

# 基于 SDN 的网络带宽保障系统<sup>①</sup>

肖军弼, 隋萌萌, 李 芑

(中国石油大学(华东)计算机与通信工程学院, 青岛 266580)

**摘 要:** 传统的数据中心网络难以为指定业务的网络带宽需求提供保障。基于 SDN (Software Defined Network) 的网络带宽保障系统通过在 Floodlight 控制器中开发网络带宽保障服务, 并采用 MVC (Model View Controller) 模式设计出一个客户端软件, 能够实现用户对网络状态的实时监控和对带宽保障策略的灵活调度。在 Mininet 搭建的虚拟网络环境中进行测试的结果显示, 该系统能够实时监控网络状态、灵活调度带宽保障策略, 进而准确有效地保障网络中指定业务的带宽需求。

**关键词:** SDN; Floodlight; OpenFlow; 带宽调度

## Network Bandwidth Guarantee System Based on SDN

XIAO Jun-Bi, SUI Meng-Meng, LI Peng

(College of Computer and Communication Engineering, China University of Petroleum (East China), Qingdao 266580, China)

**Abstract:** Traditional data center network has difficulty in guaranteeing the network bandwidth requirements of specified services. By developing network bandwidth guarantee services in Floodlight controller and designing client software using MVC (Model View Controller) pattern, the network bandwidth guarantee system based on SDN (Software Defined Network) is designed to enable users to monitor the network status real-timely and schedule the bandwidth guarantee strategies flexibly. Through the tests in the virtual network environment which is created by Mininet, the results show that this system can monitor the network status real-timely, schedule bandwidth guarantee strategies flexibly and then guarantee the bandwidth requirements for specified services in the network accurately and effectively.

**Key words:** SDN; Floodlight; OpenFlow; bandwidth scheduling

## 1 引言

互联网不断发展壮大, 需要服务的业务类型日益增多, 而传统网络的复杂结构和管控难度已难以满足各类新型业务的需求<sup>[1]</sup>。就传统的数据中心网络而言, 一般只尽力转发, 并不为某一业务所需的额外带宽提供保障。当用户对某一特定业务有较高的带宽需求时, 其需求若不能得到较好的保证, 将会影响相关服务的正常运转。

软件定义网络(Software Defined Network, SDN)作为一种新型网络架构, 打破了传统网络设计理念<sup>[2]</sup>。SDN 强大的集中管控能力, 可以实现对网络中多种类型流量的灵活调度, 实时保障特定业务的带宽需求, 从而有效降低网络中不必要的冗余流量, 实现人性

化、个性定制的用户体验。

现有的针对 SDN 带宽保障方面的研究仍局限于传统网络思维方式。通过各种方式标记优先级, 在网络拥塞时优先保障高优先级的业务流量通过, 而禁止低优先级的业务流量<sup>[3,4]</sup>, 没有充分发挥 SDN 架构的特性和 OpenFlow 协议的优势, 且当业务优先级发生变化时, 在应对需求变化方面也不够灵活和及时有效。

本文在 SDN 架构之上研究带宽调度, 通过开发带宽保障系统达到实时保障服务质量的目的。在设计和实现该系统时, 需要对 OpenFlow 协议有深入的理解, 从而开发出恰当的端口队列配置工具。此外, 还需要对 Floodlight 开源控制器的主要模块和底层开发技术有全面充分的掌握, 从而实现在控制器中开发带宽保

<sup>①</sup> 基金项目: 赛尔网络下一代互联网技术创新项目(NGII20150115)

收稿时间: 2015-09-25; 收到修改稿时间: 2015-11-13 [doi:10.15888/j.cnki.csa.005221]

障服务模块, 并利用 SDN 控制器上的相关接口实现调度策略的转换和相应流表的下发。

## 2 系统概要设计

本系统主要由前台的客户端程序和后台的 SDN 控制器两部分组成, 系统整体架构如图 1 所示。用户通过客户端程序可以查看控制器的状态、全局网络拓扑等信息, 并可启动带宽保障服务和添加带宽保障策略等。SDN 控制器采用 Floodlight 开源控制器, 南向接口基于 OpenFlow v1.3 协议<sup>[5]</sup>。通过在 Floodlight 控制器中实现网络带宽保障服务, 对整个网络中的带宽进行调度, 保障指定业务的带宽。Floodlight 控制器中的带宽保障服务通过 REST API 提供相应的查询和配置功能。

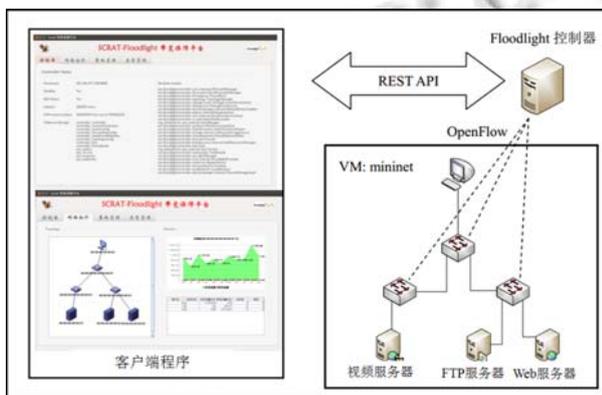


图 1 系统整体架构

### 2.1 网络带宽保障服务

Floodlight 控制器中添加的网络带宽保障服务主要用于在控制器中实现对网络带宽的调度, 从而保障指定业务的网络带宽。进一步进行模块划分, 主要包含的功能模块如图 2 所示。

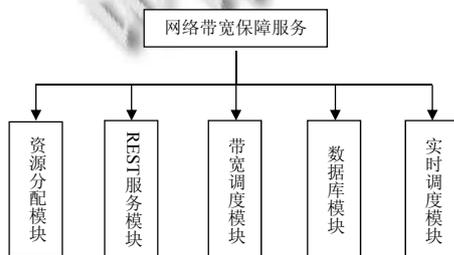


图 2 网络带宽保障服务功能模块

资源分配模块用于为网络中的交换机配置相应的队列, 以实现网络带宽的限制与预留。利用本模块可

以给指定的 Open vSwitch 交换机<sup>[6]</sup>的指定端口添加一定数量的队列, 并为每个队列采用相应的 QoS 保障算法, 以保障队列的最大与最小传输速率。

REST 服务模块是网络带宽保障服务的 REST API 提供模块, 用户可通过 HTTP 请求调用相应的 REST API, 以获得控制器中网络带宽保障服务提供的相应服务, 如启动和关闭带宽保障模块、添加和删除策略等。

带宽调度模块是本系统最主要的模块, 主要负责网络中流量的调度, 将用户添加的调度策略转化为相应的流表下发到网络中, 根据策略中指定的不同调度方式, 添加指定方式的流表处理网络流量, 从而限制一般网络业务的流量, 保障指定网络业务的网络带宽。在本模块中主要设计了三种策略调度方式, 分别是队列设置、计量添加、服务。队列设置方式是利用 OpenFlow 交换机中的端口队列特性<sup>[7]</sup>, 通过添加采用 Set Queue 动作的流表, 将网络中不同业务的流量调度到不同的队列中, 从而利用队列的最大最小速率来限制和保留指定业务的网络带宽; 计量添加方式是利用 OpenFlow 协议中的 meter 特性, 通过添加采取 meter 指令的流表, 将网络中不同业务的数据流量调度到不同的 meter 表中, 从而根据不同的 meter band 来限制一般网络业务的流量, 同时保障指定业务的网络带宽; 服务方式是可以根据 IP 数据包中的 TOS 字段, 设计一个简单的 DiffServ 模型, 通过添加 Set TOS 的动作的流表, 将指定业务流的 TOS 字段设置为一定的 DSCP 值, 从而在网络中通过控制器的 DiffServ 模块来实现对指定 DSCP 值提供服务质量的保障。

数据库模块是存储用户添加的策略、服务、队列设置和计量添加等信息的库, 还用于实时调度模块从中查找相应策略, 以便实时保障指定业务的网络带宽。

实时调度模块主要用于当网络中出现拥塞时, 实时地调度数据库中的策略, 对网络中的流量进行实时调度, 以保障指定网络业务的带宽。

### 2.2 客户端

客户端主要负责人机交互, 为用户提供一个简洁的、友好的界面。用户可以通过该界面查看控制器中的相应状态、全局网络信息, 并能够开启和关闭带宽保障服务、添加带宽保障策略等。客户端包含的主要功能模块如图 3 所示。

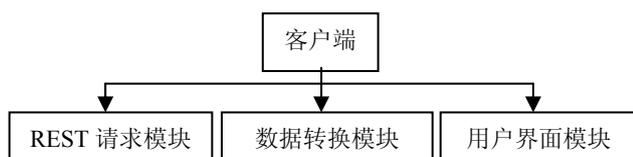


图3 客户端功能模块

REST 请求模块主要通过 HTTP 请求调用控制器北向接口的 REST API, 向控制器发送相应的 REST 请求, 以获得相应的控制器信息, 如控制器状态信息、全局网络信息、数据库中的策略信息等等, 还可以向控制器中添加策略、服务等。

数据转换模块是将 REST 请求模块得到的 JSON 格式的数据转化为相应的 Java 数据结构, 如交换机类、策略类等, 便于用户在程序中使用。该模块还会将用户通过界面添加的策略等信息转化为相应的 JSON 格式的数据, 以便于 REST 请求模块发送相应请求。

用户界面模块主要是负责与用户进行人机交互。通过设计出一个简单的、友好的界面, 在界面上显示相应的控制器、全局网络和策略服务等信息, 并能够通过该界面向控制器添加策略和服务。

### 3 系统详细设计与实现

#### 3.1 系统环境搭建

搭建系统环境是开发网络带宽保障系统中至关重要的一步。环境搭建过程如图4所示。

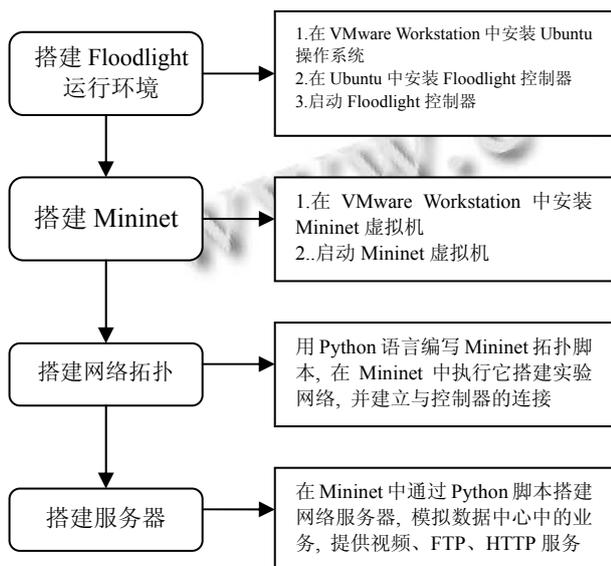


图4 系统环境搭建

#### 3.2 网络带宽保障服务

通过向 Floodlight 控制器中添加一个网络带宽保障服务模块, 来提供保障指定业务网络带宽的功能。该模块需要实现 IFloodlightModule 接口, 以成为 Floodlight 控制器的模块, 并实现 IOFMessageListener 接口, 以监听发送到控制器的 PACKET\_IN 消息<sup>[8]</sup>。

资源分配模块为 Open vSwitch 交换机提供端口队列配置的功能。该模块是用 Python 语言实现的 Linux 终端命令行交互程序, 是一个端口队列配置工具。

数据库模块是存储策略、服务、配置队列和添加的 meter 等信息的库, 它是利用 Floodlight 中的 MemoryStorageSource 模块实现的, 是内存中的 NoSQL 存储, 在数据库中存储了策略表、服务表、队列配置表、meter 添加表等数据表信息。

带宽调度模块是网络带宽保障服务模块的核心, 它能够对网络中的流量实现调度, 以保障指定业务流的带宽。该模块根据用户添加的策略中相应的业务信息、交换机信息和流量调度方式等, 生成一定的流表, 并写入到指定的交换机中, 来调度通过交换机的流量。

在本模块中设计的策略主要由七部分组成, 其结构如表1所示。

表1 策略(Policy)结构

id	name	switch	priority	policyType	match	enable
----	------	--------	----------	------------	-------	--------

其中策略的匹配项主要有 11 个, 其结构如表2所示。

表2 match 结构

inp ort	eth src	eth dst	eth type	vla nid	lps rc	ip dst	prot ocol	tos	srcc port	dstp port
------------	------------	------------	-------------	------------	-----------	-----------	--------------	-----	--------------	--------------

策略的类型主要有三种, 即队列设置、计量添加、服务。下面针对这三种方法分别进行讨论。

对于利用端口队列特性的方法, 是将指定业务流调度到特定的队列, 通过该队列来保障该业务流的带宽。调度业务流到指定的队列, 是利用 OpenFlow 中的流表, 其中流表的匹配项为想要保障的业务标识, 流表的动作是 Set Queue。这种策略类型的结构如表3所示。

表3 队列设置的结构

id	name	port	queue
----	------	------	-------

对于利用 meter 特性的方法, 通过向网络中下发流表调度一般的业务流到特定的 meter 表中, 其中的流表匹配项为除要保障业务以外的其他业务的标识, 流表的指令是 meter id。这种策略类型的结构如表4所示。

表 4 计量添加的结构

id	name	type	rate	burst	maxrate
----	------	------	------	-------	---------

对于根据 IP 中的 TOS 字段设计的简单 DiffServ 模型的方法, 首先是将指定的业务流中的 TOS 字段修改为一定的 DSCP 值, 然后再在 Floodlight 控制器中的 DiffServ 模块的作用下, 将指定的业务流进行带宽的保障. 这种方式的策略类型的结构如表 5 所示.

表 5 服务的结构

id	name	tos
----	------	-----

该方法中使用的 DiffServ 模块, 是根据传统的网络中的 DiffServ 模型设计的, 主要包括流分类和入队两部分组成. 流分类是根据该模块收到的 PACKET\_IN 中的 TOS 字段来进行的分类, 其中的分类方法同传统的 IP 网络. 入队是根据流分类得到类别, 将相应的数据流调度到一定的队列, 进而限制业务流的带宽.

实时调度模块是当网络中发生拥塞时, 对网络流量进行调度. 在该模块的实现中, 通过调用 Floodlight 中 ThreadPoolService 服务向控制器的线程池中添加一个定时启动的线程, 该线程定时的发送 OpenFlow 端口状态请求, 查看网络中的端口流量情况, 当某个端口中的流量达到设定的阈值时, 相应的策略表中的策略会转化为流表下发到网络中, 实现网络流量的调度.

REST 服务模块利用 Java 中的 Restlet 包, 将网络带宽保障服务的相应功能通过 REST API 的形式提供出去. 在实现该模块时, 首先编写网络带宽保障服务的相应功能的 Resource, 比如策略添加、删除及显示的 Resource 等, 然后通过相应的实现 RestletRoutable 接口的类, 将相应的 Router 添加到 Floodlight 控制器的 REST 服务器中.

### 3.3 客户端

客户端是本系统中与用户进行交互的部分, 该部分在实现时, 采用的是 MVC 的开发模式. 客户端中主要包括 REST 请求模块、数据转换模块和用户界面模块三大主要功能.

REST 请求模块主要负责客户端向 Floodlight 控制器发送 REST 请求, 并得到控制器响应的 JSON 格式数据. 本模块实现时, 在线程中通过调用 HttpClient 包提供的 HTTP 请求, 将一定的 JSON 格式的数据发送给控制器, 并获取 JSON 格式的数据回复.

数据转换模块是将 REST 请求模块获得回复的

JSON 格式的数据转换为模型中指定类的对象, 或将模型中相应的类的对象转换为 JSON 格式的数据, 以便 REST 请求模块使用. 在该模块的实现中, 以 Jackson 包为主, 以 org.json 包为辅处理 JSON 数据.

用户界面模块是提供与用户之间进行交互的界面, 即 MVC 模式中的视图. 在开发该模块时, 使用的是 Java 中的 SWT 界面开发包, 该界面开发包能够开发出跨平台的用户界面, 实现程序的可移植性.

## 4 系统测试

由于本系统包括 Floodlight 控制器中的网络带宽保障服务和客户端两部分, 系统的结构和功能比较复杂, 测试的过程也比较复杂, 因此在该测试过程部分, 只针对队列配置类型的策略方法进行主要的阐述. 在系统测试中, 数据中心网络是通过 Mininet<sup>[9]</sup>网络仿真工具搭建的, 具体的网络拓扑如图 5 所示.

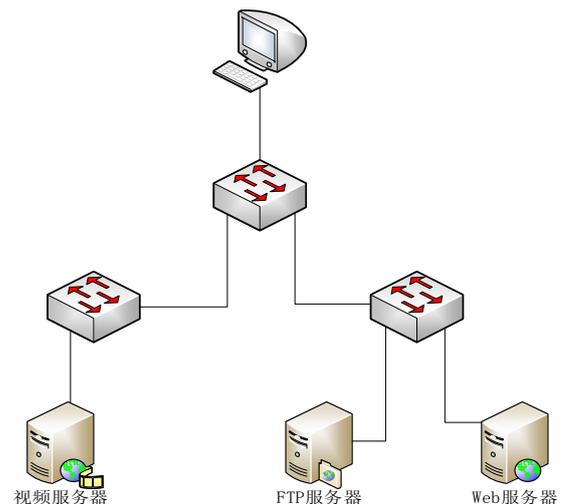


图 5 系统测试网络拓扑图

实验网络中的数据中心提供三种不同的网络业务, 在测试过程中, 主要为视频业务提供网络带宽的保障. 在该虚拟网络中的链路带宽大约为 10Gbps, 而视频的播放码率仅为 17Mbps. 为了能使网络中传输的视频业务受到影响以测试本系统, 测试过程中需要利用资源分配模块开发的队列配置工具, 对网络中的相应的交换机端口的默认队列进行配置, 该测试中配置端口速率为 30Mbps.

测试过程中, 用户使用 PC 访问视频服务器提供的视频服务, 由于当前网络中仅有视频业务在传输, 因而

用户能够获得正常的视频服务. 此时, 用户再通过 PC 访问 FTP 服务器提供的 FTP 服务, 从 FTP 服务器上下载一个较大的文件, 这时网络中就存在这两种网络业务, 而且由于下载的文件较大, FTP 业务占用的网络带宽就比较大, 视频业务的传输就受到了影响, 如图 6 所示. 然后用户可以通过客户端向控制器中下发一定的队列配置策略, 保障视频业务的带宽. 队列配置类型的策略添加到控制器后, 查看 PC 中播放的视频, 发现视频又能正常播放, 如图 7 所示. 在测试中, 用户通过客户端向控制器下发一定的带宽调度的策略, 控制器会将相应的策略转化为一定的调度流量的流表, 调度网络中不同业务的流量, 从而实现了保障指定业务的网络带宽.

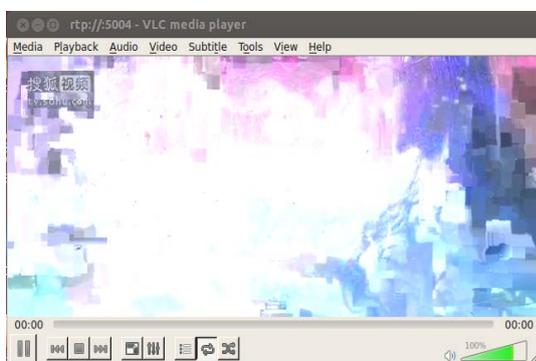


图 6 受干扰的视频



图 7 启动网络带宽保障服务后的视频

在队列配置类型策略下发之前, 网络中交换机 S1 的 s1-eth3 端口的队列流量信息如图 8 所示, 可以发现此时的网络流量皆通过默认队列 queue0 传输. 向控制器中下发队列配置类型策略保障视频业务带宽后, 交换机 S1 的 s1-eth3 端口的队列流量信息如图 9 所示, 此时可以发现网络中流量有一部分被调度到 queue2 中传输, 再查看交换机 S1 中的流表, 可以发现有一条用户添加的队列配置类型的策略转化的流表. 通过以上的分析, 可以说明本系统能实现对指定业务的网络带宽进行保障.

```

OFPOST_QUEUE reply (OF1.3) (xid=0x2): 3 queues
port 3 queue 0: bytes=718, pkts=8, errors=0, duration=15195.945s
port 3 queue 1: bytes=0, pkts=0, errors=0, duration=15195.945s
port 3 queue 2: bytes=0, pkts=0, errors=0, duration=15195.945s
mininet@mininet-vm:~/mininet/custom$

```

图 8 交换机 S1 的 s1-eth3 端口的队列信息

```

OFPOST_QUEUE reply (OF1.3) (xid=0x2): 3 queues
port 3 queue 0: bytes=671252905, pkts=447027, errors=41994, duration=13141.486s
port 3 queue 1: bytes=0, pkts=0, errors=0, duration=13141.486s
port 3 queue 2: bytes=1942864, pkts=1420, errors=0, duration=13141.486s

```

图 9 策略下发后交换机 S1 的 s1-eth3 端口的队列信息

## 5 总结

本文以流量调度为主要研究内容设计了基于 SDN 的网络带宽保障系统. 系统基于 SDN 架构和 OpenFlow 协议的特性提出了三种带宽调度方式, 使得用户能够实时监控网络运行状态并在必要时灵活下发调度策略, 保障指定业务的带宽需求, 在网络带宽合理分配方面取得了较为理想的效果. 较之其他流量转发和带宽分配方式, 本文提出的基于 SDN 的带宽调度更为灵活多变, 能够适应不同的网络环境和业务需求. 由于 Open vSwitch 交换机目前没有实现计量表, 无法对计量添加类型的策略进行验证; 当前 DiffServ 模块的设计比较简单. 在后续工作中将对这两种调度策略进一步予以完善, 以增强系统的流量调度和带宽保障功能.

## 参考文献

- 1 左青云, 陈鸣, 赵广松, 邢长友, 张国敏, 蒋培成. 基于 OpenFlow 的 SDN 技术研究. 软件学报, 2013, 24(5): 1078-1097.
- 2 张朝昆, 崔勇, 唐嵩祯, 吴建平. 软件定义网络(SDN)研究进展. 软件学报, 2015, 26(1): 62-81.
- 3 李鹤飞, 董晨, 郑晓航, 褚灵伟, 陆肖元. 基于软件定义网络的流量管理应用的研究和实现. 计算机应用于软件, 2015, 32(5): 17-19.
- 4 赵笑楠. 基于 SDN 的网络流量控制模型设计与研究[硕士学位论文]. 北京: 北京交通大学, 2015.
- 5 雷葆华, 王峰, 王茜, 王和宇等. SDN 核心技术剖析和实战指南. 北京: 电子工业出版社, 2013.
- 6 Open vSwitch Overview of functionality and components. <https://github.com/openvswitch/ovs/blob/master/README.md>.
- 7 OpenFlow Switch Specification Version 1.3.0(Wire Protocol 10x04). <http://www.sdnip.com/wp-content/uploads/openflow/openflow-spec-v1.3.0.pdf>.
- 8 Floodlight Documentation. <http://www.projectfloodlight.org/documentation/>.
- 9 Mininet Walkthrough. <http://mininet.org/walkthrough/>.