HAL open science

Analyzing the Wehe Network Neutrality Monitoring Tool

Ximun Castoreo, Patrick Maillé, Bruno Tuffin

To cite this version:

Ximun Castoreo, Patrick Maillé, Bruno Tuffin. Analyzing the Wehe Network Neutrality Monitoring Tool. GECON 2021 - 18th International Conference on the Economics of Grids, Clouds, Systems, and Services, Sep 2021, Online, France. pp.155-167, 10.1007/978-3-030-92916-9_13 . hal-03177366

HAL Id: hal-03177366

https://inria.hal.science/hal-03177366v1

Submitted on 23 Mar 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Analyzing the Wehe Network Neutrality Monitoring Tool

Ximun Castoreo
Inria, Univ Rennes, CNRS, IRISA
Rennes, France
ximun.castoreo@inria.fr

Patrick Maillé
IMT Atlantique, IRISA, UMR CNRS 6074
F-35700 Rennes, France
patrick.maille@imt.fr

Bruno Tuffin
Inria, Univ Rennes, CNRS, IRISA
Rennes, France
bruno.tuffin@inria.fr

***Abstract*—Network Neutrality is protected by law in many countries over the world, but monitoring has to be performed to ensure operators conform to the rules. The Wehe application, jointly developed by Northeastern University and the French regulator ARCEP, allows users to take measurements and analyze them to detect possible traffic differentiation. This paper presents a test bed designed to evaluate the detection capacities of Wehe; by computing the detection probabilities and estimating the potential benefit of an operator willing to differentiate while avoiding detection, we fine-tune and compare the main differentiation types (throughput, packet loss and delay) that an operator could implement.**

***Index Terms*—Network Neutrality, monitoring tool, ISP benefit**

I. Introduction

A. Network Neutrality

The Internet is used by a vast and heterogeneous group of users (individuals, companies, governments, associations, etc.) who communicate with each other through inter-connected networks owned by Internet Service Providers (ISPs). These providers own the network architecture and control the way they convey traffic.

The Network Neutrality [20] principle aims to ensure a fair network experience for every user. The pieces of legislation protecting Network Neutrality over the world mostly agree on the following interpretation of that principle: no traffic differentiation based on traffic origin, destination, protocol or service is accepted [6], [10], [15].

The first benefit of Network Neutrality is a wide and complete access to the different public resources of the network, no matter the user's specifics (geographical and cultural origin, working situation, political beliefs, etc). At the same time, innovation can thrive on the network without monopoly or unfair competition, as all online services are equally accessible. Network Neutrality also helps innovation in the networking domain, because it prevents putting forward certain protocols or applications. New protocols can be freely tested and adopted without compatibility issues.

Seen by opponents, Network Neutrality is a pure ISP limitation. Being unable to manage the traffics flowing through their network, ISPs cannot propose *differentiated* offers, apply revenue management for a better return on investment, or make deals with companies for preferential treatment. Moreover, they cannot ensure Quality of Service requirements from demanding types of traffic. Network Neutrality is also limited if protection laws apply in some countries while traffic may transit through other places applying differentiation, hence barring end-to-end equality of treatment [14].

B. Measurement tools

Even if Network Neutrality is enforced by law, ISPs do not always comply with it. Pointed violations [1], [5], [8] have shown that operators tend to differentiate traffic for commercial reasons. At the other extreme, blocking is sometimes asked by governments for security or political reasons [3], [21], or for legal reasons such as for example with peer-to-peer being accused of infringing copyright rules.

Hence there is a need for *tools* to monitor ISPs behavior: such tools are required for regulators, guarantors of the law, to ensure ISPs conform to the enacted rules, but also for end users to evaluate ISPs and possibly switch operator if the current one appears to violate Network Neutrality.

The research community and user associations have created various tools to check Network Neutrality (see [2], [9] for a full list). The existing tools differ in various ways: the checked violation, the measured metrics, the interaction they have with the network infrastructure, the measure type, the tool architecture, etc. For example, the POPI tool [13] makes passive measurements, and aggregate measures from different nodes into an inference analysis model to detect packet forwarding prioritisation. This highly differs from Switzerland [4], that uses active measures to check packet integrity between a client and a server.

But as mentioned in [2], the available tools are limited in number, in scope, and are rarely maintained. A recent tool probably standing out is Wehe [16], [19] stemming from a joint development between Northeastern University and the French regulator ARCEP. This tool has been highly advertised because of the participation of a regulatory body, and is maintained. We therefore choose to focus on it in this paper.

C. Goal: determining Wehe differentiation detection limits and potential resulting gains

The main result provided by Wehe is binary, indicating whether differentiation has been detected or not. In this paper, we aim at investigating the sensitivity of that detector, to analyze how reliable its results are, and whether it could

still be beneficial for an operator to perform some carefully-designed differentiation, if that differentiation can be monetized. For those reasons, a key step is to determine, through a test bed, how much differentiation can be introduced before being detected by Wehe, for different types of differentiation: throughput limitation, packet loss and packet delay. We are then able to present which differentiation means is the most beneficial for an ISP and if a significant gain can be derived from it.

The remainder of the paper is organized as follows. Section II briefly presents the Wehe tool and its main characteristics; Section III introduces the testing platform we have developed; the experimental results are given and analyzed in Section IV; Section V discusses the test bed generality and limitations; and finally Section VI concludes and suggests directions for future work.

II. Wehe: a differentiation detection tool

Wehe is a Network Neutrality monitoring tool aiming at studying differences in terms of throughput for a traffic sent both "regularly" and in a way that the operator cannot identify the flow (the tool assumes non-differentiation in that latter case). It has been presented in 2015 [16] as a joint venture between Northeastern University and the French communication regulator ARCEP. The application targets mobile devices because of known mobile network issues (wide group of users, resource scarcity, network opacity). The interest of Wehe resides in its genericity: it theoretically allows the user to test any traffic (classic traffic, user-customised traffic, encrypted traffic...) even if specific traffic types are targeted in the application to ease usage.

A. Wehe functioning principles

Wehe is based on active measures between a client and a Wehe server, and works as follows: the tool replays twice a prerecorded traffic between the client (an app installed by the user on their device) and the server (a specific server running the Wehe service). The first replay is identical to the original traffic while the second traffic's payload is modified (by randomizing or encrypting it). In both cases, the replayed traffic has the same shape as the original one: same packet sizes with same IP and TCP/UDP protocol headers (minus the IP addresses) and same inter-packet timings (see replay similarity in [16]), but with an unidentifiable payload in the latter case (through encryption, or just by replacing the application data with random bits).

Therefore, the modified replay traffic cannot be identified by the means of Deep Packet Inspection (DPI), and cannot be differentiated afterwards when assuming that an ISP does DPI-based differentiation (e.g., targeting a specific application like YouTube that is very bandwidth-consuming): only the unmodified replay would suffer differentiation. During replays, the client and the server measure the throughput of each traffic. Then, the throughput distributions are compared using a statistical test inspired by the Kolmogorov-Smirnov test [16]. If the test does not reject the assumption of throughput samples being from the same distribution, Wehe does not raise any warning about a potential non-neutral behavior. Otherwise, Wehe considers that a differentiation occurred on the original traffic and signals it to the user.

More details on how those replays are built and performed are given below.

B. Wehe replay

Wehe records an original traffic that has been conveyed through the network. It is separated in two traces: the client trace and the server trace. These two represent the packets each side has to send to simulate the original traffic. To keep the simulation accurate, the Wehe designers have added two constrains to packet transmission: a packet cannot be sent before the prior one was received (happen-before dependency), and it also waits the duration given in the original transmission (time dependency). This way, a replay's shape is identical to the original's shape.

The actual replays are initiated by the client application: it connects to the server, specifies the traffic it is going to replay and waits the server to be ready. Then they start transmitting their trace for each replay (original and randomized), respecting the dependencies. Wehe measures the throughput of the two replayed traffics. Each side of the replay periodically measures the sent and received data amount. When the replays finish, the client asks for analysis to the server.

C. A detection "grey zone"

The modified Kolmogorov-Smirnov test that is used is a heuristic one and involves a grey zone on which it cannot clearly make a decision. In that case Wehe runs another time the replays and re-analyzes them. After a few iterations of such few unsuccessful analysis, Wehe declares that no differentiation was found to limit the risk of false positives. Avoiding as much as possible a false positive makes the test conservative in order to restrict (legal) complaints from ISPs, an important component from regulators' point of view. But it also increases the possibilities for an operator to fool the tool, highlighting the relevance of the present work.

III. Building a Test bed to evaluate Wehe

To analyze the performance of the Wehe tool, we designed a simple test bed, with a controlled environment, to perform different kinds of ISP traffic differentiation and investigate whether Wehe detects them.

A. Test bed setup

The test bed's simple topology is composed of three parts: the client side, the server side and the core network part. The client and server sides are two devices where Wehe applications are installed. We use the proof-of-concept code available from https://github.com/NEU-SNS/wehe-server.

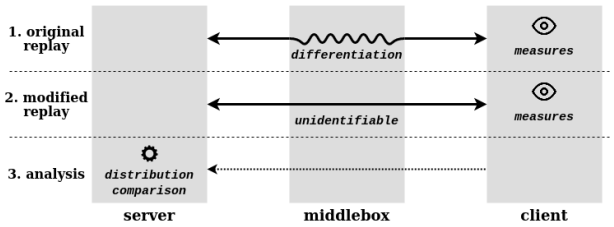
The core network part, meaning the existing ISP networks between a client and a Wehe server, is emulated by a single device running a Linux Traffic Control utility with a netem queueing discipline [7] for classifying and differentiating traffic. We call this device the test bed middlebox. Modeling the

whole network between source and destination with a unique device is common practice and sufficient since the Wehe tool only makes end-point measurements.

The tc-netem queueing discipline allows to control the throughput, the packet loss rate and the delay applied to classified packets. This way, we are able to choose between general but precise traffic deterioration (with throughput) or more random performance loss (using a packet loss rate, losses being then decided independently for each packet).

Figure 1 summarizes the test bed and the important parts of a Wehe test. The packet classifier is started on the middlebox. Then the two replays are run through the test bed. When the classifier identifies the unmodified replay (because its data correspond to the original traffic's data), it applies the differentiation. At the end of the Wehe run, the values of throughput calculated during the transmissions are sent back to the server for analysis.

Fig. 1. The three parts of the Test bed *(horizontally)* and the steps of a Wehe run *(vertically)*. What is "replayed" are packet exchanges recorded beforehand.



To differentiate a traffic, it must be identified beforehand. To do so, we use a keyword present in the targeted traffic. When the keyword is found by the middlebox, the differentiation is triggered for every packet of the corresponding flow (a flow is defined by the IP addresses and port numbers).

B. Traffic and differentiation in the experiments

Our experiments are carried out for the traffic corresponding to a **file transfer**, a basic but essential traffic type that for example corresponds to a web page request, and represents a significant part of the Internet traffic. That traffic is captured and saved beforehand, to be replayed during the tests. The file transfer is a simple web page retrieval of a random 1GB file (thus an HTTP GET request).

We implemented several types and levels of differentiation (described below). Repeated independent experiments allowed us to plot the detection probability in terms of the differentiation parameter value, together with a confidence interval. The differentiation can take three different forms, whether it affects the transmission throughput, the packet loss rate, or the packet delay. Those three types of differentiation are supported by tc-netem:

i) Throughput limitation (called traffic shaping) delays packets when the measured throughput of the transmission exceeds a certain value. If too many packets are delayed and the waiting queue fills up, the following packets are dropped.

ii) Packet loss rate differentiation applies an independent drop probability to each transmitted packet. Random packet losses can happen in a physical network, but we here simulate a deliberate loss applied by the network operator.

iii) Packet delay retains all the transmitted packets for a predefined amount of time. Delay can be observed when congestion hits the network, but in the same way as for packet losses, we emulate an intentional behavior that affects all the classified packets.

IV. Experimental results and analysis

In this section, we present and analyze the detection results from extensive experiments with Wehe on our test bed. First, for different types and intensities of differentiation, we estimate the detection probabilities. Then we use those results in a simple model to determine an optimal differentiation plan that an ISP could implement under Wehe monitoring. The designers of Wehe already tested and validated their tool [16], but our goal in this paper is different: we focus on the ISP point of view and the possibility to maximize the differentiation impact while avoiding Wehe detection.

A. Raw results: Wehe detection probabilities

Our analysis first consists in estimating the Wehe differentiation detection limits. The tool accuracy is indeed the key to further investigate how an ISP could still differentiate under Wehe monitoring. As the Wehe statistic decision model is based on the client-side calculated throughput that can be slightly different on each test, we run numerous tests for each setting and estimate the detection probability of the tool. These probabilities will then be used later to build a ISP differentiation benefit model.

To detect the parameter ranges where (non-)detection is not systematic, we first ran tests for a broad range of differentiation parameter values, and then we focused on shorter differentiation value intervals experiencing more variability in terms of detection. We present here the results on these shorter intervals for the three differentiation types. In each case, the results given are for a sample of size 150.

The graphs in Figure 2 respectively display the Wehe differentiation detection probabilities versus the traffic throughput reduction, packet loss rate, and delay, for a file transfer traffic. We also run our experiment for another traffic type, namely video streaming, for which results are given in Figure 3.

The figures illustrate the expected tendency that the more differentiation is applied, the larger the probability to be detected is.

B. Differentiating while monitored by Wehe

Given the detection probability measurements obtained in the previous subsection, we now focus on whether traffic differentiation can significantly impact traffic and therefore users, while being only rarely detected. Taking the ISP point of view, that would indicate what level(s) of differentiation can be implemented, and how worthy it would be. (Note that differentiation can be motivated by various reasons, such as to free network resources, or to slow down a specific traffic for commercial purposes).

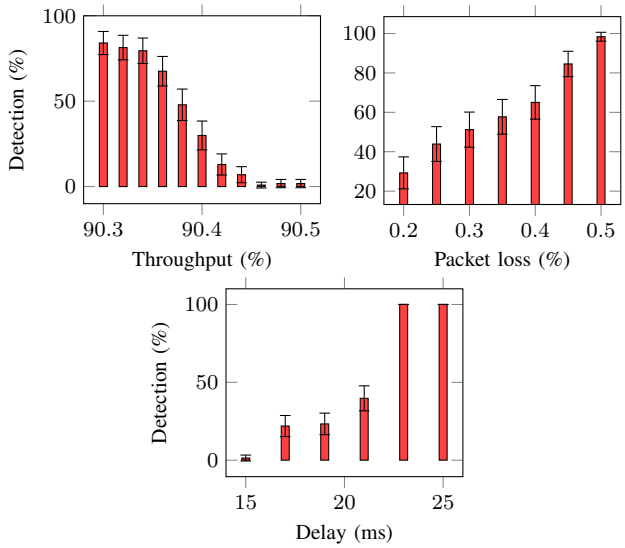


Fig. 2. Wehe detection probability estimations in the case of file transfer for three types of differentiation, with 95% confidence intervals

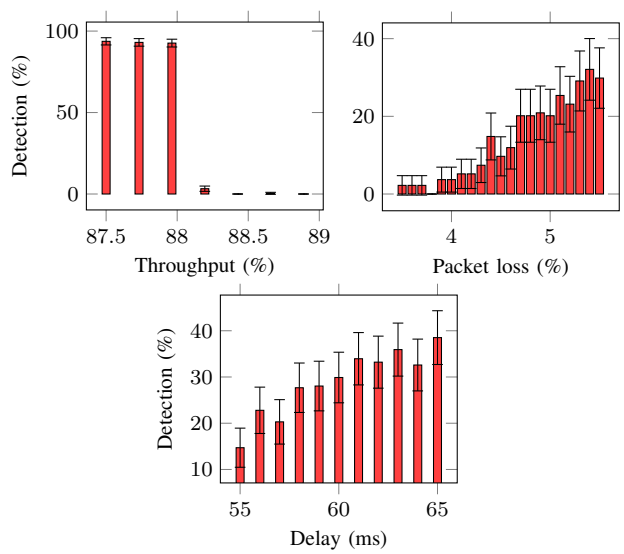


Fig. 3. Wehe detection probability estimations in the case of video streaming for three types of differentiation, with 95% confidence intervals

1) Detection threshold: A possible simple strategy for an ISP is to try to keep the detection probability below some threshold. For example, to deteriorate video streaming traffic with a detection probability no larger than 15%, from Figure 3 we deduce that the ISP can reduce throughput by no more than 12%, or apply up to 4.6% packet loss rate, or add less than 55ms of delay. For file transfer with the same 15% detection probability limit, Figure 2 shows that the ISP can reduce throughput by no more than 8.58%, or apply up to 0.15% packet loss rate, or add less than 16ms of delay.

But such a reasoning does not tell us which differentiation strategy has the largest impact.

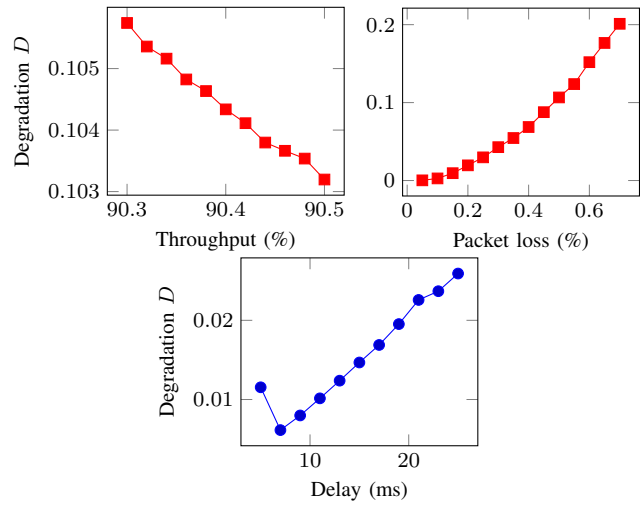


Fig. 4. Impact of the three types of differentiation on the degradation level D (relative increase of transfer time of a 1GB file).

2) Detection vs impact on user perceived quality: To further analyze the possible ISP differentiation benefits, one must study the impact that traffic differentiation has on users. That impact of course depend on the service used: file transfer and video streaming, for example, will not be equally sensitive to differentiation from a user point of view.

In the rest of this section, we focus on file transfers, for which an appropriate and simple quality metric can be provided: the total transfer time. More specifically, we will consider as the degradation metric the **relative transfer time increase**, which we will denote by D, when differentiating traffic: if differentiation leads to a total expected transfer time T_d instead of T_n, then our degradation metric D is

$$D := \frac{T_d - T_n}{T_n}. \tag{1}$$

In our experiments, that degradation is estimated for the transfer of a 1GB file. Figure 4 shows the impact of the three types of differentiation on the transfer time ratio, in the parameter intervals that were previously identified as "interesting" (with low but non-zero detection probabilities).

Since the trade-off faced by an ISP willing to monetize differentiation would be between the degradation and the detection probability, we display those two values on a common graph for all types of differentiation, combining the results from Figures 2 and 4, in Figure 5.

The figure highlights the differences between the three types of differentiation: directly degrading the throughput allows an ISP to extend the transfer time by nearly 10% without being detected by Wehe, while by affecting packet losses or delay, the detection probability is significant before reaching such an impact on the transfer time. Among the three types, playing on delay appears to be the least effective, as the detection probability increases very fast with the degradation: with only about 2% degradation the differentiation is detected. Playing with packet losses leads to a smoother curve, but again, with

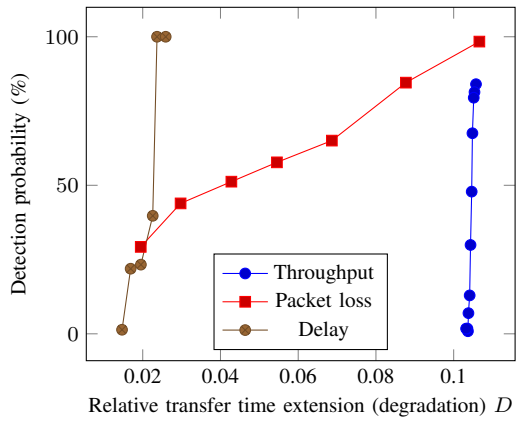


Fig. 5. Detection probability versus differentiation impact (relative transfer time extension) for each type of differentiation, varying its intensity

only 2% degradation the detection probability already exceeds 25%.

3) Optimizing the differentiation: To deal with the impact-detection trade-off faced by an operator, we build a utility model that incorporates, under the form of a single objective function, the positive impact of differentiation (assuming the operator can monetize that differentiation) and the negative impact of being detected by Wehe.

We assume the impact of differentiation on the perceived user quality of service can be monetized, for example by having some content providers pay to avoid it or to penalize their competitors. To quantify that monetization, we consider the simplest model possible, with a constant marginal value g for degradation, i.e., the differentiation can yield the operator some gain gD, with D the degradation level (given in (1) for the case of file transfer).

On the other hand, being detected is bad for the operator, at this may come with a fine to pay, a loss of reputation, or even possibly an interdiction to further operate. To represent this variety of interpretations, we consider a cost function that will depend on the probability to be detected, which we will denote by P_d, and such that:

- for low values of P_d, the cost is (approximately) proportional to P_d, and can be interpreted as the operator being fined when detected;
- with P_d increasing, the regulator is more and more likely to take more severe measures, whose cost for the operator would tend to infinity as P_d tends to one.

A simple function satisfying those conditions is $P_d \mapsto s\frac{P_d}{1-P_d}$, with a sanction parameter s interpreted as the amount of the fine when detected (for small values of P_d).

Summarizing, we will consider that when implementing some differentiation, denoted abstractly by δ, which leads to a degradation $D(\delta)$ and is detected with probability $P_d(\delta)$, the operator perceives a net expected benefit (or utility) $U(\delta)$, equal to

$$U(\delta) = gD(\delta) - s\frac{P_d(\delta)}{1 - P_d(\delta)}. \tag{2}$$

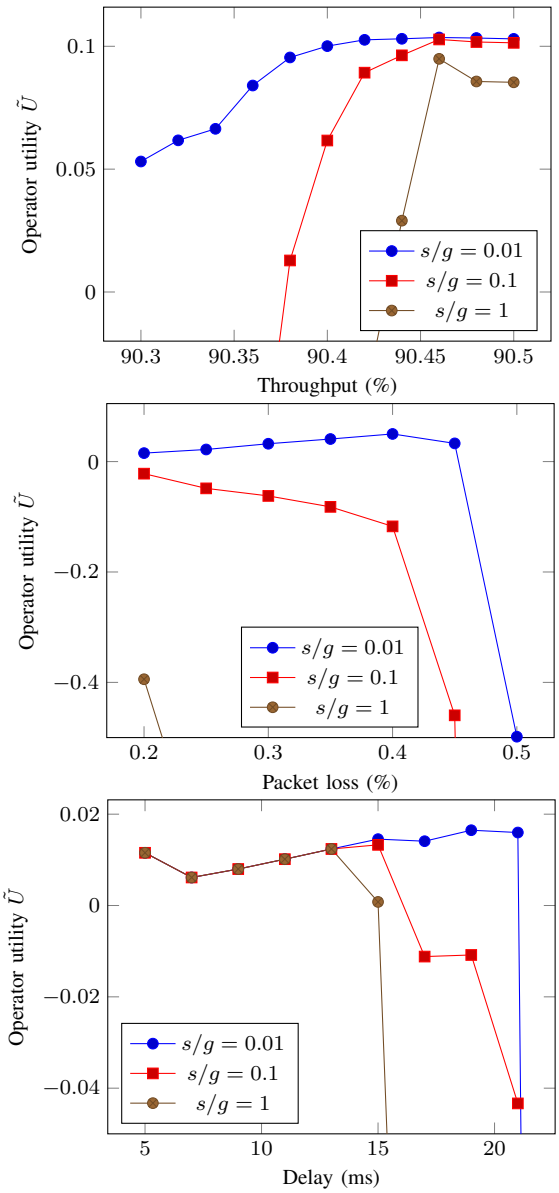


Fig. 6. ISP utility function $\tilde{U}$ when differentiating for file transfers

Note that finding a utility-maximizing differentiation δ to implement depends only on the ratio s/g, so we will focus on the quantity

$$\tilde{U}(\delta) = D(\delta) - \frac{s}{g}\frac{P_d(\delta)}{1 - P_d(\delta)}. \tag{3}$$

Using our detection and degradation measures in the case of file transfers, we plot in Figure 6 the values of $\tilde{U}$ for each type and intensity of differentiation, with different values of the ratio $\frac{s}{g}$.

This illustrates how a utility-maximizing ISP may reason to manage the differentiation/detection trade-off, once the ratio s/g is known:

- First, for a given differentiation type, the formulation (3) can be used to find the optimal differentiation level. For

example, if $s/g = 0.01$, then

- when playing on throughput the optimal reduction is around 90.45%;
- if differentiation is through packet losses, the optimal loss rate to introduce is 0.4%
- if instead differentiation means delaying packets, the optimal delay to add is 19ms.

- Second, once each differentiation type is optimized, the analysis helps to compare them decide which one maximizes the overall utility $\tilde{U}$ in (3). Here, again for $s/g = 0.01$, playing on throughput can yield a value of $\tilde{U}$ above 0.1, while with packet losses or delay $\tilde{U}$ remains below 0.06 and 0.017, respectively. Hence for the specific case of file transfers, it seems that affecting the throughput is the most effective.

V. Test bed precision, limitations and improvement axes

Even if the paper focuses on Wehe, this section comes back to our test bed design and discuses its use to analyze other network neutrality tools, presenting some points that can be limitations, and suggesting directions for improvements.

A. One-way differentiation

In its current state, the test bed applies differentiation on either the server-to-client or the client-to-server traffic.

As most traffic types are highly asymmetric (like the file transfer we focused on), there is no big issue in only applying the differentiation to packets that are sent by the server to the client.

Also, ISP differentiation is often due to a bottleneck situation in one of the two directions of a network path. Thus, a one-way differentiation model fits the reality.

B. Tool architecture support

The created test bed was designed to analyze the Wehe tool. But it can be used as a basis to possibly test other tools, as those mentioned in Section I-B.

Nevertheless, it may be hard to run tests for those tools on our current test bed, because of the already evoked problems with them, but also because our design focuses on a simple server/client communication architecture. This is not sufficient to evaluate tools such as NANO [18] that runs with multiple clients. To deal with such a tool, the test bed would need to be adapted to allow multiple clients. In the same logic, our test bed could not run a tool similar to Switzerland [4] that involves client-to-client communications.

Other tools such as NetPolice [22] and ChkDiff [17] do not focus on end-to-end differentiation detection as Wehe does. They instead infer intra-network behaviors (the first one deals with routing paths and the second targets the specific differentiating device along the path) that require a full core network simulation. It would complicate the test bed to actually have to choose a valid network simulation (a lot of parameters must be taken into account and ISP behavior cannot be accurately simulated because of the lack of information concerning the core network practices).

Another limitation of the presented test bed is that it only supports "classical" traffic differentiation, i.e., affecting throughput, packet losses or delay along the path. It does not support nor does it focus on other Network Neutrality violation types such as censorship and DNS misuses. The tool OONIProbe [11] that studies such behaviors could therefore not be tested on the test bed: it would require an extra step of simulation to setup a DNS infrastructure and implement the different suspected violations.

VI. Conclusions

In this paper, we have analyzed the traffic differentiation detection tool Wehe, that is recommended by some regulators to detect net neutrality violations. To do so, we have designed a test bed that allows us to run Wehe in a controlled environment, where three types of differentiation are implemented (transmission throughput, packet loss rate and packet delay). For each differentiation type, we have carried out intensive simulations of the detection tool, to estimate the Wehe detection probabilities and indicate thresholds over which differentiation is significantly pointed out.

For the case of file transfers, we have quantified the impact that the differentiation types have on the total transfer time, a natural metric users are sensitive to. This has enabled us to build a model, assuming operators can monetize that differentiation, where an operator weighs that possible gain with the risk associated to detection. A utility function taking into account those two aspects can be used to manage the trade-off, determining the optimal type and level of differentiation to implement. Such a reasoning can for example help regulators set the sensitivity of their monitoring tools.

This paper opens several directions for future work. First, while our study mainly focuses on file transfer as an application. We used it because there is an immediate user-oriented performance metric to apply, that is the transfer time. But we intend to also carry out a similar analysis for other types of traffic, in particular for video streaming. For that latter type of traffic, the user-perceived quality depends on the protocoles used and is less direct to evaluate: researchers usually try to estimate the Mean Opinion Score that users would give to the quality [12]. Also, despite the difficulties raised in the previous section, we would like to compare the performance of Wehe to that of other differentiation-detecting tools.

References

[1] AirFrance: AirFrance Connect WiFi on Board. https://www.airfrance.fr/FR/en/common/transverse/footer/wifi-a-bord.htm, last accessed 23 Oct 2020.

[2] Castoreo, X., Maillé, P., Tuffin, B.: Weaknesses and Challenges of Network Neutrality Measurement Tools. In: Proc. of 16th IEEE International Conference on Network and Service Management (CNSM). Virtual Conference (Apr 2020), https://hal.inria.fr/hal-02542689

[3] Clayton, R., Murdoch, S.J., Watson, R.N.: Ignoring the great firewall of China. In: Proc. of International Workshop on Privacy Enhancing Technologies. pp. 20–35. Springer, Cambridge, UK (2006)

[4] Eckersley, P.: Switzerland Design. - (May 2008), https://www.eff.org/files/2018/06/21/design.pdf, last accessed 23 Oct 2020.

[5] Electronic Frontier Foundation: Packet Forgery by ISPs: A Report on the Comcast Affair. https://www.eff.org/wp/packet-forgery-isps-report-comcast-affair, last accessed 23 Oct 2020.
[6] European Parliament & Council: Regulation (EU) 2015/2120 of the European Parliament and of the Council. https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A32015R2120 (November 2015), last accessed 23 Oct 2020.
[7] The Linux Foundation: netem. https://wiki.linuxfoundation.org/networking/netem, last accessed 23 Oct 2020.
[8] Gannes, L.: AT&T Changes TOS to Limit Mobile Video. https://gigaom.com/2009/04/02/att-changes-tos-to-limit-mobile-video/, last accessed 23 Oct 2020.
[9] Garrett, T., Setenareski, L.E., Peres, L.M., Bona, L.C.E., Duarte, E.P.: Monitoring network neutrality: A survey on traffic differentiation detection. IEEE Communications Surveys Tutorials **20**(3), 2486–2517 (2018). https://doi.org/10.1109/COMST.2018.2812641
[10] Indian Department of Telecommunications: National Digital Communications Policy - 2018. https://dot.gov.in/sites/default/files/Final%20NDCP-2018_0.pdf (2018), last accessed 23 Oct 2020.
[11] Open Observatory of Network Interference: OONIProbe. https://ooni.org/, last accessed 2 Mar 2021.
[12] Khokhar, M., Ehlinger, T., Barakat, C.: From network traffic measurements to QoE for internet video. In: Proc. of IFIP Networking. Warsaw, Poland (2019)
[13] Lu, G., Chen, Y., Birrer, S., Bustamante, F.E., Li, X.: Popi: A user-level tool for inferring router packet forwarding priority. IEEE/ACM Trans. Netw. **18**(1), 1–14 (Feb 2010). https://doi.org/10.1109/TNET.2009.2020799, https://doi.org/10.1109/TNET.2009.2020799
[14] Maillé, P., Tuffin, B.: Neutral and Non-Neutral Countries in a Global Internet: What Does it Imply? In: Springer (ed.) GECON 2019 - 16th International Conference on the Economics of Grids, Clouds, Systems, and Services. pp. 1–14. Leeds, United Kingdom (Sep 2019)
[15] Ministerio de Transportes y Telecomunicaciones de Chile; Subsecretaría de Telecomunicaciones: Ley 20.453: consagra el principio de neutralidad en la red para los consumidores y usuarios de Internet. https://www.leychile.cl/Navegar?idNorma=1016570 (August 2010), last accessed 23 Oct 2020.
[16] Molavi Kakhki, A., Razaghpanah, A., Li, A., Koo, H., Golani, R., Choffnes, D., Gill, P., Mislove, A.: Identifying traffic differentiation in mobile networks. In: Proceedings of the 2015 Internet Measurement Conference. p. 239–251. IMC '15, Association for Computing Machinery, New York, NY, USA (2015). https://doi.org/10.1145/2815675.2815691, https://doi.org/10.1145/2815675.2815691
[17] Ravaioli, R., Urvoy-Keller, G., Barakat, C.: Towards a General Solution for Detecting Traffic Differentiation At the Internet Access. In: 27th International Teletraffic Congress (ITC-27). Ghent, Belgium (Sep 2015). https://doi.org/10.1109/ITC.2015.8, https://hal.inria.fr/hal-01161795
[18] Tariq, M.B., Motiwala, M., Feamster, N., Ammar, M.: Detecting network neutrality violations with causal inference. In: Proceedings of the 5th International Conference on Emerging Networking Experiments and Technologies. p. 289–300. CoNEXT '09, Association for Computing Machinery, New York, NY, USA (2009). https://doi.org/10.1145/1658939.1658972, https://doi.org/10.1145/1658939.1658972
[19] Northeastern University: wehe-server. https://github.com/NEU-SNS/wehe-server, last accessed 23 Oct 2020.
[20] Wu, T.: Network Neutrality FAQ. http://www.timwu.org/network_neutrality.html, last accessed 23 Oct 2020.
[21] Xynou, M., Filastò, A.: Togo: Instant messaging apps blocked amid 2020 presidential election. https://ooni.org/post/2020-togo-blocks-instant-messaging-apps/, last accessed 23 Oct 2020.
[22] Zhang, Y., Mao, Z.M., Zhang, M.: Detecting traffic differentiation in backbone isps with netpolice. In: Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement. p. 103–115. IMC '09, Association for Computing Machinery, New York, NY, USA (2009). https://doi.org/10.1145/1644893.1644905, https://doi.org/10.1145/1644893.1644905