Open access • Proceedings Article • DOI:10.1109/INFCOMW.2014.6849191

Demonstrating Resilient Quality of Service in Software Defined Networking

Sachin Sharma, Dimitri Staessens, Didier Colle, David Palma ...+4 more authors

Institutions: Ghent University

Published on: 08 Jul 2014 - International Conference on Computer Communications

Topics: Software-defined networking, OpenFlow and Quality of service

Demonstrating Resilient Quality of Service in Software Defined Networking

Sachin Sharma[1], Dimitri Staessens[1], Didier Colle[1], David Palma[2], Joao Goncalves[2],
Mario Pickavet[1], Luis Cordeiro[2], and Piet Demeester[1]
[1]Department of Information Technology (INTEC), Ghent University - iMinds,
[1]Email: {firstName.LastName}@intec.ugent.be
[2]OneSource, Consultoria Informatica, Lda, Coimbra, Portugal,
[2]E-mail: {palma, joagonca, cordeiro}@onesource.pt

***Abstract*—Software defined Networking (SDN) such as OpenFlow decouples the control plane from forwarding devices and embeds it into one or more external entities called controllers. We implemented a framework in OpenFlow through which business customers receive higher Quality of Service (QoS) than best-effort customers in all conditions (e.g. failure conditions). In the demonstration, we stream video clips (business and best-effort customer's traffic) through an emulated OpenFlow topology. During the demonstration, we trigger a failure in the paths of video clips and show an effectively higher QoS for business customers when compared against best-effort customers. This is demonstrated by simply watching the video clips at the receiver.**

I. Introduction

Nowadays, providing users with a guaranteed Quality of Service (QoS), meeting the service level agreements, is of paramount importance. However, implementing such a QoS system is challenging in the current Internet. This is because in the current Internet, each forwarding device runs its own control plane software to make the forwarding decisions, lacking a broader picture of network resources, and there is no standard protocol available to configure QoS parameters in the forwarding devices. In this environment, a QoS provider (e.g. bandwidth broker in a single autonomous system) uses vendor-specific protocols to configure QoS parameters. However, not all forwarding devices support all of these protocols.

The Software Defined Networking (SDN) approach, such as OpenFlow [1], is one of the emerging Future Internet technologies in which control plane software is removed from all forwarding devices (switches or routers) of a network and is embedded into one or more external entities called controllers. In OpenFlow, the available network resources can be known by simply requesting the controller and in addition, there are standard protocols (OpenFlow configuration protocols [1], [2]) available to configure QoS parameters.

We implemented a QoS framework in OpenFlow, which divides different types of traffic (business and best-effort traffic) into different flows (services), configures priority queues, and redirects different flows to a suitable priority queue. Upon a failure, our framework reconfigures the network and provides high QoS to the business customers. In future versions of OpenFlow, namely since version 1.3, flow-related meters can also be used in this framework.

In the demonstration, we stream video clips (business and best-effort traffic) in an emulated OpenFlow pan-European topology, and show that business customers achieve high quality of service than best-effort customers using our framework. In addition, during the demonstration, we trigger a failure in the paths of video clips and show an effectively higher QoS for business customers as compared to best-effort customers.

978-1-4799-3360-0/14/$31.00 2014 IEEE

II. Resilient QoS framework for OpenFlow

In our framework, we use the OpenFlow protocol together with the OVSDB (Open vSwitch Database Management Protocol) configuration protocol [2] to provide high QoS. The OpenFlow protocol is used to divide different types of traffic into different flows and to redirect these flows through a suitable priority queue. The configuration protocol is used to configure suitable priority queues in the OpenFlow routers (or switches). Both of these protocols are used between the controller and the OpenFlow switches. As the current controllers such as Floodlight do not support the OVSDB protocol, the Floodlight controller is extended to support this feature. In addition, for communication with a QoS provider, we use the Northbound API (Application Program Interface) of the controller and for routing, we use a standard routing protocol (OSPF, Open Shortest Path First). Furthermore, for running OSPF in OpenFlow, we rely on our previously presented framework [3] for RouteFlow [4].

Starting on an OpenFlow router, three queues are configured on each port of the OpenFlow router. The first queue has the highest priority and therefore, traffic from this queue is forwarded first, then from the second queue, and so on. The first queue is configured to traverse control traffic, the second queue is configured to traverse business traffic, and the third queue is configured to traverse best-effort traffic. The traffic is called business traffic, if the TOS (Type of Service) field of the traffic is enabled. The traffic is called best-effort traffic, if the TOS field is not enabled. The edge OpenFlow router enables the TOS field of business traffic.

When the controller, running the RouteFlow framework, discovers a new routing entry for an OpenFlow Router, the controller establishes two corresponding forwarding entries (flow entries) on the router. With the first flow entry, business traffic (TOS field enabled) is traversed through the second queue (configured above), and with the second flow entry, best-effort traffic (TOS field disabled) is traversed through the third queue (configured above).

When a QoS provider receives a request to reserve a bandwidth from a business customer, a confirmation regarding

the availability of network resources is performed through the NorthBound API of the controller. If the resources are available on the path retrieved by OSPF, a rate limiter queue (Q) having the same priority as the second queue is configured on the edge router. Moreover, in order to enable the TOS field of business traffic and to redirect this traffic to the rate limiter queue, a forwarding entry is established on the edge router.

Upon a failure, the flow entries on the affected paths are re-established and the edge router reconfigures its rate limiter queues appropriately, along the available alternative path.

A. Results and Discussions

In order to assess the described framework, experiments were performed on the OFELIA testbed facility provided by iMinds [5]. Fig. 1A represents an emulated pan-European

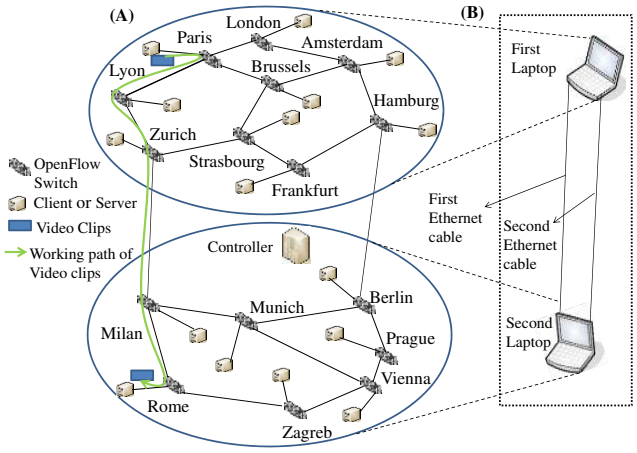

Fig. 1. (A) Emulated Pan-European Topology (B) Portable Testbed

topology. Each switch in the topology makes an out-of-band connection with a single controller. For emulation purposes, Open vSwitch was used as OpenFlow software and RouteFlow with our QoS framework was used as controller software. The bandwidth capacity of each link in the topology was limited to 50 Mb/s. In the experiments, each server sent both business traffic (30%) and best-effort traffic (70%) to all other servers in the topology using DITG (Distributed Internet Traffic Generator) [6]. In order to assess the framework, the rate of both traffic sources, following a Poisson distribution, was varied and one of the links was torn down. Afterwards, the effects on QoS of these operations on business and best-effort traffic were thoroughly analyzed. Regarding failure recovery, we do not focus on providing fast-failure recovery [7] but instead, we focus on the scenarios in which high-priority traffic always gets a higher precedence than best-effort traffic.

Three distinct scenarios of business traffic were analyzed: low data-rate (less than 2.4 Mb/s of business traffic from each server to other server); medium data-rate (between 2.4 and 7 Mb/s); and high data-rate (more than 7 Mb/s). For the low data-rate scenarios, neither business nor best-effort customer traffic received the degraded service. This was because the failure-free path (before and after the link down) had enough bandwidth to accommodate both business customer and best-effort traffic. In the medium rate scenarios, only best-effort traffic received the degraded service. This was because the failure-free path (before and after the link down) had only enough bandwidth for business traffic. As a result, some of the best-effort traffic had to drop in order to meet the requirements of the business traffic. Finally, for the high data rate scenarios, business traffic had also received the degraded service after the link down. This was because in this scenario, some of links in the failure-free path after the link down had not the enough bandwidth to accommodate all the business customer's traffic. Therefore, some of business traffic was also dropped. In these links, we observed about 0 Mb/s best-effort traffic.

III. Demonstration on Portable Testbed

With the portable testbed (two laptops, Fig. 1), we show the working of our QoS framework using an emulated pan-European topology. With the Mininet software [8], the half of the topology is emulated on the first laptop and the other half is emulated on the second laptop. The connection between the emulated topologies on different laptops is done using two Ethernet cables, shown in the figure. The controller, which runs our framework, is located on the second laptop. The controller controls all the switches of the topologies including the switches on the first laptop by the second Ethernet cable.

The link and traffic characteristics in the demonstration is replicated from the scenarios presented in the previous subsection. Hence, DITG is used to send business and best-effort traffic from each server. In addition, the server connected to Paris (which is present on the first laptop) streams two video clips – one as business traffic and the other as best-effort traffic – to the server connected to Rome (which is present on the second laptop). These video clips follow the path through the first Ethernet cable of the laptops.

In the demonstration, we show all the three scenarios presented in the previous subsection by simply watching the video clips of business and best-effort traffic on the second laptop. These three scenarios are shown by varying business and best-effort traffic (DITG traffic) from each server of the topology. For a failure condition of these scenarios, during the demonstration, we remove the first Ethernet cable of the laptops (the working path of video clips) and show switching of the video clips from the first Ethernet cable to the second Ethernet cable. After the failure, we show that business traffic always gets better QoS than best-effort traffic.

Acknowledgment

This research has received funding from the EU FP7 under agreement n^o 317576 (CityFlow), and n^o 258365 (OFELIA).

References

[1] OpenFlow and OF-ConFig: https://www.opennetworking.org/

[2] B. Pfaff et al., The Open vSwitch Database Management Protocol, IETF, 2013 (http://tools.ietf.org/html/draft-pfaff-ovsdb-proto-04)

[3] S. Sharma et al., Automatic configuration of routing control platforms in OpenFlow networks, ACM SIGCOMM, Vol. 43(4), pp. 491-492, 2013

[4] RouteFlow code: https://sites.google.com/site/routeflow/

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

[6] A. Botta et al., A tool for the generation of realistic network workload for emerging networking scenarios, Computer Networks, 2012

[7] S. Sharma et al., OpenFlow: Meeting carrier-grade recovery requirements, Computer Communications, Vol. 36(6), pp. 656-665, 2013

[8] Mininet Software: http://mininet.org/