefore it is not considered a violation. However, the regulation does not enumerate what is considered "unlawful content", requiring interpretations to clarify when one regulatory instruction applies to a specific TD by evaluating its DifferentiatedEntity. The getEntityClasses(TD, regulation) (line 3) returns the classes that the differentiated entity is classified following the regulation interpretations. This classification uses attributes of the differentiated entities (*e.g.*, users, applications) that indicate that they are a member of a regulation class. For instance, if the regulation considers Peer-to-Peer (P2P) applications as "unlawful content", one application could have an attribute "type = P2P" to indicate this. For all regulatory instructions (lines 4–13), it is checked if the instruction is related to the classes that the differentiated entity pertains to (line 5). If the instruction is about one of the regulation classes of the differentiated entity, the boolean variables are adjusted to specify if the instruction establishes prohibition (lines 6–8) or permission (lines 9–11). If the TD is prohibited and there is no permission established, it is considered an NN violation by the regulation (line 14).

The second auxiliary function, IsViolationInLocality (Procedure 3), assesses whether the TD is considered an NN violation considering all regulations that one locality may be subjected to. The evaluation looks for the regulations that may be applied for that locality following the bottom-up approach from state-, national-, regional-, and global-level regulations (line 1 order). The getRegulationHierarchy(Locality)

Procedure 3 IsViolationInLocality

```
Input: TD, Locality
 1: regulation_levels ← [state, country, region, global]
 2: regulation_hierarchy ← getRegulationHierarchy(Locality)
 3: is_violation ← false
 4: for all level ∈ regulation_levels do
 5:     if regulation_hierarchy[regulation_level] is defined then
 6:         regulation ← getRegulation(regulation_hierarchy[regulation_level])
 7:         is_violation, is_defined ← IsViolationByRegulation(TD, regulation)
 8:         if is_defined then
 9:             return is_violation
10:         else
11:             continue
12:         end if
13:     end if
14: end for
15: return is_violation
```

(line 2) returns an array with the established regulation for each level. For each level, the IsViolationByRegulation function (Procedure 2) is invoked (line 7) to check whether the TD is considered a violation for that regulation. The processing stops when one regulation specifies whether the TD is an NN violation (is_defined = true, line 8). Therefore, the regulations from the lower levels (smaller scope) are preferred, which is usual in legal systems. If no regulation gives a verdict whether the TD is an NN violation, it is not considered an NN violation (line 15). These auxiliary functions are used by the functions that implement the judgment step following each jurisdiction establishment criteria.

Procedure 4 IsViolationByPlaceOfInjury

```
Input: Occurrence, OccurrenceZone, Locality
 1: differentiated_entity ← getDifferentiatedEntity(Occurrence, OccurrenceZone)
 2: TD ← getTD(Occurrence, OccurrenceZone)
 3: if differentiated_entity.type ∈ {User, Host} then
 4:     is_violation ← IsViolationInLocality(TD, differentiated_entity.located_at)
 5: end if
 6: if differentiated_entity.type ∈ {Network, Service, Application} then
 7:     communication_graph ← getCommunicationGraph(Occurrence)
 8:     source ← getSource(communication_graph)
 9:     destination ← getDestination(communication_graph)
10:     is_violation_source ← false; is_violation_destination ← false
11:     if getCountry(source.located_at) == getCountry(Locality) then
12:         is_violation_source ← IsViolationInLocality(TD, source.located_at)
13:     end if
14:     if getCountry(destination.located_at) == getCountry(Locality) then
15:         is_violation_destination ← IsViolationInLocality(TD, destination.located_at)
16:     end if
17:     is_violation ← is_violation_source or is_violation_destination
18: end if
19: return is_violation
```

By the place of the injury criterion, the function for the judgment step is named IsViolationByPlaceOfInjury (Procedure 4). Its structure is similar to the procedure of the admissibility control for the same criterion (Procedure 1) because it also varies accordingly to the differentiated entity. When users or hosts are differentiated, their locations are used in the assessment (lines 3–5). When networks, services, or applications are differentiated, the locations of the communication endpoints are used instead. However, their locations are used in the judgment process instead of just their countries. For instance, if the location is a state, this state may have its NN regulation that must be considered in the assessment. The source and destination country are rechecked (lines 11 and 14) because the TD may have been admitted because only one endpoint is within the country that adopts the criterion, but the criterion only applies to the endpoint within that country. It is important to note the auxiliary function IsViolationInLocality (Procedure 3) invocation in lines 4, 12, and 15. This function returns a boolean that indicates if the TD is considered an NN violation or not. Noteworthy that Procedures 1 and 4 are specific for the place of injury criterion. They use the locations of the endpoints or the differentiated entities, which are known and do not impose any uncertainty on the judgment process. Therefore, the result whether the TD is an NN violation by this criterion is conclusive by design.

By the place of the harmful action criterion, the judgment process needs to evaluate the location of nodes of the occurrence zone. In turn, the process needs to accommodate situations where the occurrence zone spans multiple localities with distinct regulatory instructions for the same TD. For instance, the TD is considered an NN violation in one locality and may not be considered a violation in another. The function named IsViolationByPlaceOfHarmfulAction (Procedure 5) computes the result of the judgment step for this criterion. For each locality within the occurrence zone, it counts how many nodes are located there (getLocationCounts(), line 2). For each location, it invokes the auxiliary function IsViolationInLocality (Procedure 3) to assess whether the TD is considered a violation (line 5). As one locality may consider the TD an NN violation while others do not, the judgment process needs to account for the similarity of the verdicts in the occurrence zone. It uses an adaptation of the Jaccard similarity index [20] to compare the number of nodes with the same verdict against all nodes within the occurrence zone (line 11). The max function is used because the best similarity index among the two verdicts possible represents the similarity of the occurrence zone. When the occurrence zone similarity index is 1.0, the verdict is conclusive because the verdict is the same for all nodes within the occurrence zone. Otherwise, the regulation for some nodes in the occurrence zone may disagree about the verdict and, therefore, the TD cannot be judged as an NN violation or not.

Procedure 5 IsViolationByPlaceOfHarmfulAction

```
Input: Occurrence, OccurrenceZone
 1: is_violation_count ← 0; is_not_violation_count ← 0
 2: location_counts ← getLocationCounts(Occurrence, OccurrenceZone)
 3: TD ← getTD(Occurrence, OccurrenceZone)
 4: for all locality ∈ location_counts do
 5:     if isViolationInLocality(TD, locality) then
 6:         is_violation_count += location_counts{location}
 7:     else
 8:         is_not_violation_count += location_counts{location}
 9:     end if
10: end for
11: OZ_similarity ← max(is_violation_count, is_not_violation_count) / (is_violation_count + is_not_violation_count)
12: if is_violation_count ≥ is_not_violation_count then
13:     is_violation ← true
14: else
15:     is_violation ← false
16: end if
17: return OZ_similarity, is_violation
```

These functions implement the two criteria discussed in Section 2.1 for the steps performed by Judgment Algorithms. Next, the functions used by each Judgment Algorithm to perform each step are listed.

4.2.2. The judgment algorithms

The functions presented previously are used to implement the steps of Judgment Algorithms. Each Judgment Algorithm adopts distinct criteria to establish jurisdiction and decide whether the TD is an NN violation. In Table 1, the functions used by each Judgment Algorithm (Place of TD deployment, Targeting test, and German test) in each step (admissibility control and judgment) are listed.

The Place of TD deployment: the straightforward way of judging TDs as NN violations considering the regulation is using the place where the TD is deployed to establish the jurisdiction, *i.e.*, the place where the harmful action took place. As almost all courts around the world accept this criterion, the admissibility control step does not need to check conditions about the TD occurrence. Case one jurisdiction does not accept this criterion, the admissibility control could be amended to exclude TD occurrences that span such jurisdiction using the HarmfulActionWithinLocalityCountry function. The judgment process uses the IsViolationByPlaceOfHarmfulAction function (Procedure 5). As may have a level of uncertainty on the results of this function due to the occurrence zone characteristics, the Judgment Algorithm needs to evaluate the similarity index of the verdicts in the occurrence zone. If the similarity index is 1.0, the algorithm can signal the verdict as conclusive. Otherwise, the verdict is considered inconclusive.

Table 1
Auxiliary functions used by judgment algorithms.

Auxiliary function	Judgment algorithms		
	Place of TD deployment	Targeting test	German test
Admissibility control step:			
EffectsFeltWithinLocalityCountry		✔	✔
HarmfulActionWithinLocalityCountry	✔		✔
Judgment step:			
IsViolationByPlaceOfInjury		✔	✔
IsViolationByPlaceOfHarmfulAction	✔		✔

The Targeting test: the US courts adopt tests to decide whether they have the competence to judge a claim based on where the effects of action occur. For instance, if one organization located at a jurisdiction A targets its business to a jurisdiction B, the organization acts that affect clients/users in jurisdiction B can be claimed in jurisdiction B. A few tests are based on the effects of action (*e.g.*, Zippo, Calder, and Targeting tests). The Judgment Algorithm adopting the Targeting test is detailed. Since this test applies to the US, the admissibility control uses the EffectsFeltWithinLocalityCountry function to evaluate whether the effects of the TD are felt within the US. The judgment step uses the IsViolationByPlaceOfInjury function to achieve the verdict about the TD. As this criterion is based on the locations of the endpoints or differentiated entities, which are known and do not impose any uncertainty, the results are conclusive by design.

The German test: in Germany, the jurisdiction may be established by two criteria: where the harmful action happened and where the injury was incurred. Therefore, Germany adopts, at the same time, the criteria adopted by the above two Judgment Algorithms. The admissibility control is the junction of the functions EffectsFeltWithinLocalityCountry (Procedure 1) to evaluate the place of the effects of the TD criterion and the function HarmfulActionWithinLocalityCountry to evaluate the place of the harmful action criterion. In the end, if Germany is found as the country of the affected entities (where the injury was incurred) or as the country of at least one node within the occurrence zone (the place of the harmful action), then the admissibility control accepts to judge the TD. The same happens with the judgment process that is also the junction of the two functions to evaluate whether the TD is an NN violation by each criterion: IsViolationByPlaceOfInjury and IsViolationByPlaceOfHarmfulAction (Procedures 4 and 5, respectively). This algorithm also must account for the level of uncertainty on the results of the IsViolationByPlaceOfHarmfulAction function, evaluating the similarity index of the occurrence zone before signaling whether the TD is an NN violation when such criterion is adopted.

In this subsection, a few examples of Judgment Algorithms are presented. They represent different approaches to establishing jurisdictions and judging TDs as NN violations considering the regulation. These Judgment Algorithms output their decision that may be accompanied by one metric to indicate its level of certainty on the verdict (*e.g.*, similarity index). Other algorithms could be designed using these same criteria to establish jurisdictions, but that outputs a different metric using further information or that processes the TD information in another way. Other algorithms could also be designed to accomplish other criteria to establish jurisdictions. The point is that given the multiple factors that affect the judgment process, it is not expected that a unique Judgment Algorithm could accommodate all these possibilities.

In the next section, an evaluation of the Judgment Algorithms focusing on the conclusiveness (the TD is an NN violation or not) of their results using the information provided by TD detection solutions is presented. This evaluation uses TD information of a dataset collected by a TD detection solution from the state-of-the-art.

5. Evaluation

In this section, JurisNN is evaluated using information about TDs detected by Wehe [4]. The evaluation aims to answer whether the TD detection solutions provide enough information to properly judge TDs as NN violations using the NN definitions from regulations. For this evaluation, a prototype following the architecture depicted in Fig. 6 was developed, which is briefly presented in Section 5.1. In Section 5.2, the dataset of the TDs detected by Wehe is presented along with the required steps to complement it with network paths collected using the RIPE Atlas platform. In Section 5.3, the results achieved by each Judgment Algorithm are presented along with a discussion about the caveats of the performed analysis.

5.1. JurisNN prototype

In this subsection, the JurisNN prototype that implements the conceptual architecture depicted in Fig. 6 is presented.

One investigation was conducted to evaluate which existing data models could be suitable to represent the information models presented in Section 3. Due to the capacity of representing unidirectional links and easiness of extensibility, the models were represented using YANG data models [21]. Two YANG models were designed for JurisNN: nn-regulation and tdo. The nn-regulation model is designed to represent the NN regulations and related information, such as the jurisdictions, the regulatory instructions, the regulatory interpretations, the traffic differentiations, and the entities who suffer the differentiations. This model does not rely on information from existing IETF's YANG models. The tdo model is designed to represent the occurrence zones and related information, such as the network topology, the communication graph, and elements of the occurrence zones (nodes and links where the differentiation is taking place). This model relies on information from ietf-network [22], ietf-network-topology [22], and ietf-l3-unicast-topology [23] models.

JurisNN was implemented using the JetConf framework [24]. This framework is an implementation of the RESTCONF protocol [25] based on Python [26]. This framework allows to inform the YANG modules to be used and provides the basic functionality to perform CRUD operations in the Datastore based on the data structure of the modules. It is important to note that RESTCONF is not required for the JurisNN solution. However, as the JetConf framework implements several facilities to build solutions based on YANG models, and the interaction with this component may be based on REST, it was used to build the prototype. However, the conceptual architecture depicted in Fig. 6 could be implemented using another framework. In turn, the Judgment Algorithms are implemented in Python.

5.2. Evaluation data

In this subsection, the data used in the JurisNN evaluation concerning the conclusiveness of judgment results is presented. The detected TD information was collected from the Wehe dataset [27]. As this dataset does not provide the network path between the clients and the servers of tests, the RIPE Atlas platform was used to collect traceroutes between the client and server Autonomous Systems (ASs). As the Wehe tests traversed multiple jurisdictions, the NN definitions found in these jurisdictions are also presented. The details of how these data were used are presented along with this subsection.

The Wehe solution provides a dataset of conducted tests within the M-Lab [28] since October 2020 and within its website [29] since November 2018. In the dataset provided through the website, the amount of available data fluctuates along with time. In the dataset provided through the M-Lab, the amount is stable after December 2020. This analysis considered the data that was collected from January 1, 2021, until February 28, 2021, and publicized in the M-Lab platform [27] consisting of 170 GiB of data about 77 270 tests.

Wehe authors provide two codes to perform the dataset analysis [29]. The first code processes the dataset and creates a folder structure separating the tests by the client ISP, based on the AS responsible for its IP. The second code performs the TD detection, characterizing the tests as True Positives, True Negatives, False Positives, and False Negatives for each ISP. One script was developed to collect the information about the tests that detected TDs (True Positives and False Negatives), although only True Positives were found.

For the analysis performed by the JurisNN prototype, both the communication graph (nodes and links used to perform the communication) and the occurrence zone information (the nodes and links where the TD was detected) are required. However, the Wehe does not collect the network path between the client and the server of the tests nor perform the TD positioning to establish the occurrence zone. In order to complement the Wehe dataset with network paths between client and servers, the collection of traceroutes was performed using the RIPE Atlas platform [5]. It is a platform composed of probes (hardware and software) to perform network measurements such as ping, traceroute, Domain Name System (DNS) resolution, and Secure Sockets Layer (SSL), Hyper Text Transfer Protocol (HTTP), and Network Time Protocol (NTP) requests. The probes used to perform a test can be selected by criteria such as their area in the globe, country, prefix, or Autonomous System Number (ASN). The platform provides a Python API named Cousteau [30] to request tests.

Information about each TD detected by Wehe was used to collect network paths. The Wehe analysis code provides the /24 subnet of the client IP, but it does not provide the server's IP that processed the client test. However, the Packet Capture (PCAP) files of each test are available within the dataset. One script was developed to extract the server IP from the respective PCAP file. In Table 2, the number of detected TDs for each pair of source ASN (derived from the client subnet) and the destination ASN (derived from the server IP) are presented, receiving an id that is used to identify the source/destination ASN pairs in the results along with this section. The ASNs were retrieved using the PyASN [31] using the Routing Information Base (RIB) information from 2021-02-01 (middle of the time interval).

The Python Cousteau API was used to request traceroute tests from each source ASN from Table 2 to one server IP in the corresponding destination ASN. Preliminary tests using the client /24 prefix to select probes do not yield satisfactory results. Therefore, the probe selection was performed by the source ASN, which also has not found network paths for a few source/destination ASN pairs due to the lack of probes. The traceroute measurements were requested for one week (from 2021-04-12 until 2021-04-19) six times a day (0:00, 4:00, 8:00, 12:00, 16:00, and 20:00 GMT) resulting in 42 requests for each source/destination ASN pair. The resulting traceroutes were compared to only consider different traceroutes between the source/destination ASN pair in the analysis. For the traceroute comparison, the internal network addresses (10.0.0.0/8, 172.16.0.0/16, and 192.168.0.0/16) were discarded from the traces. Traceroutes with the same hop indexes associated with the same addresses are considered equal. Otherwise, they are considered different. In Table 2, the number of different traces found by the RIPE Atlas platform for each source/destination ASN pair is presented. It is possible to note that among the 42 collected traces, most of the source/destination ASN pairs have less than ten different traces. Only the pair id 1 has a higher number of different network paths (36).

In Table 3, the jurisdictions found as source (based on the client /24 subnet), destination (based on the server IP), and along with the network path (based on the traceroutes) are presented. The Geo Localization was performed using the MaxMind GeoLite2 database [32]. The database has the "most specific" location for some IP addresses, which was used to determine state-level jurisdictions (*e.g.*, US-CT, US-HI). Otherwise, the jurisdiction was considered at the country level.

In Table 3, it is possible to note that most of the jurisdictions found (source, destination, and path) are within the US or US states. Many US states were identified as source jurisdictions, given that the identified ASs has a presence in multiple US states. Some of the jurisdictions found are in the EU. In addition, Canada (CA) and the United Kingdom (GB) also were found. It is possible to note also that a few ASN pair ids point out that Wehe tests were originated in one jurisdiction and were processed by a server in a far jurisdiction. For instance, for ASN pair id 3, the Wehe tests were originated in the US, and the server is located in Sweeden (SE). For ASN pair id 8, the tests were originated in Poland (PL), and the server is located in the US, but the packets also traversed Germany (DE). Similar situations also can be noted in ASN pair ids 15, 16, 20, 29, 30, 31, and 32. The fact of tests traversing multiple jurisdictions (that may be far away in other countries or even maybe close like a neighbor US state) exposes the possibility of these traffics crossing distinct regulatory frameworks with different NN definitions.

Table 2
AS pair ids involved in TDs, the number of detected TDs, number of traces collected, and occurrences submitted to JurisNN.

id	Src. ASN	Dst. ASN	TDs	Traces	Occurrences (TDs x Traces)
1	6167	174	21	36	756
2	6167	1280	8	12	96
3	6167	1299	17	9	153
4	6167	3257	11	5	55
5	6167	3356	20	6	120
6	6167	6453	15	6	90
7	6167	6461	24	6	144
8	12912	3257	1	3	3
9	20057	174	7	3	21
10	20057	3257	7	0	0
11	20057	3356	8	7	56
12	20057	6453	9	3	27
13	20057	6461	4	5	20
14	21928	174	45	20	900
15	21928	1280	1	6	6
16	21928	1299	24	10	240
17	21928	3257	36	5	180
18	21928	3356	34	5	170
19	21928	6453	63	4	252
20	21928	6461	31	6	186
21	21928	6939	3	4	12
22	22394	174	13	0	0
23	22394	1280	2	0	0
24	22394	1299	7	0	0
25	22394	3257	8	0	0
26	22394	3356	7	0	0
27	22394	6453	17	0	0
28	22394	6461	10	0	0
29	57269	1299	4	3	12
30	57269	3257	3	2	6
31	57269	3356	1	2	2
32	57269	6453	8	5	40

In Table 4, the regulatory instructions about throttling (the TD type detected by Wehe) are presented for the jurisdictions traversed by Wehe tests, whose definitions are detailed next.

Some jurisdictions allow throttling. The allowance may be explicit as in CA that allows throttling since justifiable [33]. Alternatively, the allowance may be implicit given to the lack of explicit prohibitions established *a priori* as in the US [8]. In both cases, ISPs are entitled to perform the traffic management practices deemed necessary.

The lack of country-level regulation in the US opened the opportunity for US states to establish their NN regulations. The National Conference of State Legislatures (NCSL) maintains a website [34] organizing the NN regulation efforts conducted by US states, which was used in this research. Several states started discussions around the subject on their legislative organizations. However, only three states (California, US (US-CA), Columbia, US (US-CO), and Vermont, US (US-VT)) finished the legislative proceedings of such regulations, reestablishing the Federal Communications Commission (FCC) Open Internet rules from 2015 [35], which prohibits the throttling of lawful content, applications, services, and devices.

Table 3
The jurisdictions found in source (**S**), destinations (**D**), and network path (**P**)

id	Jurisdictions
1	**S** (US, US-CT, US-HI), **D** (US), **P** (US, US-MA)
2	**S** (US, US-HI), **D** (US), **P** (US, US-MA)
3	**S** (US, US-TN, US-UT, US-VA), **D** (SE), **P** (SE, US, US-MA)
4	**S** (US, US-FL, US-PA, US-TN, US-UT), **D** (US), **P** (US, US-MA)
5	**S** (US, US-CT, US-MA, US-PA), **D** (US, US-VT), **P** (US, US-MA)
6	**S** (US, US-CT, US-FL, US-TN, US-VA), **D** (US), **P** (US, US-MA)
7	**S** (US, US-FL, US-HI, US-MA, US-UT), **D** (US), **P** (US, US-MA)
8	**S** (PL), **D** (US), **P** (DE, DE-HE, PL, US)
9	**S** (US, US-FL, US-OH), **D** (US), **P** (US, US-CA, US-NC)
10	**S** (US-CA, US-FL), **D** (US), **P** (–)
11	**S** (US, US-CA, US-FL, US-TX), **D** (US, US-FL), **P** (US, US-CO, US-FL, US-NC)
12	**S** (US, US-FL, US-GA, US-TX), **D** (US, US-FL, US-GA, US-TX), **P** (US, US-CA, US-FL, US-NC)
13	**S** (US-FL, US-TX), **D** (US, US-IN), **P** (US, US-AZ, US-CA, US-NC)
14	**S** (US-AZ, US-CA, US-FL, US-IL, US-KY, US-MA, US-MI, US-NV, US-NY, US-PA, US-TX, US-WA), **D** (US), **P** (US, US-RI)
15	**S** (US-CA), **D** (US), **P** (SE, US, US-RI)
16	**S** (US, US-AL, US-CO, US-FL, US-NC, US-NY, US-TX, US-UT, US-WA), **D** (SE), **P** (SE, US, US-RI)
17	**S** (US-CO, US-FL, US-IL, US-IN, US-MA, US-NC, US-NY, US-PA, US-WA), **D** (US), **P** (US, US-RI)
18	**S** (US-CA, US-FL, US-IL, US-IN, US-KY, US-MA, US-NY, US-PA, US-TX, US-VA, US-WA), **D** (US), **P** (US, US-RI)
19	**S** (US, US-AL, US-CA, US-FL, US-GA, US-IL, US-IN, US-MA, US-MN, US-NC, US-NY, US-TX, US-WA), **D** (US), **P** (US, US-RI)
20	**S** (US-AZ, US-CA, US-CO, US-FL, US-IN, US-MA, US-MI, US-PA, US-TX), **D** (US), **P** (CA, ES, GB, IE, US, US-RI)
21	**S** (US-MI), **D** (US), **P** (US, US-RI)
22	**S** (US-CA, US-IL, US-IN, US-NY, US-OH), **D** (US), **P** (–)
23	**S** (US-CA), **D** (US), **P** (–)
24	**S** (US-GA), **D** (SE), **P** (–)
25	**S** (US-IL, US-NY), **D** (US), **P** (–)
26	**S** (US-CA, US-IN, US-NY, US-TX), **D** (US), **P** (–)
27	**S** (US-CA, US-GA, US-IL, US-IN, US-NY, US-OH, US-VA), **D** (US), **P** (–)
28	**S** (US-CA, US-IN, US-NY, US-OH), **D** (US), **P** (–)
29	**S** (ES), **D** (SE), **P** (ES, RO-TR, SE)
30	**S** (ES), **D** (US), **P** (ES, GB, HU-JN, US)
31	**S** (ES), **D** (GB), **P** (ES, GB, RO-TR, US)
32	**S** (ES), **D** (IE), **P** (CA, ES, GB, IE, RO-TR, SE, US)

Table 4
Regulatory instructions about throttling valid on the jurisdictions from 2021-01-01 onward.

Jurisdiction	Throttling	Jurisdiction	Throttling
CA	✔*	US-IL	–
DE	✗*↑	US-IN	–
DE-HE	✗*↑	US-KY	–
ES	✗*↑	US-MA	–
GB	✗*	US-MI	–
HU-JN	✗*↑	US-MN	–
IE	✗*↑	US-NC	–
PL	✗*↑	US-NV	–
RO-TR	✗*↑	US-NY	–
SE	✗*↑	US-OH	–
US	–	US-PA	–
US-AL	–	US-RI	–
US-AZ	–	US-TN	–
US-CA	✗*	US-TX	–
US-CO	✗*	US-UT	–
US-CT	–	US-VA	–
US-FL	–	US-VT	✗*
US-GA	–	US-WA	–
US-HI	–		

Legend: allowed (✔), prohibited (✗), upper-level regulation (↑), none regulation (–), has exceptions (*).

Some jurisdictions found as the source, destination, and network paths do not have local NN regulations. However, they are within the jurisdiction of a broader upper-level NN regulation. All the jurisdictions in this situation are within the EU jurisdiction (DE, Hessen, DE (DE-HE), Spain (ES), Jász-Nagykun-Szolnok, Hungary (HU-JN), Ireland (IE), PL, Teleorman, Romania (RO-TR), and SE) under the Body of European Regulators for Electronic Communications (BEREC) regulation. This regulation establishes that throttling is prohibited, but it may be allowed under certain situations that the National Regulatory Authority (NRA) shall evaluate and decide [36].

The Brexit process has finished on December 31, 2020. Therefore, the GB left the EU on January 1, 2021 (the beginning time interval considered in the dataset). However, Ofcom (the British NRA) still is applying the EU regulation [37].

In this subsection, the data used in the JurisNN evaluation was presented, which consists of the detected TD information (provided by the Wehe dataset), the network paths between the client and server ASs (provided by the RIPE Atlas platform), and the NN definitions found on the jurisdictions traversed by Wehe tests. In the next subsection, the details of how this data was used to evaluate JurisNN regarding the judgments conclusiveness are presented.

5.3. JurisNN judgment results

This subsection details how the data presented in the previous subsection was used to evaluate JurisNN regarding the judgment results. The NN definitions summarized in Table 4 were introduced into the regulation datastore of JurisNN.

For each TD detected by Wehe, the JurisNN was used to judge whether such TD is considered an NN violation accordingly to the regulation established in the jurisdictions traversed. As for most TDs, multiple network paths were found between the client and server ASs using the RIPE Atlas platform, each TD detected by Wehe was associated with each network path found by RIPE Atlas. In order to evaluate a TD Occurrence, information about the communication topology and the TD itself are submitted to JurisNN, which are detailed next.

Each network path associated with each TD was submitted to JurisNN as one ietf-network:networks/network. Each hop in the network path was submitted as one node in the ietf-network using the hop index as node-id and identified Geo Localization in the located-at attribute. The TD and the associated network path were used to compose one tdo:tdo-occurrences/occurrence. All the nodes in the network path were referenced in the communication-graph of the occurrence. Also, as the Wehe does not point were in the network path the TD was deployed, all the nodes in the communication-graph were referenced in the occurrence zone. Wehe detects the throttling of applications. Thus, one occurrence-zone about the Differentiation "Throttling" and the DifferentiatedEntity "Application" was submitted to JurisNN for each detected TD and network path.

The judgment of the TDs is performed by the Judgment Algorithms, presented in Section 4.2. These algorithms have two steps: admissibility control and the judgment itself. Next, the results achieved by the three Judgment Algorithms are presented.

The first algorithm is the Place of the TD deployment, which evaluates the NN definitions established in the place where nodes within the occurrence zone are deployed. This algorithm applies to any jurisdiction. Therefore, all TDs are accepted by the admissibility control step to be judged by this algorithm. In Fig. 7, the achieved results using box plots that represent the minimum, first quartile, median, third quartile, and maximum values of the distribution of the results are presented. The ASN pair ids presented in the x-axis were admitted by the algorithm, which are all the pairs submitted to JurisNN. The missing pair ids are those that RIPE Atlas has not identified network paths. Therefore, their related TDs were not submitted to JurisNN (ids 10, 22, 23, 24, 25, 26, 27, and 28).

The judgment by the Place of the TD deployment algorithm returns the occurrence zone similarity index, and the most prevalent verdict about the TD be an NN violation. The similarity index reflects the certainty in the verdict, in which similarity of 1.0 indicates that for all nodes within the occurrence zone, the verdict is the same. For the ASN pair ids 1, 2, 4, 6, 7, and 21, this similarity index is 1.0 for all

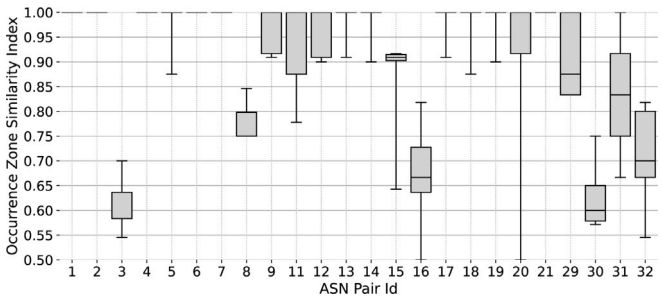


Fig. 7. The place of the TD deployment: results distribution.

occurrence zones analyzed. Looking for the jurisdictions traversed by the tests associated with these ASN pair ids in Table 3, all tests traversed jurisdictions in the US. However, none of these jurisdictions are those that reestablished the Open Internet rules (US-CA, US-CO, and US-VT). Therefore, the TDs associated with these ASN pair ids are not NN violations when the regulatory instructions are considered. For all other ASN pair ids, the similarity index ranges below 1.0, indicating that the achieved verdicts have an uncertainty level, hindering signaling whether the TD is an NN violation or not. The high similarity indexes for the ASN pair ids 5, 9, 11, 12, 13, 14, 17, 18, and 19 are explained because their tests traversed jurisdictions within the US but traversed the jurisdictions that reestablished the Open Internet rules. Therefore, the TD is considered an NN violation in part of the occurrence zone, and it is not considered in the other part. The high similarity index for ASN pair 29 is explained because its tests traversed countries in the EU, where the BEREC regulation is in place, prohibiting throttling. However, it was not possible for a few nodes to get their countries by the Geo Localization, hindering the verdict achievement for these nodes. The wide ranges in the similarity index for the ASN pair ids 15, 16, 20, 31, and 32 are explained because the tests traversed jurisdictions in US and EU, whose verdicts diverge about throttling. The same situation happens for ASN pair ids 3, 8, and 30, but with narrower similarity index ranges, indicating that network paths are a more homogeneous mix of jurisdictions traversed on the US and EU.

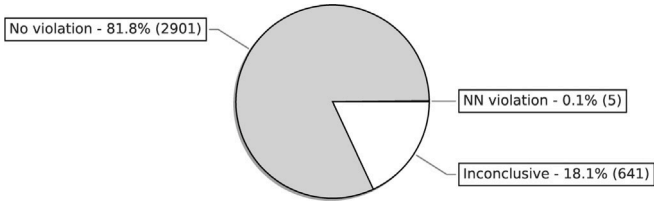


Fig. 8. Place of the TD deployment: verdicts distribution.

The results expose various situations in which, for most scenarios, it is not possible to point out whether the TD is considered an NN violation (except those that the similarity index is 1.0 without ranges in the box plots). The overall result is that only 6 (from 24) ASN pair ids scenarios achieved conclusive results, indicating the need for better TD positioning. However, the results of analyzing the TDs without grouping them in the client/server ASN pairs point to a high amount of conclusive verdicts. In Fig. 8, the distribution of verdicts is presented. The conclusive verdicts (No Violation and NN violation) sum 81.9% of the occurrences submitted to JurisNN. In turn, the inconclusive results sum 18.1% of the occurrences. However, most TDs detected by Wehe involved ASs within the US, where most jurisdictions allow the throttling of applications due to the lack of NN regulation. Therefore, the dataset may be biased towards situations where the TD is not considered an NN violation. Thus, the results grouping ASN pair ids as scenarios seem more representative. Therefore, considering the better representativeness of scenarios, this analysis still points to the need for better TD positioning to judge TD practices considering the regulatory instructions using the Place of the TD deployment algorithm.

Table 5
Targeting test: results distribution.

	Judgment			Judgment	
id	No violation	NN violation	id	No violation	NN violation
1	756	0	13	20	0
2	96	0	14	880	20
3	153	0	15	0	6
4	55	0	16	220	20
5	108	12	17	160	20
6	90	0	18	160	10
7	144	0	19	232	20
8	3	0	20	156	30
9	21	0	21	12	0
11	42	14	30	6	0
12	27	0			

The second algorithm is the Targeting test, which considers the location of the endpoints of the communication affected by the TD to establish the jurisdiction. The Wehe detects the throttling of applications. Therefore, the Targeting test only considered TDs whose the DifferentiatedEntity is Application. In such cases, the algorithm evaluates both endpoints (client and server) because the TD affects both the application provider and users. US courts adopts the Targeting test. Therefore, the admissibility control checks if the client or server are within the US to admit to judging the TD. In Table 5, the results listing the ASN pair ids admitted to being judged by the algorithm are presented. It is important to note that this algorithm does not depend on the TD positioning accuracy because only the jurisdictions in the endpoints are considered. Therefore, the algorithm returns conclusive results by design. For 12 ASN pair ids (1, 2, 3, 4, 6, 7, 8, 9, 12, 13, 21, and 30), the algorithm just pointed out TDs that are not considered NN violations. For 8 ASN pair ids (5, 11, 14, 16, 17, 18, 19, and 20), the algorithm pointed a few TDs that are considered NN violations, but most of their associated TDs are not considered NN violations. Only for the ASN pair id 15, all the TDs are considered NN violations because all the TDs have US-CA (that reestablished the Open Internet rules) as source jurisdiction. As this algorithm does not depend on the TD positioning, the information provided by the TD detection solutions was enough to judge the TDs according to the regulation.

The third algorithm is the German test, which applies both criteria adopted by the previous algorithms. The algorithm evaluates the jurisdictions in the endpoints (the place where the injury incurs) and along with the network path (the place where the harmful action takes place). Courts in DE apply this test. Therefore, only the TDs that have the endpoints in DE or that traversed the DE are admitted to be judged by the algorithm. In Fig. 9, the achieved results using box plots that represent the minimum, first quartile, median, third quartile, and maximum values of the distribution of the results are presented. Only the ASN pair id 8 was admitted to be judged because their associated TDs have the source in PL and destination in the US but traversed DE along with the network path. Therefore, the results were achieved by the place where the harmful action took place criterion, that returns the most prevalent verdict and the similarity index that reflects the verdict certainty. As the network path is part in the EU and part in the US, the similarity index ranges from 0.62 to 0.77, also indicating the divergence about the TD within the occurrence zones. Therefore, all these results are inconclusive, indicating also that TD positioning accuracy (the whole network path as occurrence zone) was not enough to judge the TDs by the German test algorithm.

The findings of the analysis are summarized now. The Targeting test algorithm does not depend on the jurisdictions traversed along with the network path because its judgment is based on the jurisdictions of the endpoints established by the Geo Localization of the client and server. Thus, the algorithm is not affected by the positioning of the TD. Therefore, the available information provided by the state-of-the-art solution was enough for the judgment by this algorithm. Both the Place

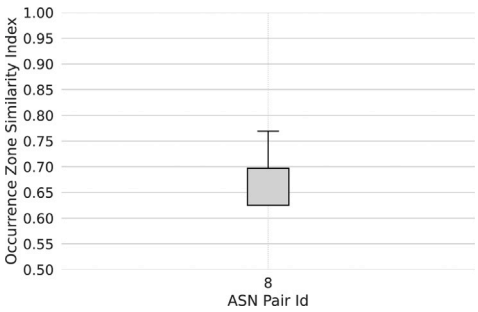


Fig. 9. German test: results distribution.

of the TD deployment and the German test algorithms depend on the network path information because they evaluate the place where the TD was deployed. The analysis shows that the judgment performed by such algorithms using the whole identified network path as the occurrence zone achieved inconclusive results (similarity index below 1.0). Therefore, it is required a better TD positioning (the whole network path is not enough) to point occurrence zones narrow to the actual TD deployment, to correctly judge the TDs as NN violations using the definitions established in the NN regulation. Indeed, Garrett et al. [17] propose a method to determine where the TD was deployed at AS-level. Therefore, it is expected that developers of solutions devoted to NN violation detection be encouraged to include into their solutions the functionalities to identify network paths and to position the TD. Indeed, the results of this article show that this is a requirement to achieve actionable evidence to help users support claims against unfair traffic management practices deployed by ISPs.

5.3.1. Analysis caveats

It is essential to point out the caveats of the presented analysis. Most of them are related to the lack of network path information provided by TD detection solutions and the Geo Localization process.

The Geo Localization was performed using the MaxMind GeoLite2 IP database [32], one of the most used databases, and used in several scenarios. For instance, firewalls use its information to block IP addresses from foreign countries [38]. The database has good coverage. Considering all addresses submitted to JurisNN (39 595), only for 2 475 hops (6.3%) the database did not have Geo Location information. However, the database may have imprecise information. For instance, the ASN pair id 32 is related to a client in ES connecting to a server in IE, both in Europe. However, the identified network path traversed CA or the US in North America at some point, which may be possible but is unlikely to be true. Therefore, the results may suffer the influence of the imprecision in the Geo Localization process.

The network path between the client and the server was collected using the RIPE Atlas platform [5] conducting traceroutes from the client AS targeting the server IP a few months after the TD detection. This approach has some caveats. As the network path was collected months after the TD detection, the network has likely changed along with this time. Even the network conditions are different from those when the TD was detected, which may impact the routing. Another issue is that the traceroute uses the Internet Control Message Protocol (ICMP) protocol and may not identify all hops in the network path because the ISP routers can be configured not to respond to ICMP messages or rate-limit such messages. Another issue is that ICMP packets may be routed differently from application packets. Therefore, the identified network path may not reflect the path traversed by the application traffic. However, this may indicate that the ISP is performing TD (by routing application traffic differently) that some NN regulations prohibit. In order to overcome such issues, the TD detection and positioning solution could identify the network path using similar packets used to perform the TD detection (as performed by NeutMon [14] that crafts the TTL field of probe packets to identify routers along with the network path). Therefore, the network path could be collected simultaneously as the TD detection is performed. Thus, facing the same network conditions and being routed as the application packets were routed.

The RIPE Atlas platform offers methods to select the probe to perform the tests based on the network prefix, ASN, country, or globe area. In the initial phase of this analysis, it was tried to request probes from the same network prefix of the client of the TD. However, very few probes were selected using this criterion. Therefore, the criterion was shifted to select probes based on their ASN, which selected proper probes for 398 TDs (84.9%) from the 469 TDs. However, the use of the client ASN to select the probes may introduce issues because most of the ASs listed in Table 2 have a presence in multiple jurisdictions (*e.g.*, multiple states within the US). Indeed, several jurisdictions were identified as sources (Table 3). This issue could be overcome if the TD detection solution performed the network path identification.

6. Conclusion

A novel approach is introduced to signal NN violations considering the multiple NN definitions found in an end-to-end network path. The Regulation Assessment step is added after state-of-the-art solutions perform the TD detection and positioning steps. In addition, given the difficulty that such assessment be performed by every solution devoted to detecting TDs and NN violations, a service is proposed that is responsible for the management of information about NN definitions stated on regulatory instructions around the world, the processing of TD information to compare it to these definitions, and the signaling of the NN violation when the TD violates these definitions. Information models were designed to represent the information required to perform the regulation assessment. These models represent information about the regulation, the Traffic Differentiations, and the network topology that support the affected communication. The research identified that the jurisdiction of a case (which impacts what regulatory instruction should be evaluated) could be established by different methodologies in distinct legal systems. Judgment Algorithms can be developed to accommodate such nuances. In the end, these algorithms are responsible for inspecting the TD information submitted by TD detection solutions and for analyzing the appropriated regulatory instructions to signal the NN violation when the submitted TD violates the regulation.

The JurisNN prototype was developed to perform the regulation assessment step using TDs collected by Wehe and to analyze the conclusiveness of the results achieved with the available information. The prototype performs three Judgment Algorithms that represent the jurisdiction establishment methodologies identified during the investigation: Place of TD deployment, Targeting test, and German test. For the Targeting test, adopted by US courts, the provided information was enough to signal NN violations using the information provided. This test establishes the jurisdiction based on who was targeted by the TD that may be users and application providers. Thus, the jurisdiction is established by the endpoints that are easily positioned by the source/destination addresses of the communication.

In turn, the Place of TD deployment and German test were affected by the lack of information. Both Judgment Algorithms depend on the place in the network path where the TD was introduced to establish the jurisdiction (for consequence, the proper regulatory instructions to be evaluated). The Wehe dataset does not provide such information. The conducted analysis complemented the dataset collecting network paths between the source AS of the client and the destination IP of the server of each TD. In this analysis, the whole network path was considered as the occurrence zone, given the lack of information on where the TD was deployed. For the Place of TD deployment algorithm, the analysis found that for 18 of 24 identified scenarios (source and destination AS pairs), the results were inconclusive (similarity index of the verdicts within the occurrence zone is below 1.0). For the German test, only one client/destination AS pair was admitted to being

judged by it (only one traversed DE). The similarity indexes of the verdicts within the occurrence zones range from 0.62 to 0.77, which are also inconclusive. The results of both Judgment Algorithms point to the need for better information about where in the network path the TD is being deployed, which already had been identified by previous impact analysis [1] but based on less information. It is noteworthy that the network path information had to be collected to complement the information provided by Wehe to allow the conducted analysis, thus also indicating the lack of enough information.

The investigations conducted and the results presented in this article support that the state-of-the-art solutions do not collect and provide the information required to signal NN violations based on regulatory instructions properly. However, available proposals could be incorporated by solutions to collect such information. The end-to-end network paths could be collected using the approach of NeutMon [14]. The positioning of the TD could be achieved using Garrett et al.'s approach [17]. This article also showed the Regulation Assessment step's importance in providing reliable information to support users' claims. Therefore, it is expected that given these findings, the authors of TD detection and positioning solutions be encouraged to incorporate the collection of the required information in their solutions. It is also expected that solutions could be adapted to submit the collected TD information for the Regulation Assessment service.

Based on the studies conducted, it is possible to identify future work opportunities. For instance, this article focused on the relationship between TD detection and positioning solutions and the Regulation Assessment service. However, regulatory instructions allow certain TDs to be deployed under certain situational network conditions (*e.g.*, congestion) but require that the ISPs publicize such information for transparency. Therefore, the service could be complemented with transparency information that Judgment Algorithms could use. The development of other Judgment Algorithms is also an open issue. The developed algorithms just considered information about the nodes in the network topology because none solution provided complex network topology information where the link information could be considered. For instance, it could have a Judgment Algorithm that considers the weights of each network path on the similarity index calculation when the traffic traverses multiple paths.

CRediT authorship contribution statement

Márcio Barbosa de Carvalho: Conceptualization, Methodology, Software, Writing – original draft. **Lisandro Zambenedetti Granville:** Conceptualization, Methodology, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Funding information

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

References

[1] M.B. Carvalho, V.A. Cunha, E. da Silva, D. Corujo, J.P. Barraca, R.L. Aguiar, L.Z. Granville, Quantifying the influence of regulatory instructions over the detection of network neutrality violations, in: 2020 IFIP Networking Conference (Networking), 2020, pp. 334–342.

[2] J. Bustos-Jiménez, C. Fuenzalida, All packets are equal, but some are more equal than others, in: Proceedings of the 8th Latin American Networking Conference, LANC 2014, in: LANC '14, ACM, New York, NY, USA, 2014, pp. 5:1—-5:8, http://dx.doi.org/10.1145/2684083.2684088.

[3] M.B. Carvalho, V.G. Schaurich, L.Z. Granville, Considering jurisdiction when assessing end-to-end network neutrality, IEEE Internet Comput. (2018) http://dx.doi.org/10.1109/MIC.2018.2877836.

[4] F. Li, A.A. Niaki, D. Choffnes, P. Gill, A. Mislove, A large-scale analysis of deployed traffic differentiation practices, in: SIGCOMM 2019 - Proceedings of the 2019 Conference of the ACM Special Interest Group on Data Communication, 2019, pp. 130—144, http://dx.doi.org/10.1145/3341302.3342092.

[5] Réseaux IP Européens, RIPE atlas, 2021, Acessed on July 2021. URL https://atlas.ripe.net/.

[6] V. Mayer-Schönberger, The shape of governance: Anaylzing the world of Internet regulation, Va. J. Int. Law 43 (January) (2002) 605–674.

[7] C.I. Keller, Between exception and harmonization: The theoretical debate on internet regulation, Publicum 5 (1) (2019) 137–166, http://dx.doi.org/10.12957/publicum.2019.44573.

[8] Federal Communications Commission, FCC-17-166: Restoring internet freedom, 2018, Accessed on July 2021. URL https://docs.fcc.gov/public/attachments/FCC-17-166A1.pdf.

[9] B.E. May, J.C.V. Chen, K.W. Wen, The differences of regulatory models and internet regulation in the european union and the united states, Inf. Commun. Technol. Law 13 (3) (2004) 259–272, http://dx.doi.org/10.1080/1360083042000289077.

[10] M. Chertoff, P. Rosenzweig, A primer on globally harmonizing internet jurisdiction and regulations, in: Global Commission on Internet Governance, Paper Series No. 10 (10), 2015, pp. 1–16.

[11] F.F. Wang, Obstacles and solutions to internet jurisdiction a comparative analysis of the EU and US laws, J. Int. Commer. Law Technol. 3 (4) (2008) 233–241.

[12] W.G. Jiménez, A.R. Lodder, Analyzing approaches to internet jurisdiction based on a model of harbors and the high seas, Int. Rev. Law Comput. Technol. 29 (2–3) (2015) 266–282, http://dx.doi.org/10.1080/13600869.2015.1019204.

[13] V.G. Schaurich, M.B. Carvalho, L.Z. Granville, ISPANN: A policy-based ISP auditor for network neutrality violation detection, in: The 32-nd IEEE International Conference on Advanced Information Networking and Applications (AINA-2018), Pedagogical University of Kracow, Poland, 2018, pp. 1081–1088.

[14] E. Gregori, V. Luconi, A. Vecchio, NeutMon: Studying neutrality in European mobile networks, in: INFOCOM 2018 - IEEE Conference on Computer Communications Workshops, 2018, pp. 523–528, http://dx.doi.org/10.1109/INFCOMW.2018.8407022.

[15] Z. Zhang, O. Mara, K. Argyraki, Network neutrality inference, in: SIGCOMM 2014 - Proceedings of the 2014 ACM Conference on Special Interest Group on Data Communication, 2014, pp. 63–74.

[16] G. Hadley, M. Andenæs, D. Fairgrieve, Judicial Review in International Perspective, in: Judicial Review in International Perspective, vol. 2, Springer Netherlands, 2000.

[17] T. Garrett, L.C.E. Bona, E.P. Duarte, A holistic approach for locating traffic differentiation in the internet, Comput. Netw. 200 (2021) 108489, http://dx.doi.org/10.1016/j.comnet.2021.108489.

[18] H. Baumann, P. Grässle, P. Baumann, UML 2.0 in Action: A Project-Based Tutorial, in: From Technologies to Solutions, PACKT, 2005.

[19] C. Padovani, M. Santaniello, Digital constitutionalism: Fundamental rights and power limitation in the Internet eco-system, Int. Commun. Gaz. 80 (4) (2018) 295–301, http://dx.doi.org/10.1177/1748048518757114.

[20] G. Ivchenko, S. Honov, On the jaccard similarity test, J. Math. Sci. 88 (6) (1998) 789–794.

[21] M. Bjorklund, The YANG 1.1 Data Modeling Language, RFC 7950, in: Internet Request for Comments, RFC Editor, Fremont, CA, USA, 2016, http://dx.doi.org/10.17487/RFC7950, Acessed on July 2021. URL https://www.rfc-editor.org/rfc/rfc7950.txt.

[22] A. Clemm, J. Medved, R. Varga, N. Bahadur, H. Ananthakrishnan, X. Liu, A YANG Data Model for Network Topologies, RFC 8345, in: Internet Request for Comments, RFC Editor, Fremont, CA, USA, 2018, http://dx.doi.org/10.17487/RFC8345, Acessed on July 2021. URL https://www.rfc-editor.org/rfc/rfc8345.txt.

[23] A. Clemm, J. Medved, R. Varga, X. Liu, H. Ananthakrishnan, N. Bahadur, A YANG Data Model for Layer 3 Topologies, RFC 8346, in: Internet Request for Comments, RFC Editor, Fremont, CA, USA, 2018, http://dx.doi.org/10.17487/RFC8346, Acessed on July 2021. URL https://www.rfc-editor.org/rfc/rfc8346.txt.

[24] CZ.NIC, Jetconf, 2021, Acessed on July 2021. URL https://jetconf.readthedocs.io/en/latest/.

[25] B. Claise, J. Clarke, J. Lindblad, Network Programmability with YANG, Addison Wesley, 2019.

[26] M. Jaworski, T. Ziadé, Expert Python Programming - Fourth Edition: Master Python by Learning the Best Coding Practices and Advanced Programming Concepts, Packt Publishing, 2021.

[27] Measurement Lab, The M-lab Wehe data set (2021-01-01 – 2021-02-28), 2021, Acessed on July 2021. URL https://measurementlab.net/tests/wehe.

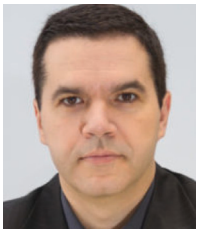
[28] Measurement Lab, Measurement lab, 2021, Acessed on July 2021. URL https://www.measurementlab.net/.

[29] D. Choffnes, Wehe website, 2021, Acessed on July 2021. URL https://wehe.meddle.mobi/.

[30] Réseaux IP Européens, RIPE atlas cousteau, 2021, Acessed on July 2021. URL https://ripe-atlas-cousteau.readthedocs.io/en/latest/.

[31] Economics of Cybersecurity Research Group, Delft University of Technology, PyASN, 2021, Acessed on July 2021. URL https://pypi.org/project/pyasn/.
[32] Maxmind, GeoLite2 databases, 2021, Acessed on July 2021. URL https://dev.maxmind.com/geoip/geoip2/geolite2/.
[33] CRTC - Canadian Radio-Television and Telecommunications Commission, Telecom Regulatory Policy CRTC 2009-657: review of the internet traffic management practices of internet service providers, 2009, Acessed on July 2021. URL https://crtc.gc.ca/eng/archive/2009/2009-657.htm.
[34] National Conference of State Legislatures, Net neutrality 2021 legislation, 2021, Acessed on July 2021. URL https://www.ncsl.org/research/telecommunications-and-information-technology/net-neutrality-2021-legislation.aspx.
[35] Federal Communications Commission, FCC-15-24: Protecting and promoting the open internet, 2015, Accessed on July 2021. URL https://docs.fcc.gov/public/attachments/FCC-15-24A1.pdf.
[36] Body of European Regulators for Electronic Communications, BoR (16) 127 - BEREC guidelines on the implementation by national regulators of European net neutrality rules, 2016.
[37] Ofcom, Monitoring compliance with the net neutrality rules, 2020, Acessed on July 2021. URL https://www.ofcom.org.uk/research-and-data/internet-and-on-demand-research/net-neutrality.
[38] F5, K22162026: Geolocation network firewall rule with Europe (EU) or Asia Pacific (AP) address/region not matching as expected, 2020, Acessed on July 2021. URL https://support.f5.com/csp/article/K22162026.

Márcio Barbosa de Carvalho received his B.Sc. (2010) and M.Sc. (2015) degrees in computer science from Federal University of Rio Grande do Sul (UFRGS). He is currently a Ph.D. student in the Institute of Informatics of UFRGS. His current research interests include network neutrality and network softwarization.



Lisandro Zambenedetti Granville received the M.Sc. and Ph.D. degrees in computer science from the Federal University of Rio Grande do Sul (UFRGS), Brazil, in 1998 and 2001, respectively. He is currently a Full Professor with the Institute of Informatics, UFRGS. He has served as a TPC member for many events in the area of computer networks, such as IM, NOMS, and CNSM. He has served as the TPC Co-Chair of DSOM 2007 and NOMS 2010.