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

Bandwidth in a network, such as a home network, is limited. Users of such network demand certain levels of Quality of Service (QoS) for different services. However, in general, normal users do not have the knowledge to configure the underlying network to meet their needs.

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 consistent 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.

, .

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

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.

[2]

2. Related work

2.1 Traditional QoS strategies

2.2 SDN enabled QoS approach

3. Approach

3.1 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

3.2 Flow classfier module

Static flow classifier

3.3 Traffic monitor module

- Flow statistics
- Port statistics

3.4 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.

3.4.1 Dynamic queue assignment algorithm

3.5 Web Portal Module

This module is for user configuration and traffic statistics.

3.6 Implementation

Implementation is based on Ryu SDN controller [1] and OpenVSwitch.

4. Evaluation

4.1 Evaluation setup

Mininet

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

References

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

[2] Timothy Zhu, Alexey Tumanov, Michael A. Kozuch, Mor Harchol-Balter, and Gregory R. Ganger. Prioritymeister: Tail latency qos for shared networked storage. In *Proceedings of the ACM Symposium on Cloud Computing*, SOCC '14, pages 29:1–29:14, New York, NY, USA, 2014. ACM.