# On Bringing Private Traffic into Public SDN Testbeds

Vasileios Kotronis
ETH Zurich
vkotroni@tik.ee.ethz.ch

Dominik Schatzmann
ETH Zurich
schadomi@tik.ee.ethz.ch

Bernhard Ager
ETH Zurich
bager@tik.ee.ethz.ch

## ABSTRACT

The search for improved communication paradigms has fostered the emergence of publicly available testbeds supporting Software Defined Networking all around the world. However, a common shortcoming among these testbeds is the lack of real user-driven Internet traffic for experimentation. While having real user traffic inside a testbed is an indisputable advantage, the users' right for privacy and wish for availability of the network often make it impossible to simply make a testbed part of the communication path.

In this paper, we discuss how a testbed operator can give privacy and availability guarantees to users who are willing to share part of their traffic with experimenters, thus making it less risky for users to opt-in to experiments. Moreover, we share the insights gained from implementing a Privacy and Availability Layer demonstrator.

## Categories and Subject Descriptors

C.2.1 [**Network Architecture and Design**]: Experimentation

## Keywords

SDN, Testbed

## 1. INTRODUCTION

The combined efforts on exploring the possibilities of Software Defined Networking (SDN) for new network applications have led to publicly accessible SDN testbeds all around the world. Examples include OFELIA [6], GENI, FIBRE, and JGN-X. However, these testbeds often run as stand-alone islands and have only limited possibilities to exchange traffic with the Internet for safety and privacy reasons. Safety considerations come into play when thinking about the damage, e.g., network outages, that can result from an experiment going bad. Moreover, an experimenter in an SDN testbed has vast control over the traffic, including the possibility to intercept, manipulate, and redirect communication. This immense power of the experimenter raises immediate privacy and availability concerns when thinking about having user traffic inside an SDN testbed. Nonetheless, experimenters would like to test out their inventions with user traffic for diverse reasons, e.g., investigating how a system performs under a real-world work load.

In this paper, we look for a compromise between the users' concerns and the experimenters' needs. Ideally, an experimenter should be able to describe the type of traffic he needs for his experiment, and a user should be able to specify which parts of his traffic he is willing to make available under certain constraints. Such constraints could include keeping the traffic payload private, anonymizing endpoint addresses, or not passing some traffic through the testbed at all. Moreover, network availability should be guaranteed to the user whenever possible. This still leaves the question why users would be willing to share part of their traffic. However, building such a meeting point for experimenters and users enables a marketplace, where in addition to voluntarily donated traffic, experimenters can offer advantageous network features to users or even pay users to get access to the interesting parts of their traffic.

To enable this marketplace we propose to include a *Privacy and Availability Layer (PAL)* in today's testbeds, consisting of a proxy that is placed between the experimenters' controllers and the network elements observing all control plane traffic, and a set of gateways to control the injection of user traffic into the testbed. We show how these elements can be used to protect the user from possible privacy and availability policy violations. For the sake of concreteness and brevity we limit ourselves to OpenFlow [3] version 1.0 [4]. Yet, our methodology is adaptable to newer OpenFlow versions and, at least partially, to future SDN systems.

The problem of providing privacy and availability guarantees on an SDN based network is not limited to testbeds, and we believe that this work is the first step towards a more general policing layer that should be integrated, as best practice, in any SDN network. We point out that state-of-the art network slicing and policing solutions [1, 2, 5] *are not sufficient* for our use case.

## 2. PROBLEM SPACE

Before delving further into our proposal we need to understand the underlying problems when bringing user traffic into a publicly available testbed. Therefore, we first study the capabilities of an attacker based on typical testbed setups and discuss the resulting privacy and availability threats for the user.

The goal of an SDN testbed is to enable researchers to perform SDN experiments. Typically, SDN testbeds consist of SDN capable networking elements and physical hosts, forming the substrate for building multiple virtual networks called *slices*. The separation of the individual slices is ensured by a networking hypervisor such as FlowVisor [5]. The experimenters can then generate and forward traffic through the testbed to implement their experiments.

We propose to bring user traffic into the testbed by effectively putting the testbed on the path between the user and the Internet


*HotSDN'13,* August 16, 2013, Hong Kong, China.
ACM 978-1-4503-2178-5/13/08.

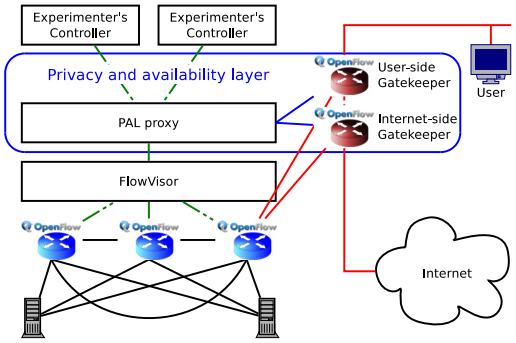


**Figure 1: Integrating the privacy and availability layer.**

(see Figure 1). By using SDN switches as *gatekeepers,* and provided we have a user's consent, we can precisely control which part of the traffic is traversing which slice inside the SDN testbed.

The attacker in this scenario is an experimenter desiring to disrupt communication, to access or even to modify private user traffic. The attacker is quite powerful, as the network path inside the testbed is chosen by him. Moreover, the attacker can use a testbed host to capture traffic or use network address translation to redirect traffic to an Internet host under his control. Consequently, we have to deal with two types of threats: *(i)* privacy threats in which the attacker can learn about the user's communication, e.g., via flow statistics, by exploiting the OpenFlow control channel, or by redirecting traffic to a host under his control, and *(ii)* availability threats, i.e., interruption of communication.

## 3. CREATING A MARKETPLACE

In this section we discuss how to bring user traffic to a testbed. Note, there are three parties involved: the user, the experimenter, and the testbed operator mediating between the former two.

A user willing to share part of his traffic in principle might be hesitant to do so because of the threat of privacy violation or network outages. In order to overcome the user's concerns, a testbed operator can offer privacy and availability guarantees. A user should be able to freely choose how he wants his traffic to be treated. To that end, he needs to specify three things for each part of the traffic he is willing to donate: *(i)* which part of the traffic is affected, *(ii)* with which guarantees, and *(iii)* to which experiments. These three features define a user policy. Table 1 shows a number of examples. Allowing a user to grant specific experiments a higher level of access to his traffic has two advantages. First, an experimenter may be trusted by the user and be asking for higher privileges. Second, some experiments might offer particular benefits to the user, yet require certain types of control in order to work. This establishes a marketplace in which experimenters can compete for user traffic.

Guarantees a user may be interested in include "direct-delivery" as a baseline, which, by bypassing the testbed, ensures both complete privacy and no path impairment. Less strict guarantees comprise "anonymize" to rewrite endpoint addresses before injecting the traffic into the testbed, and "no-sniff" to deny the experimenter access to packet payload. Moreover, we also propose a "none" guarantee, i.e., the experimenter gets full access to the communication. Of course, these guarantees can be combined where reasonable.

In order to enforce the guarantees requested by the user, we propose to add a privacy and availability layer (PAL) between FlowVisor and the experimenter's controllers (Figure 1). The PAL acts as a transparent OpenFlow proxy, policing OpenFlow messages according to users' policies. Further, the PAL uses header space analysis (HSA) [1] to infer delivery paths of user traffic inside the testbed.

| Part of traffic | Guarantees | Experiment |
|---|---|---|
| Facebook | no-sniff | any |
| BitTorrent | none | transparent BT cache |
| E-Banking | direct-delivery | - |

**Table 1: Examples of possible user policies.**

For protection against availability threats the PAL needs to additionally monitor if packets injected into the testbed are eventually leaving the testbed towards their destination.

When detecting forbidden paths or excessive packet loss inside the testbed, we generate policy violation events. For handling these policy violations, we short-circuit the testbed and deliver the traffic directly between the gatekeepers, thus denying the attacker access to user traffic while ensuring network availability for the user.

## 4. INSIGHTS AND CONCLUSION

In this paper, we propose a marketplace where experimenters can attract users' traffic to their SDN testbed slices. We introduce a privacy and availability layer (PAL) to give guarantees to the user contributing his traffic. While implementing a demonstrator of the PAL we gained two main insights:

**OpenFlow-specific problems:** Preserving privacy in an OpenFlow testbed is more difficult than anticipated. We identified two technical reasons as culprits: *(i)* The rule (soft) timeout mechanism in OpenFlow forces the controller to regularly poll the switch if a timeout may happen soon in order to proactively suppress policy violations. Changing the timeout semantics to send notification messages and letting the controller decide how to act upon such a notification would remedy this problem. *(ii)* Flow-space analysis is exponentially expensive in the worst case. Moreover, the OpenFlow standard [4] is reluctant to specify how to resolve rule conflicts on the same priority level. Therefore, in our unfriendly environment an attacker has at least two easy ways to stage exponentially expensive attacks on HSA. We overcome these kinds of attacks by putting a hard limit on the effort we spend on HSA, and declaring an excess of this limit a policy violation in itself.

**Market place and user incentives:** Getting access to real-world traffic has always been hard for researchers. In addition to users' concerns, communication privacy laws are often undermining the attempt to analyze users' traffic. With the PAL, we enable users to voluntarily donate part of their traffic, and we enable researchers to offer incentives to users to do so. Incentives could include network features such as configurable tunnels with exits on any OFELIA [6] island, transparent network caches to improve network performance, IP anonymization services, or providing network usage statistics to the user, as well as real-world goods.

Future work includes integrating PAL into our OFELIA island, and extending the PAL to cross-island experimentation.

**Acknowledgments:** The work presented in this paper is partially funded by the EU FP7 project OFELIA [6].

## 5. REFERENCES

[1] M. Kazemian, Peyman and Chang, H. Zeng, G. Varghese, N. McKeown, and S. Whyte. Real time network policy checking using header space analysis. In *Proc. of USENIX NSDI*, 2013.

[2] A. Khurshid, X. Zou, W. Zhou, M. Caesar, and P. B. Godfrey. VeriFlow: verifying network-wide invariants in real time. In *Proc. of USENIX NSDI*, 2013.

[3] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner. Openflow: enabling innovation in campus networks. *SIGCOMM CCR*, 2008.

[4] ONF specifications. https://www.opennetworking.org/sdn-resources/onf-specifications.

[5] R. Sherwood, G. Gibb, K.-k. Yap, M. Casado, N. McKeown, and G. Parulkar. Can the production network be the testbed. In *Proc. of USENIX OSDI*, 2010.

[6] OFELIA. http://www.fp7-ofelia.eu/.