Configurable and Adaptive QoS Management via SDN

Yi Zhang
CS538 Advanced Computer Networks
Department of Computer Science
University of Illinois at Urbana-Champaign
yzhng173@illinois.edu

Qi Wang
CS538 Advanced Computer Networks
Department of Computer Science
University of Illinois at Urbana-Champaign
qiwang11@illinois.edu

Abstract

In small broadband access networks (e.g., home networks), the bandwidths are usually limited. In the meanwhile, different applications have different Quality of Service (QoS) requirements and they all compete for the scarce bandwidth. Unfortunately, many users are not savvy enough to configure QoS parameters of the underlying network to meet their needs. In this paper, we present *QoSManager*, a configurable and adaptive QoS management framework where users can specify QoS requirements for different applications at a high level. It monitors network and dynamically installs traffic shaping rules for application flows. In QoSManager, we propose a novel algorithm that takes QoS requirements and flows information as input and outputs an assignment of flows to queues of different rate to satisfy as many QoS requirements as possible. We implement QoSManager on Ryu and Open vSwitch (OVS) and demonstrate that it can provide QoS guarantee according to users' specifications in the presence of active competing traffic.

1. Introduction

In a home setting, different home members may request different services through the home network. For example, someone may play games, someone may watch videos, someone may use VoIP to make voice calls and someone may use the network to download files. However, the bandwidth of the network in a home setting is limited. Different applications compete for relatively scarce bandwidth resoruces. But the network users demand certain levels of Quality of Service (QoS) for different services. For example, the boy in the family who loves play games would demand high and conssitent bandwidth for the games. He would like to sacrifice the bandwidth of file downloading for his games. Users' demands of QoS also change based on different scenarios. For example, when the boy goes to school, the parents would like to demand high QoS for watching vidoes online. However, normal users don't have the knowledge to configure the underlying network to meet their needs.

In this project, we use SDN to manage the network and provide users the flexibility to configure their QoS requirements. Users do not need to have knowledge about the underlying network, they only need to configure the QoS requirements for the services they are using. Our SDN controller will dynamically assign appropriate bandwidth to different services to satisfy a maxinum number of QoS requirements accourding to the user configuration with the limited link capacity.

The rest of the paper proceeds as follows. Section 2 presents related work in QoS, as well as SDN-based solutions for home and broadband access networks. Section **??** describes the design of our system, as well as its implementation using Ryu and Open vSwitch. Section 4 evaluates our system in the context of competing flows. Section 5 discusses future work and concludes.

http://dx.doi.org/10.1145/2502323.2502331

2. Related work

2.1 Traditional QoS strategies

There are two types of strategies in traditional QoS: integrated services and differentiated services. Integrated services are fine-grained and per-flow.

- Every network element has to reserve resources for each flow.
- Router has limited computational resources; hard to classify all app flows.
- Not scalable.

Differentiated services [1] are more coarse-grained.

- Rely on the 8-bit DS field in the IP header. DiffServ routers then decide on per-hop basis how to forward packets based on their class.
- It is static (because of the predefined number of classes) and lacks the ability to fine-tune the QoS of separate flows.

2.2 SDN enabled QoS approach

2.2.1 Resource Reservation

FlowQoS [6] and EuQoS [7]

2.2.2 Per-flow Routing Frameworks

For Open-QoS [5], after classification, high priority flows are placed on QoS guaranteed routes.

2.2.3 Queue Management and Packet Scheduling

2.2.4 Policy Enforcement

2.3 QoS in home netowrks

Several other approaches explore QoS in home networks. Yiakoumis et al. proposed letting users notify the ISP about their bandwidth needs for a given application; in this case, provisioning occurs in the ISPs last mile, not in the home. Georgopoulos et al. proposed an OpenFlow-assisted framework that improves users quality of experience (QoE) in home networks for multimedia flows, subject to fairness constraints. The system allocates resources to each device but does not perform per-application or per-flow QoS. Mortier et al. developed Homework, a home networking platform that provides per-flow measurement and management capabilities for home networks. Homework allows users to monitor and control per-device and per-protocol usage, but it does not provide QoS support or perform any application classification.

3. Approach

In this section, we describe the design and implementation of our system. We first present the overview of our design and then describe the each component in detail.

3.1 Overview

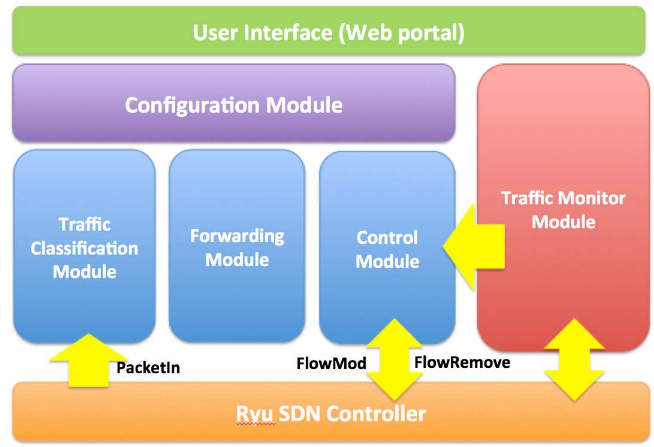

Figure 1: The architecture of our system.

Figure. 1 shows the high-level architecture of our system. Users could get the statistics of the network and configure QoS requirements for the services they are using via a Web portal. Users specify the mininum rate, recommended rate and priority they want for each application.

When the first packet of a flow arrives at the switch, a copy of this packet is forwarded to the controller. The switch continues to perform default forwarding of the traffic flows until application identification has been performed, essentially lazily implementing QoS only once application classification is complete. The controller determines which classifier should classify the application type for the flow. Each application type is associated with a different queue, each of which is shaped according to the traffic shaping policy for the application.

3.2 Configuration Module

In the configuration file, users only need to configure 3 parameters for a services. The following figure is part of a configuration file. The minimum rate parameter specifies the minimum bandwidth the service requires.The recommended rate parameter specifies the desired bandwidth for the service. The priority parameter specifies the users' prefenrece for the service. For example, in the figure, the minimum rate for video is

1.5M and the recommended rate for vidoe is 7M. The priority for video, which is 10, is much higher than other types of services. So our SDN controller will try to first satisfy the QoS requirements for the video services in the network.

User define configuration in YAML format

3.3 Flow classfier module

The flow classifier maintains a lookup table where the key is a flows five-tuple (e.g., source IP address, destination IP address, protocol, source port, and destination port). When a sender initiates a new flow, the switch sends a copy of the first packet of the flow to the controller. The flow classifier then checks whether the flows tuple correspond to an entry in this lookup table. The lookup then returns the type of application, such as video, VoIP, P2P, gaming, or web.

As traffic classification is not the focus of our project, in our implementation, we use static flow classifer to classfity the traffic.

It is very easy to extend our traffic classfication module to be more fine-grained and flexible. For example, we can easily integrate other approaches, such as DNS-based classifier [6], to our module. Or we can use the nmeta project [3] as our traffic classification module.

3.4 Traffic monitor module

- Flow statistics
- Port statistics

3.5 Control Module

On a high level, our control module tries to satisfy a maximum number of QoS requirements with the limited bandwidth. Each type of service has been configured by the user with a priority. We assign a weight to the service based on the actual bandwidth it will get. For example, the weight of a service is 0.6 if the minimum rate bandwidth is achieved. The weight is 1.0 if the recommended rate bandwidth is achieved. Our algorithm allocates bandwidth to different services to get the maximum value of

$$\sum_{i=1}^{n} Weight_i * Prioriy_i$$

under the condition that

$$\sum_{i=1}^{n} Bandwidth_i <= C$$

. C is the total bandwidth of the network.

3.5.1 Dynamic queue assignment algorithm

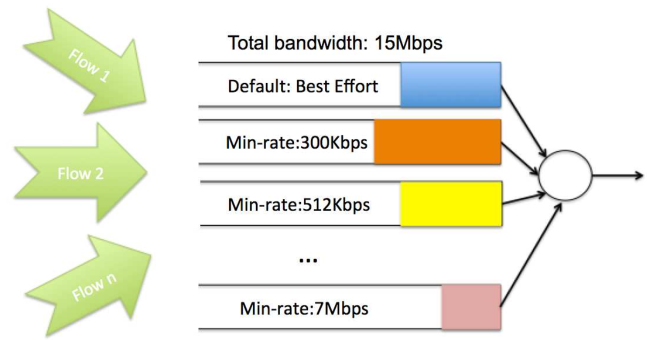

Figure 2

3.6 Web Portal Module

This module is for user configuration and traffic statistics.

3.7 Implementation

Implementation is based on Ryu SDN controller [4] and OpenVSwitch.

4. Evaluation

4.1 Evaluation setup

We used Mininet [2] in our experiments. We implemented the control application on top of Ryu [4], an open-source SDN controller.

Figure. 3 shows the experiment setup. We configuraed an Internext connection to be 15 Mbps.

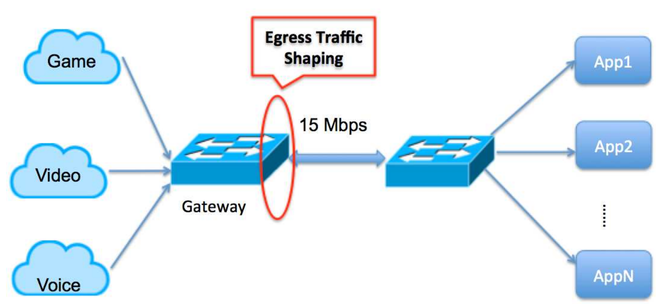

Figure 3

4.1.1 Scenario 1

Generate traffic for different services at the same time

4.1.2 Scenario 2

Generate traffic for one service first, then generate other traffic later

5. Conclusion and Future work

For the future work, we plan to add more features to our system

- Multiple path routing
- Time-based QoS
- Different device QoS

[2] Mininet: An instant virtual network on your laptop (or other pc). http://mininet.org/.

[3] Nmeta. http://mattjhayes.github.io/nmeta/.

[4] Ryu sdn framework. https://osrg.github.io/ryu/.

[5] H. E. Egilmez, S. T. Dane, K. T. Bagci, and A. M. Tekalp. OpenQoS: An OpenFlow controller design for multimedia delivery with end-to-end Quality of Service over Software-Defined Networks. In *Signal & Information Processing Association Annual Summit and Conference (APSIPA ASC), 2012 Asia-Pacific*, pages 1–8. IEEE, December 2012.

References

[1] Differentiated services. https://en.wikipedia.org/wiki/Differentiated_services.

[6] M. Said Seddiki, Muhammad Shahbaz, Sean Donovan, Sarthak Grover, Miseon Park, Nick Feamster, and Ye-Qiong Song. Flowqos: Qos for the rest of us. In *Proceedings of the Third Workshop on Hot Topics in Software Defined Networking*, HotSDN '14, pages 207–208, New York, NY, USA, 2014. ACM.

[7] Sachin Sharma, Dimitri Staessens, Didier Colle, D Palma, J Goncalves, R Figueiredo, D Morris, Mario Pickavet, and Piet Demeester. Implementing quality of service for the software defined networking enabled future internet. In *The European Workshop on Software Defined Networking, Proceedings*, pages 49–54. IEEE, 2014.