Automatic QoS Management on OpenFlow Software-Defined Networks

Airton Ishimori, Fernando Farias, Igor Furtado, Eduardo Cerqueira, Antônio Abelém
Federal University of Pará
{airton, fernnf, igor, cerqueira, abelem}@ufpa.br

***Abstract*—OpenFlow/SDN is emerging as one of the most promising and disruptive networking technologies of recent years. However, new modules and improvements must be done to allow the usage of OpenFlow in large scale, such as Quality of Service (QoS) components. In its current version, OpenFlow is not able to configure QoS parameters in a dynamic and on-demand manner (only manual), which is not required for networking proposal. Therefore, this paper introduces a novel QoSFlow framework to enhance QoS management procedures in OpenFlow networks.**

***Index Terms*—Sofware-Defined Network; OpenFlow; QoS.**

I. Introduction

Despite the success of the Internet, IP core technology is the cause of its own limitations and nowaday they are becoming more evident due to introduction of new techonologies and services. The main goal of these activities, which can be described as the Future Internet (FI), is to formulate and evaluate alternative architectures to replace or complement Internet architecture.

However, the major challenges are "where will the proposed approaches be enabled and tested?" and "How could we do it without sacrificing the current infrastructure?". Other challenges are related to the implementation of new components into routers and switches as well as the creation of resource allocation and monitoring schemes in real-scale-experimentations facilities. In order to solve the problems mentioned, the best option is to design and validate novel proposal in FI Testbeds based on SDN (Sofware-Defined Network) architecture.

In the SDN architecture, the control and data planes are decoupled, network intelligence and state are logically centralized, and the underlying network infrastructure is abstracted from the applications[1]. The main component of the architecture is the OpenFlow protocol [3], whose components supported on network device are used between control and data planes.

As a result, OpenFlow is the first standard interface designed specifically for SDN, providing high-performance, granular traffic control across multiple vendors' network devices, besides the enterprises and carriers gained unprecedented programmability, automation, and network control, enabling them to build highly scalable, flexible networks that readily adapt to changing business needs [2].

Nevertheless, the OpenFlow is not mature enough to allow to solve some challenges on both FI or FI experimentation networks, where new modules must be implemented and tested. The implementation of QoS (Quality of Service) components into Openflow devices are required to enhance experiments about networking proposal, such as resource reservation, buffer control and monitoring.

In order to deal with QoS problems in switches Openflow, this paper proposes QoSFlow, a framework that enables QoS management in Openflow environment. QosFlow adds new QoS functions and allows the management of class and queues through rules or policy. These functionalities assure to manage QoS resources (e.g., bandwidth, queue size or delay), without changing the SDN achitecture, in other words, all actions are invoked by an Openflow controller (control plane) and in a dynamic manner. This proposal has a great potential to realize the proposition and evaluations of technical models for delay-sensitive applications, such as real-time and multimedia content.

The rest of the paper is structured as follows. Section 2 introduces QoSFlow architecture. In Section 3, conclusions and future work on the proposal are presented.

II. QoSFlow Architecture

The QoSFlow architecture is modular, flexible and composed of 2 components: the QoSFlow controller Fig. 1(i) and the QoSFlow datapath Fig. 1(ii), where components are distributed in modules with specific goals in the architecture. The description of these parts are in the following sub-sections below.

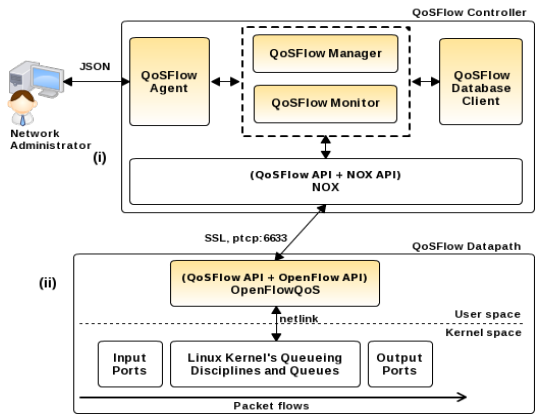

Figure 1. The QoSFlow Architecture

A. QoSFlow Controller

The QoSFlow controller is based on NOX (most used operation system for OpenFlow network), which is part of the proposed software and runs in a remote server. NOX is responsible for managing and monitoring actions and for controling signalling messages.

In order to enable NOX to manage QoS resources of all Openflow infrastructure, new QoS primitives have been added into the Openflow implementation. They are a new group of control messages and functions able to manage QoS resources on data plane. They can be invoked by a component running a application with QoS aspects over the openflow switches.

This new controller, besides NOX API, is composed by new components as follows: **QoSFlow Agent**, **QoSFlow manager**,**QoSFlow monitor** and **DB-QoSFlow client**. These four modules have been designed to extend the NOX API with QoS features called QoSFlow API.

QoS Agent is responsible for creating a communication module between a administrator management tool and the other two QoSFlow components, the manager and monitor QoSFlow. By using JSON interface, the agent is able to receive policies, manage or monitor commands coming from a third-part administrator application.

The **QoSFlow monitor** and **manager** components, respectively, monitor and manage the QoS of OpenFlow domains. However, these two modules run just after the decision of QoSFlow Agent. In other words, the agent chooses the right component to be used. This choice depends on the action sent by the network administrator. There are available four new actions OpenFlow that is able to configure class, filters, qdisc (queue disciplines) and QoS statistics. For example, the administrator could create a traffic shaping on an ethernet port using HTB qdisc for flow class or allocate dynamically queue size of a class.

Finally, the **DB-QoSFlow client** gives support for monitoring and management schemes, enabling querying, inserting, removing or updating of registered information from resources in the database.

B. QoSFlow Datapath

The QoSFlow datapath component called OpenFlowQoS is responsible for creating all low-level actions on the switch ports, and it is based on the original Openflow datapath (the current implementation lacks QoS functions).

This component allows openflow to get all the required primitives to run management commands created by either the administrator's tool or through header packet information.

In QoS management tool, the actions are processed in the QoSFlow Agent. When receiving those actions, it checks the type (management or monitoring) of the received requests in order to select the procedure to be done (QoS control message and action). This new message is automatically sent to OpenFlowQoS through NOX.

The QoS actions can be applied automatically through the packet header information. For example, the bandwidth configuration can be set for each flow type (traffic shaping), gathering resources on demand to avoid losing of information because of concurrence on switch port queues.

The list below is not exhaustive, but it aims to show our available components to be used for QoSFlow administrators. It introduces the types of problems that can be solved by using QoSFlow to maximize the usage of network resources.

- Limit total bandwidth to a known rate;
- Limit the bandwidth of a particular user, service or client;
- Reserve bandwidth for a particular application or user;
- Manage oversubscribed bandwidth;
- Allow equitable distribution of unreserved bandwidth;

C. QoSFlow policies

QoS policies in QoSFlow allow the administrator to manage an administrative domain, including tasks such as the mapping of configuration models in a low level, scaling the management of hundreds of entities and the control behavior with end-to-end desirable features. In this context, those policies are high-level rules abstraction, defining the behavior of a system, in which a set of instructions may be called when prearranged conditions are satisfied. QoSFlow has an administrative interface to set these policies, which they will be converted in low-level configuration, turning the definition of strategies of QoS management on switches easier.

The polices follow the standards established by RFC's such as 3703, 3060, 3460 and some drafts for policy schemes (Policy Core Schema), besides of other instances to store and create SLA (Service Level Agreement). For these other instances, it is expected that the QoSFlow software, in a QoS context, deals with the business among the involved parts, provisioning agreement level of end-to-end services for both Openflow and non-Openflow domain. By doing this, the resources are formally allocated through contracts provided by QoSFlow.

III. Conclusions and Future Work

QoSFlow is under development and it intends to make QoS management operations on-demand. It will avoid manual configuration tasks on each switch on the network, making the need of external tools unnecessary. Besides these advantages, the software offers novel conditions to define, in a high-level of abstraction, management of QoS policies.

After finishing QoSFlow development, the first tests are going to be done using Openflow switches available at GERCOM/UFPA lab and, after, running some tests using the FIBRE testbed, which is a project involving Brazilian and European universities - FP7 CNPq.

References

[1] Bob Lantz, Brandon Heller, and Nick McKeown. A network in a laptop: rapid prototyping for software-defined networks. In *Proceedings of the Ninth ACM SIGCOMM Workshop on Hot Topics in Networks*, pages 19:1–19:6, New York, NY, USA, 2010. ACM.

[2] Nick McKeown and et. al. Openflow: enabling innovation in campus networks. *SIGCOMM Comput. Commun. Rev.*, 38(2):69–74, March 2008.

[3] Rob Sherwood and et.al. Carving research slices out of your production networks with openflow. *SIGCOMM Comput. Commun. Rev.*, 40(1):129–130, January 2010.