OpenFlow-based Link Dimensioning

Ricardo de O. Schmidt*, Luuk Hendriks*, Aiko Pras* and Ronald van der Pol‡

*University of Twente, The Netherlands
Email: {r.schmidt,luuk.hendriks,a.pras}@utwente.nl

‡SURFnet, The Netherlands
Email: ronald.vanderpol@surfnet.nl

***Abstract*—In this demo we will demonstrate the possibility of using OpenFlow traffic measurements for link dimensioning purposes. Our solution runs on top of the Ryu OpenFlow controller and retrieves per-flow statistics metered at the OpenFlow switch. The statistics are obtained by using messages defined by the OpenFlow protocol. These statistics are then applied to a flow-based link dimensioning approach, originally proposed to operate with NetFlow input. By demonstrating our solution in a testbed, we are able to compare the OpenFlow-based approach with a NetFlow-based one and with the actual traffic demands calculated directly from the packet traces. With that, we show how quality of OpenFlow measurements affects the link dimensioning and how feasible their use in such applications is.**

I. Overview

The ever increasing demand for bandwidth resources and the trend towards virtualizing services and even networking infrastructures present many challenges for the management of the future Internet. We envision scenarios in which major Internet companies (*e.g.,* Google, Apple and Amazon) will use existing infrastructure to provide end-to-end dedicated services to their customers, creating their own *Internet ecosystems*. These scenarios bring challenges ranging from financial to ethical, political and technological. Concerning this last one, we believe that link dimensioning can be of utmost importance for NRENs and commercial providers to fairly allocate and share bandwidth resources among multiple tenants. In addition, link dimensioning can also support operations such as load balancing.

On the one hand, although easy to deploy, simplistic approaches based on interface counters do not provide enough data to accurately estimate required capacity for a given link. Accurate link dimensioning approaches on the other hand, such as [1], [2], are difficult to deploy because they often demand traffic measurements at the packet level, which is financially and operationally challenging (*e.g.,* due to increasing volume of data in high-speed links). Aiming at an easy to deploy solution, we have proposed a flow-based link dimensioning approach [3]. This approach calculates required link capacity from NetFlow-like traffic measurements.

Motivation. Triggered by the increasing interest of industry and academia on SDN (Software-Defined Network), we decided to investigate the possibility of using OpenFlow traffic measurements for link dimensioning purposes. OpenFlow specification [4] defines basic per-flow statistics counters to be maintained by the OpenFlow switch. These statistics provide enough information as needed by the flow-based link dimensioning in [3]. Since the flexible definition of flows in OpenFlow allows us to define the *match fields* in the same format of NetFlow v5 flow key, similar traffic measurements can be expected from OpenFlow as compared to NetFlow.

II. Approach

We will demonstrate an OpenFlow-based link dimensioning. This approach applies OpenFlow traffic measurements to the flow-based link dimensioning approach from [3]. Our solution solely relies on per-flow statistics metered at the switch and sent to the controller via the OpenFlow protocol. Therefore, we are only interested on a handful of messages exchanged between controller and switch that carry measurement-related information. Figure 1 shows these messages and the processes that generate them.

For every incoming packet the OpenFlow switch checks whether the packet information matches with a flow entry in the flow table (note that there might be multiple flow tables). If no match is found for the packet, the switch sends to the controller a `OFPT_PACKET_IN` message. The controller decides on forwarding actions and replies to the switch with a `OFPT_FLOW_MOD` message of type `OFPFC_ADD`. On receiving the `OFPT_FLOW_MOD` message the switch installs a new entry in the flow table and starts accounting for packets that match to it. The add message contains, among others, the *match fields* and the timeouts, namely hard and soft timeouts. The *hard timeout* defines for how long an entry should remain in the flow table even in case of active flows (same as NetFlow's *active timeout*). The *soft timeout* tells the switch how long the entry should remain in the flow table after the last matching packet was seen (same as NetFlow's *idle timeout*).

Notice that our approach requires `OFPT_FLOW_MOD` messages, for adding or modifying flow table entries, to have the flag `OFPFF_SEND_FLOW_REM` set. This flag tells the OpenFlow switch that the controller expects to receive a `OFPT_FLOW_REMOVED` message for each flow table entry that is removed due to timeout expiration. Among others, this message tells us for how long the entry was active in the flow table and how many packets matched to the flow entry and the sum of bytes of matched packets.

Another way to obtain the per-flow statistics counters from the OpenFlow switch is by explicitly requesting them. To do so, we ask the OpenFlow controller to send a `OFPT_MULTIPART_REQUEST` message of type `OFPMP_FLOW`, specifying the target flow table. On receiving this message, the OpenFlow switch replies with a `OFPT_MULTIPART_REPLY` message containing the statistics counters of all active entries in the requested flow table.

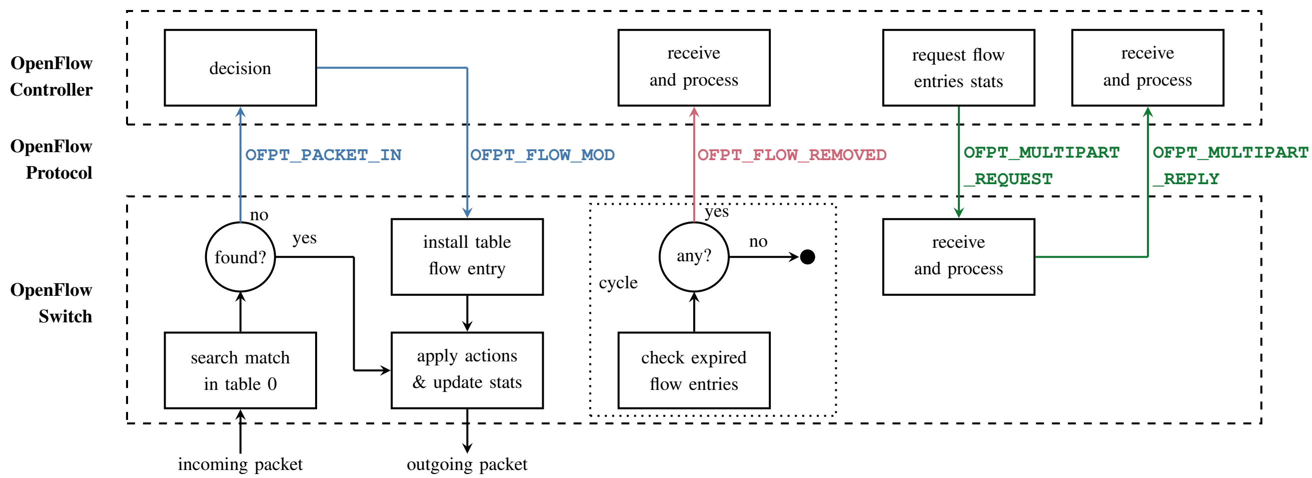

Fig. 1. Messages exchange between OpenFlow controller and switch.

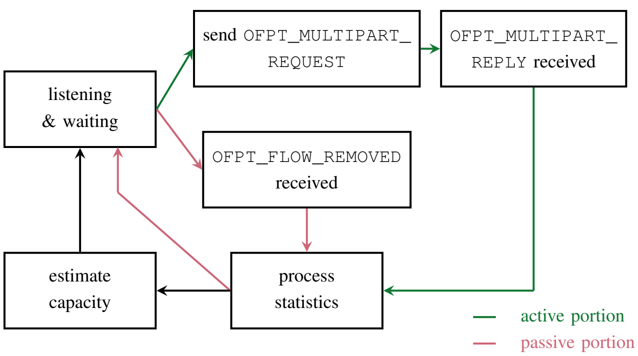

Fig. 2. Proposed approach running on top of the OpenFlow controller.

Figure 2 shows the steps of our proposed OpenFlow-based approach and when the above described messages are used. The approach can be divided in two portions, namely passive and active portions. The former consists of passively receiving the per-flow statistics via the OFPT_FLOW_REMOVED messages from the switch to the controller. The latter consists of actively retrieving the information by sending a OFPT_MULTIPART_REQUEST from the controller to the switch. The per-flow measurements are then applied to the flow-based link dimensioning approach from [3], which ultimately results in estimations of required link capacity based on the measured traffic by OpenFlow.

III. Setup

This demo is implemented using OpenFlow 1.3. Our physical testbed consists of a Pica8 Pronto 3295 OpenFlow switch running PicOS 2.3 and four hosts connected to the switch: one configured as OpenFlow controller (Ryu) and the other three as nodes of a simple load balancing scenario that will use of the OpenFlow-based link dimensioning approach. The demo also includes a virtual setup, consisting of the same topology, with a VM running OVS 2.1.2. This allows us to compare traffic measurements obtained from PicOS with those obtained from pure OVS. In addition, for comparison purposes we will demonstrate a NetFlow-based and a packet-based link dimensioning approach. This allows us to validate both the quality of OpenFlow measurements and accuracy of the link dimensioning.

IV. Relevance

As the use of OpenFlow is increasing in a plethora of different fields, the quality of statistics obtained from switches is becoming more important. Most works assume counters in OF-switches to be accurate, but initial experiments show that is not true per se. Basic but essential applications of these statistics such as accounting, network forensics or link dimensioning can easily be impaired when fed the wrong input. Envisioning these and other applications to be based on OpenFlow statistics in the future, we underline the importance of trustworthy numbers. Our contribution of this demo and its relevance is therefore twofold. First, we show the quality of traffic measurements obtained from counters in OpenFlow and second, we validate the use of OpenFlow measurements for link dimensioning purposes.

V. SCInet Requirements

There are no required SCInet network resources. The prototype is implemented on testbeds in Europe. We only need Internet connection with at least 1 Gbps to to connect to our servers and provide visualization of the demo operation.

Acknowledgments. EU FP7 FLAMINGO (318488), EU FP7 MCN (318109) and SURFnet GigaPort3+ (RES2014/0044).

References

[1] H. van den Berg, M. Mandjes, R. van de Meent, A. Pras, F. Roijers, and P. Venemans, “QoS-aware bandwidth provisioning for IP network links,” *Elsevier Computer Networks*, vol. 50, no. 5, pp. 631–647, 2006.

[2] A. Pras, L. J. M. Nieuwenhuis, R. van de Meent, and M. R. H. Mandjes, “Dimensioning Network Links: A New Look at Equivalent Bandwidth,” *IEEE Network*, vol. 23, no. 2, pp. 5–10, 2009.

[3] R. de O. Schmidt, A. Sperotto, R. Sadre, and A. Pras, “Towards Bandwidth Estimation using Flow Measurements,” in *Proceedings of the 6th IFIP WG 6.6 International Conference on Autonomous Infrastructure, Management, and Security*, ser. AIMS, 2012, pp. 127–138.

[4] Open Networking Foundation, “Openflow switch specification – version 1.4.0,” https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-spec-v1.4.0.pdf, 2013, online. Accessed Jun. 2014.