# SBAR: SDN flow-Based monitoring and Application Recognition

José Suárez-Varela
UPC BarcelonaTech, Spain
jsuarezv@ac.upc.edu

Pere Barlet-Ros
UPC BarcelonaTech / Talaia Networks, Spain
pbarlet@ac.upc.edu

## ABSTRACT

We present SBAR, a monitoring system compliant with OpenFlow that provides flow-level measurement reports similar to those of NetFlow in traditional networks, but additionally enriched with labels that classify flows at the application layer. For the sake of scalability, we implement flow sampling to control both, the processing overhead in SDN controllers and the memory needed in switches to maintain the flow measurements. Moreover, we leverage the particularities of OpenFlow networks to implement a combination of classification techniques based on DPI and Machine Learning without incurring in high overheads. In particular, we accurately classify the traffic at two different levels: (i) every monitored flow is classified by application protocol, and (ii) for web and encrypted traffic, we apply specific DPI techniques to identify the applications generating each flow. In our demo, we will use real-world traffic to generate flow-level reports with SBAR that are then processed by a commercial monitoring tool to provide a comprehensive high-level view of the traffic in the network [6].

## CCS CONCEPTS

• **Networks** → **Network measurement**; **Network monitoring**; **Programmable networks**; *Deep packet inspection*; • **Computing methodologies** → *Supervised learning by classification*;

## KEYWORDS

Software-Defined Networking, Traffic measurement, Traffic classification, OpenFlow

**ACM Reference Format:**
José Suárez-Varela and Pere Barlet-Ros. 2018. SBAR: SDN flow-Based monitoring and Application Recognition. In *SOSR '18: SOSR '18: Symposium on SDN Research, March 28–29, 2018, Los Angeles, CA, USA.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3185467.3190788

## 1 INTRODUCTION

In the Software-Defined Networking (SDN) paradigm, it is essential to perform comprehensive traffic monitoring in order to provide the control plane with an accurate view of the network state. This enables to perform such an effective fine-grained network management with different purposes (e.g., traffic engineering, security).

In traditional networks, one of the most deployed solutions for network measurement is NetFlow/IPFIX. There are plenty of tools in the market based on NetFlow that harness the flow-level measurement reports to produce traffic statistics useful for network management. Additionally, flows are often labeled (e.g., by protocol) using port-based classification techniques. However, these techniques are becoming obsolete in current network scenarios, where it is quite common to find very diverse applications operating over the same application protocols (e.g., web services).

Likewise, the QoE perceived by end-users significantly depends on the type of application and the QoS level provided by the network (e.g., bandwidth, delay). This reflects the necessity of a more comprehensive level of classification where the control plane is aware of the specific applications and services generating the traffic.

Regarding the latest trends in traffic classification, two lines can be mainly remarked. On the one hand, *Deep Packet Inspection* (DPI) typically achieves very accurate traffic classification by inspecting the packet payloads. However, applying DPI over all the packets traversing a network is often too resource consuming. This is even more unfeasible in SDN-based scenarios, where it is essential to preserve the processing power in controllers given that they are prone to become network bottlenecks. On the other hand, *Machine Learning* (ML) classifiers were proposed with the aim of alleviating the processing burden. These techniques typically use features up to the transport layer to classify the traffic. They are able to achieve similar accuracy than DPI when classifying the traffic by application-level protocols. However, they become useless when applied to distinguish among different applications generating traffic over the same protocol. This is, for instance, the case of web-based traffic (e.g., HTTP/HTTPS protocols), which nowadays constitutes the bulk of the Internet traffic. In this context, some novel techniques based on DPI were proposed to discover the specific applications (e.g., Netflix) within web and encrypted traffic by leveraging some data in the first few packets of the connections and in DNS queries sent by clients prior to establish connections [1, 2].

In view of this, our contribution is the design and implementation of SBAR, a SDN-based monitoring system compliant with OpenFlow that provides flow-level measurement reports including application labels. SBAR implements flow sampling, which permits to reduce the processing overhead in the controller(s) and the memory consumption in switches to maintain the measurements [3]. Moreover, we leverage the particularities of OpenFlow networks to efficiently implement a combination of techniques based on ML and DPI to accurately classify the traffic in the controller.

## 2 MONITORING SYSTEM

Fig. 1, shows the architecture of our SDN-enabled monitoring system. It can be logically divided in two blocks: (i) SBAR, our system based on OpenFlow that produces flow-level monitoring reports, and (ii) a data analytics tool (Sec. 2.3), which processes the resulting reports from SBAR and displays valuable network statistics through a web graphical user interface (GUI). SBAR, in turn, is composed by a *Measurement module* (Sec. 2.1), which is in charge of maintaining the flow measurements in the switches and report them to the

*SOSR '18, March 28–29, 2018, Los Angeles, CA, USA*

ACM ISBN 978-1-4503-5664-0/18/03.
https://doi.org/10.1145/3185467.3190788

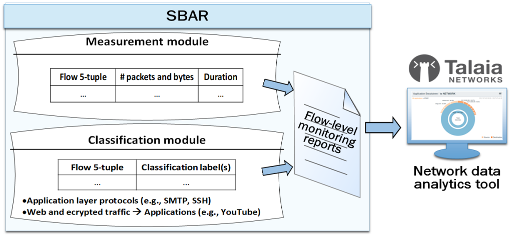
SBAR
Measurement module
Flow 5-tuple
# packets and bytes
Duration
...
Classification module
Flow 5-tuple
Classification label(s)
...
•Application layer protocols (e.g., SMTP, SSH)
•Web and ecrypted traffic → Applications (e.g., YouTube)
Flow-level monitoring reports
Talaia NETWORKS
Network data analytics tool

**Figure 1: Architecture of the monitoring system**

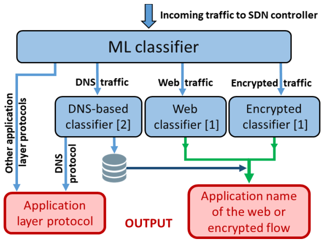
Incoming traffic to SDN controller
ML classifier
Other application layer protocols
DNS traffic
Web traffic
Encrypted traffic
DNS-based classifier [2]
Web classifier [1]
Encrypted classifier [1]
DNS protocol
Application layer protocol
OUTPUT
Application name of the web or encrypted flow

**Figure 2: Classification module**

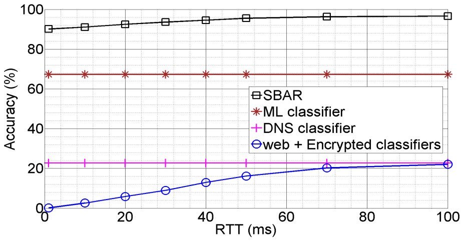

**Figure 3: Classification accuracy**

controller(s), and a *Classification module* (Sec. 2.2), which produces classification labels for each flow in the measurement reports.

## 2.1 Measurement module

This module generates flow-level measurement reports as those of NetFlow/IPFIX in traditional networks. To this end, we leverage the support of OpenFlow to maintain the flow measurements (# of packets and bytes, and duration) in the flow tables of the switches. Likewise, OpenFlow provides an interface that permits to report the measurements to the controller(s) when some predefined timeouts (idle and hard) expire. Moreover, we implemented flow sampling using only native features of OpenFlow, which enables to address common scalability issues in OpenFlow-based networks. Particularly, this allows us to control the processing power in controllers and the memory required in switches to maintain the measurements. Additionally, we make use of multiple tables of OpenFlow to decouple the operation of this module from other modules executing different network tasks (e.g., forwarding) in the controller. More details about this module are painstakingly described in [3].

## 2.2 Classification module

This module classifies the traffic at two different levels of detail: (i) it performs per-flow classification by application protocols (e.g., SMTP, SSH) using a ML model and, (ii) for web and encrypted flows, it applies specific DPI techniques [1, 2] to identify the applications (e.g., Netflix, Facebook) generating traffic. Fig. 2 outlines the architecture and operation of this module.

For our ML model, we based on the work in [4]. In particular, we use a C5.0 decision tree with the following features: src and dst ports, IP protocol and size of the first few packets (max. 6 packets). For the training set, we used application protocol labels provided by the open-source tool "nDPI".

Regarding the classification of web and encrypted traffic, we implemented the techniques in [1] to extract information from both, the *host* field in HTTP headers, and the *SNI* field in SSL/TLS certificates. To this end, we considered a scenario where controller(s) reactively install specific flow entries for web and encrypted flows. This enables to receive (and process) in the controller only the first few packets of these flows, where typically HTTP headers and SSL/TLS certificates are present. Additionally, we process the DNS traffic to apply the technique in [2], which permits to infer the applications associated to specific flows. This combination of techniques allows us to identify the applications generating web and encrypted traffic without producing a high processing overhead.

We evaluated this module with real-world traffic. Fig. 3 shows the results of the classification accuracy achieved by SBAR as well as the individual contribution of the different classifiers to the overall accuracy. These accuracy results mainly depend on the RTT between the switches and controllers, as it has an impact on the amount of packets per-flow that are inspected in the controllers. This, in turn, affects to the overhead contribution of the monitoring system. This issue is described in more detail in [3].

## 2.3 Network data analytics tool with GUI

We adapted a commercial network monitoring tool [5] based on NetFlow to process the flow-level reports produced by SBAR and harness the information within them to infer high-level traffic statistics. This provides a comprehensive view of the network traffic including graphs with statistics such as the percentage and volume of traffic by application protocols and by specific applications for web and encrypted traffic, or the identification of the top N talkers. All the results are presented via an intuitive web application.

# 3 DEMONSTRATION

We implemented our *Measurement module* (Sec. 2.1) within the OpenDaylight controller and deployed it in a testbed using Open vSwitch. Furthermore, we integrated in the SDN controller our *Classification module* (Sec. 2.2). To this end, we used the following tools: (i) C5.0 decision tree to classify the flows by application protocol, and (ii) "Bro IDS" to process DNS, HTTP and encrypted traffic in order to apply the techniques in [1, 2] to identify the applications. Lastly, the flow-level reports produced by SBAR are sent to a data analytics platform (Sec. 2.3) to be processed.

In our demo [6], we will use real-world traffic from a 10 Gbps access link of a large Spanish university, which connects about 25 schools and 40 departments to the Internet. We will show the traffic statistics via a user-friendly web application with the aim of illustrating how can we benefit from the reports provided by SBAR to perform an effective network management in SDN environments.

**Acknowledgements:** This work was supported by the European Union's H2020 SME Instrument Phase 2 project "SDN-Polygraph" (grant agreement n° 726763), the Spanish Ministry of Economy and Competitiveness and EU FEDER under grants TEC2014-59583-C2-2-R (SUNSET project) and TEC2017-90034-C2-1-R (ALLIANCE project), and by the Catalan Government (ref. 2014SGR-1427).

# REFERENCES

[1] M. Goossens, et al. "Towards web service classification using addresses and DNS." *In IWCMC, 2016, pp. 38-43.*
[2] T. Mori, et al. "Statistical estimation of the names of HTTPS servers with domain name graphs." *Computer Communications, vol. 94, pp. 104-113, 2016.*
[3] J. Suárez-Varela, and P. Barlet-Ros, "Towards a NetFlow implementation for OpenFlow Software-Defined Networks." *In 29th IEEE ITC, vol. 1, pp. 187-195, 2017.*
[4] V. Carela-Español, et al., "Analysis of the impact of sampling on NetFlow traffic classification." *Computer Networks, Vol. 55, no 5, pp. 1083-1099, 2011.*
[5] Talaia product, https://www.talaia.io
[6] Demo web application video, https://goo.gl/s2qeYN