

## **An SDN Approach: Quality of Service using Big Switch's Floodlight Open-source Controller**

**Ryan Wallner<sup>1</sup> and Robert Cannistra<sup>2\*</sup>**

<sup>1</sup> Marist College  
3399 North Road  
Poughkeepsie, NY 12601

E-mail: [ryan.wallner1@marist.edu](mailto:ryan.wallner1@marist.edu), [robert.cannistra@marist.edu](mailto:robert.cannistra@marist.edu)

\* Author to whom correspondence should be addressed; Tel.: +1-845-575-3675

---

### **Abstract:**

Quality of Service in today's packet-switched networks have various effecting factors that include both technical and non-technical aspects. Such factors include the reliability of service, availability of service, delay, scalability, effectiveness, grade of service etc. These factors play an important role into how a particular network may treat and handle packets within a given network. Properties of packet-switched networks can cause low throughput, dropped packets, corruption errors, latency, jitter, disordered delivery and more. The role of SDN allows for network operating systems to gain greater governance of the control plane within a given network. Using this SDN approach and the specifications provided by OpenFlow (OF) 1.0[1] we show an approach to Quality of Service that is managed and defined by a centralized network controller.

**Keywords:** Software-Defined Networking, SDN, Floodlight, Quality of Service, DiffServ Queue, Network

## 1. Introduction

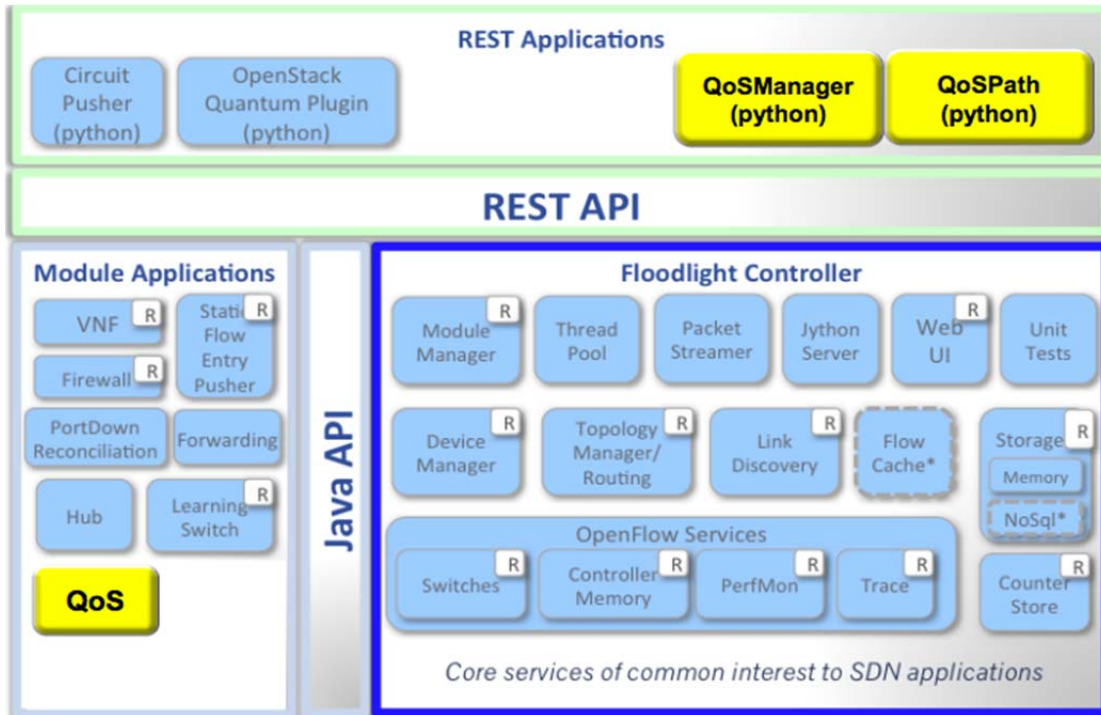
Delivering quality of service has historically been an issue of operating expense, risk and requirements to deliver differentiated types of quality to end-users otherwise known as the consumer. Technologies, such as VoIP and services like Netflix™ and Amazon™ Instant Video, not only need quality of service to guarantee clarity and effectiveness, but they depend on these necessary qualities within their networks to run their business. While efforts in this area over the past include ToS (Type of Service), Overprovisioning, IntServ (Integrated Services), MPLS (Multiprotocol Label Switching), RSVP (Resource Reservation Protocol), DiffServ (Differentiated Services) and Traffic Shaping; this paper will only focus on the latter of the two. Specifically this will focus on how Traffic Shaping (rate limiting) and DiffServ DSCP (Differentiated Services Code Point) based classification approaches can be used within an SDN network using BigSwitch's Floodlight open-source controller [2] and the virtual switch, Open vSwitch [3].

## 2. Quality of Service and the Adoption of SDN

We must consider techniques in which traditional networking hardware and software provide quality of service to Ethernet and TCP traffic and apply it to the new realm of software-defined networking. Software-defined networking and virtualization can have different meanings depending on which person is on one end or the other. Therefore, for the reasons defined in this paper, SDN refers to the architecture and model in which the underlay/physical network consists of OpenFlow 1.0 capable switches with a northbound connection to a NOS (Network Operating System). This network operating system has northbound and southbound APIs that allow networking equipment and network applications to communicate over a common control plane, the controller.

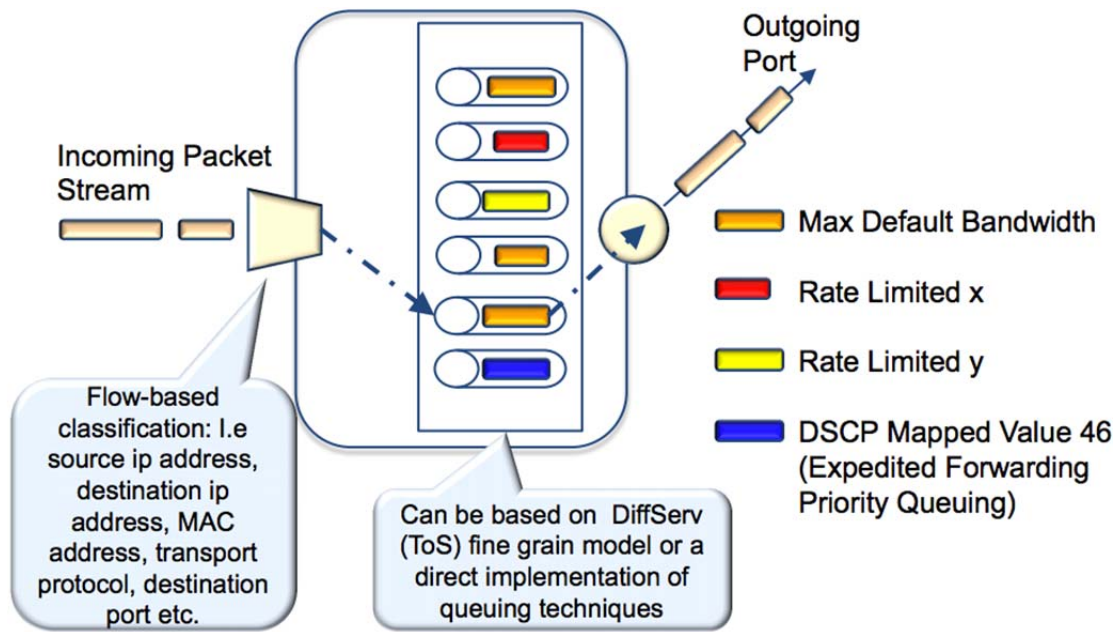
The quality of service mechanisms that we will use to approach a software-defined quality of service module will be DiffServ DSCP [4,5] and the use of common queuing techniques [6] within Open vSwitch. In order to achieve a software-defined approach to both, we define network wide class of service as well as rate limiting paths through a given network of N switches. The first mechanism we will explore in relation to this module is Differentiated Services Code Point. This DSCP value was a predecessor the ToS bits within a TCP header. The 8 ToS bits in the IP header now include both IP Precedence and the DSCP values. These DSCP values are used to declare a "Class" of service across a given network segment. The approach starts by providing a "module" inside the Floodlight controller that will do the brunt of the matching, classification, flow insertion, flow deletion, and policy handling for QoS. The modularized architecture can be referenced and found here [7]. QoS is added to the architecture as shown in figure 1 and has the responsibilities of tracking and storing classes of services with their associated DSCP values, applying policies that either take advantage of a class of service or apply a type queuing technique on queues attached to ports on a switch. The module also has the responsibility of tracking which policies are in each switch while verifying there are no

duplicates. The module will also track any topology updates that may skew the QoS policies within a network and notify the network administrator. The module takes advantage of two actions within the OF 1.0 spec: “Enqueue” and modifying the “Network ToS” bits. As new specifications are released, the advancement in software will need to take place; however the immediate idea is to take advantage of the most stable OpenFlow specification of the current date.



**Figure 1.** *QoS Module & Tools in Floodlight Architecture*

The application as a whole does not have to use the adapted python application that utilize its northbound API; however in this case they are used to demonstrate how we can manage QoS and apply provisional reservation-like QoS paths through a network. Essentially the module itself allows the network to define two types of policies: a “Queuing Policy” and a “ToS/DSCP Policy”. A queuing policy utilizes the enqueue action in OF 1.0 to enqueue certain types of flows within a network that match on certain criteria (defined by the OF 1.0 match structure [8]) and send them to a queue on a switch port that has a given action of its own, such as rate limits, minimum bandwidth and/or bandwidth ceilings. Figure 2 demonstrates a brief representation of how this may be represented with an ingress flow to an OpenFlow switch; a classification based match process, an enqueueing process, and an egress flow of packets.



**Figure 2.** *Queue based classification in OVS*

The actual configuration of the queues within a given switch is still the responsibility of the network administrator and a CLI-based configuration process. Fortunately, there is a protocol currently under development that seeks to rid this process by providing a control and data-plane-like separation for the configuration of queues, ports, and other QoS based arrangements. This protocol is called OF-Config [9]. OF-Config is not covered in this paper, though may prove to be valuable overtime. The class based method uses a class of service that can be defined with a given name and an associated DSCP value. Please see example 1.

### Example 1:

```
{ "name": "Expedited Forwarding", "tos": "101000" }
```

```
{ "name": "Best Effort", "tos": "0" }
```

These classes of service can then be associated within a policy that may hypothetically match on all traffic from a given subnet that is using the Ethernet protocol and apply a “Best Effort” based service on this type of traffic throughout an entire network using the QoS module and manager. The QoSPath application allows the addition of both types of policies along a given Dijkstra-

based shortest path first routing circuit through a network using a “circuitpusher” based application which ships with the open-source Floodlight controller.

### 3. Conclusions

SDN and OpenFlow are at the anterior side of the networking industry and allow for the adoption of traditional-based networking techniques like QoS, load balancing and routing to SDN approaches. In years past, it would take sizeable amounts of time and energy to pass a new protocol and produce new software. Software-defined networking techniques now allow for speed, innovation and collaboration around networking communities like academia, service providers and enterprises alike. Companies such as BigSwitch© that release software like Floodlight allow for experimentation and a faster adoption into the newfangled realm of SDN and network virtualization.

### References and Notes

1. “Openflow 1.0 Switch Specification”, December 31, 2009,  
<http://www.openflow.org/documents/openflow-spec-v1.0.0.pdf>
2. “Production Quality, Multilayer Open Virtual Switch”, January 28 2013:  
<http://www.openvswitch.org/>
3. “Floodlight Is An Open SDN Controller”, December 30 2011,  
<http://floodlight.openflowhub.org/>
4. “Guidelines and Suggested Standardized Traffic Classes”, 2012,  
<http://tools.ietf.org/html/rfc4594>
5. “Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers”, December 2013, <http://tools.ietf.org/html/rfc2474>
6. “QoS Rate-Limiting, Rate-limiting VM’s using QoS policing”,  
<http://openvswitch.org/support/config-cookbooks/qos-rate-limiting/>
7. “Floodlight Architecture”, December 2013,  
<http://www.openflowhub.org/display/floodlightcontroller/Architecture>
8. “5.2.3 Flow Match Structures”, January 27, 2013,  
<http://www.openflow.org/documents/openflow-spec-v1.0.0.pdf>
9. “OpenFlow Configuration and Management Protocol”, ONF, January 27, 2013,  
<https://www.opennetworking.org/standards/of-config>

10. “How to implement Quality of Service using Floodlight”, January 9, 2013,  
<http://www.openflowhub.org/display/floodlightcontroller/How+to+implement+Quality+Of+Service+using+Floodlight>

© 2013 by the authors; licensee Asia Pacific Advanced Network. This article is an open-access article distributed under the terms and conditions of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/3.0/>).