一种基于SDN的QoS应用编程接口设计方案

付 健,沈苏彬
(南京邮电大学 计算机学院,江苏 南京 210003)

摘 要:随着互联网技术及其应用的发展,SDN(Software Defined Networking,软件定义联网)技术越来越受到关注。为满足复杂多样的网络应用功能需求,SDN控制器通过应用编程接口,为应用层提供灵活的网络可编程性。已经实现的控制器应用编程接口,大多只提供了网络基本功能,难以满足实际应用需求。针对视频会议、在线视频等具有服务质量保证要求的SDN网络应用需求,设计具有服务质量参数的应用编程接口,并实现了REST(REpresentational State Transfer,表述性状态转移)风格的控制器应用编程接口。通过QoS客户端测试应用接口具有支持服务质量保证的能力,结果表明,文中设计的应用接口具有良好的灵活性、可扩展性,能够对具有服务质量需求的SDN网络应用提供支持。

关键词:软件定义联网;OpenFlow;应用编程接口;服务质量保证
中图分类号:TP393 **文献标识码**:A **文章编号**:1673-629X(2015)11-0099-06
doi:10.3969/j.issn.1673-629X.2015.11.020

A Design Scheme of Application Programmable Interface for QoS Based on SDN

FU Jian, SHEN Su-bin
(School of Computer, Nanjing University of Posts and Telecommunications, Nanjing 210003, China)

Abstract: With the development of Internet techniques and application, SDN (Software Defined Networking) technology takes more and more attention. To meet the complex and diverse needs of network applications, SDN controller provides network programmability for application via the application programming interface. However, most of the programmable interface of SDN controller only provides the underlying functionality. It does not satisfy the requirements of practical applications. For video conference, streaming media and other use cases that require the guaranteed quality of service, the application programming interface with a set of QoS parameters are designed and implemented by REST. Finally, design a QoS client to test performance of the application interface and test results show that the application interface has good flexibility and scalability, which provides support for QoS applications.

Key words: SDN; OpenFlow; API; QoS

0 引 言

随着网络技术的发展,SDN[1](Software Defined Networking,软件定义联网)技术越来越受到关注。SDN是一种新兴的基于软件实现的网络架构和技术,其最大优势在于具有松耦合的控制平面和数据平面、支持集中化的网络状态控制、灵活的网络编程能力以及底层网络基础资源设施对上层的透明化[2]。虽然SDN网络已经具备大部分的网络基础功能,但还缺少许多基本的网络服务功能,如服务质量保证。

目前,业界广泛认可ONF[3](Open Network Foundation,开放网络基金会)定义的三层SDN网络架构,包括应用层、控制层和数据层。该架构以控制层面为中心,负责集中处理数据层面资源,部署交换机转发规则,维护网络拓扑等。软件可编程能力是SDN网络的一大特性,控制器向应用层提供开放的可编程接口,使应用层开发者可以通过软件编程的方式调度网络基础资源。业界对于控制器面向应用层接口的标准化方面还存在较大争议,原因在于应用层接口直接为上层应用服务,其设计势必与应用层业务密切相关,应用层业务的多样性以及需求的复杂性决定了应用层接口设计标准难以统一。ONF已经成立专门的研究小组制定应用层的开放接口标准,但统一的应用层接口标准在

收稿日期:2015-02-04 **修回日期**:2015-05-05 **网络出版时间**:2015-11-04
基金项目:江苏省未来网络前瞻性研究资助项目(BY2013095-1-08)
作者简介:付 健(1989-),男,硕士研究生,研究方向为计算机网络;沈苏彬,博士生导师,研究方向为计算网络、下一代电信网以及网络安全。
网络出版地址:http://www.cnki.net/kcms/detail/61.1450.TP.20151104.0948.016.html

短时间内难以实现。NOX[4] 等开源主流控制器也仅仅提供了控制器基础功能的应用接口,缺少针对具体应用的可编程接口。

针对以上问题,文中设计了具有服务质量保证的控制器可编程接口方法,并提出了可行的实现方案。SDN 网络中至今并未提出服务质量保证的完整性解决方案。文中首先基于 OpenFlow 交换机流表、队列、计量表等服务质量保证技术,采用 REST 风格网络服务实现技术,设计并实现了具有服务质量保证的控制器应用可编程接口。其次,按照区分服务的思想,根据数据分流、策略制定、入队和调度的处理流程,提出了基于可编程接口的 QoS 服务方案,提高了应用的服务质量。

1 相关技术分析

1.1 OpenFlow 协议

OpenFlow 协议[5] 定义了 SDN 控制器与交换机之间的接口标准,以及控制器与交换机之间的通信标准。

通过对各版本协议规范中定义的 SDN 网络构件和网络操作进行分析,OpenFlow 协议各版本中均有 QoS 服务的支持的描述。SDN 网络中的 QoS 服务支撑技术如表 1 所示。

表 1 SDN 服务质量保证支撑技术

QoS 能力	交换机构件	关键消息类型
流量控制	队列	Queue Configuration Messages
区分服务	计量表队列	Meter Modification Messages
MPLS	动作表	Push-Tag/Pop-Tag,Push-MPLS
路由控制	流表	Flow Table Modification Messages

(1)流量控制。

交换机端口可以配置一个或者多个队列;分组经流表处理,执行 enqueue 动作,可将分组加入指定队列。端口队列具有最大和最小速率限制,映射到队列的分组转发速率得到相应的控制。

(2)区分服务。

OpenFlow 1.3 版本[6] 为交换机添加计量表,计量表不仅能够测试分组的转发速率,并且能够控制分组的转发速率。计量表与端口队列配合使用,可以实现复杂的 QoS 框架,如区分服务模型。

(3)MPLS(Multi-Protocol Label Switching,多协议标签交换)。

MPLS 能够通过快速转发的方式提高 IP 分组流传递的性能,提高实时服务的支持能力。

(4)路由控制。

流表是 SDN 网络中数据层唯一的转发规则,而流表由控制器制定并集中管理。路由的制定依赖于数据层转发规则。因此,控制器是 SDN 网络中路由控制的核心。

文中利用交换机流表、队列等 QoS 资源,在应用接口中为应用层提供 QoS 资源控制能力,使网络应用能够灵活调用服务质量保证资源。

1.2 SDN 可编程接口技术

目前,主流控制器中应用可编程接口的实现方式主要有以下几种。Nicira 团队开发的第一个开源 SDN 控制器 NOX[5] 及其后续版本 POX 利用 Python 定义应用可编程接口;Big switch 团队开发的 FloodLight[7] 控制器和最新的 OpenDayLight[8] 控制器提供标准的 REST 风格的北向接口。REST API 以其灵活易用性在 SDN 接口设计中得到了广泛的应用。文中将采用 REST 风格 Web 服务的实现方式,将控制器资源进行划分,确定资源的 URI,设计资源的操作方法,最后实现具有服务质量保证的应用可编程接口。

1.3 QoS 技术

QoS[9](Quality of Service,服务质量)指一个网络能够利用各种基础技术,为指定的网络通信提供更好的服务能力,是网络的一种安全机制,是用来解决网络延迟和阻塞等问题的一种技术。尽管 IETF 已经提出集成服务、区分服务、资源预留协议等模型框架用于解决传统网络 QoS 问题,但传统网络尽最大努力的交付以及逐跳计算路由分布式的网络管理方式缺乏对整个网络资源的全局管理,上述 QoS 模型均没有得到广泛的部署和使用。

OpenFlow 协议提出之后,其控制与转发层路由松耦合的特点,使 SDN 网络中实现集中式的 QoS 管理系统成为可能。在文献[10-11]中,Egilmez H E 等通过分析传统的区分服务、集成服务模型的弊端,提出考虑 QoS 服务质量的动态路由规划算法,以 QoS 性能指标作为基本的参数和约束条件,动态制定路由。这样分组将会按照满足其服务质量要求的路径传输。Raumer D 在 2014 年提出了在 SDN 网络中实现 QoS 支持的监测器 MonSamp[12]。上述研究均在假定的实验场景中,设计一整套的 SDN 服务质量管理框架,缺乏灵活性和可移植性。文中按照区分服务的思想,根据控制器应用接口提供的流表、队列操作功能,实现分流、策略、队列和调度的处理流程,提高 SDN 网络的服务保证能力,具有良好的灵活性和可扩展性。

2 设计目标

文中在 SDN 集中式网络环境下,利用 OpenFlow 交换机提供的 QoS 支撑技术,结合控制器应用可编程接口技术,采用 REST 技术设计并实现具有服务质量保证的应用可编程接口。本节首先阐述了具有服务质

量保证的应用可编程接口的功能需求;其次说明了应用接口所要解决的关键问题。

具有服务质量保证的控制器应用接口需向应用层提供以下功能[13]:

(1)为网络应用提供实时、有效、正确的网络资源使用状况,使应用制定正确的 QoS 策略。

(2)为网络应用和控制器之间提供可靠的通信机制,保证网络应用制定的 QoS 策略能够可靠地传递到控制器。

(3)为网络应用提供灵活的 QoS 资源操控能力,如交换机流表、队列、计量表等资源。QoS 资源是服务质量保证最终的实现部署者,从应用的角度,控制器需向应用提供资源控制能力。

(4)为网络应用的可扩展性提供支持。应用接口可扩展性,使网络应用新功能的部署更加便捷。

服务质量保证的基本要求是根据网络状态制定 QoS 策略。在策略制定之前,需首先获得网络目前资源状况,而控制器通过拓扑管理功能,集中掌控网络中的资源使用情况。因此,应用接口首先应具备拓扑管理 API。其次,应用需要对交换机流表、队列、计量表等资源进行操控,并实时获取这些资源的状态,此外应用接口还需提供 QoS 资源控制 API。最后,应用接口必须提供策略解析 API,对网络应用制定的 QoS 策略进行解析,并准确下达给控制器。

综上所述,服务质量保证应用接口必须解决拓扑管理、QoS 资源控制、QoS 策略解析三个基本问题。

3　应用接口设计

文中利用 REST 网络服务实现技术,设计具有服务质量保证的应用接口。

REST 风格的网络服务是一种强调面向网络资源的网络服务实现框架。REST 利用 HTTP 最基本的 GET/PUT/POST/DELETE 作为服务的基本动作。REST 将所有的网络构件抽象为网络资源,并设置统一资源标识(Uniform Resource Identifier,URI)。通过 HTTP 服务的基本动作实现对网络资源的查询、添加和修改、删除操作。

利用 REST,可以解决文中提出的几个关键问题。如图 1 所示,使用 REST API 提供的四种操作,可将应用制定的 QoS 策略部署到 OpenFlow 交换机。

图 1 描述了通过调用 REST 接口资源,下发 QoS 策略的实例。实例中网络应用制定两条 QoS 策略,分别为 Policy1 和 Policy2。Policy1 表示将流 Flow A 加入队列 Queue1;Policy2 表示将流 Flow B 加入队列 Queue0。Queue0 和 Queue1 分别代表不同的带宽速率限制。调用接口中的 Policy 资源,使用 POST 方法,将两条策略下达给控制器。调用接口中的 Queue 资源,使用 GET 方法获取节点上队列的信息,查询 Queue0 和 Queue1 的 ID。最后调用接口中的 Flow 资源,使用 POST 方法,配置流表项,设置数据流的入队操作。

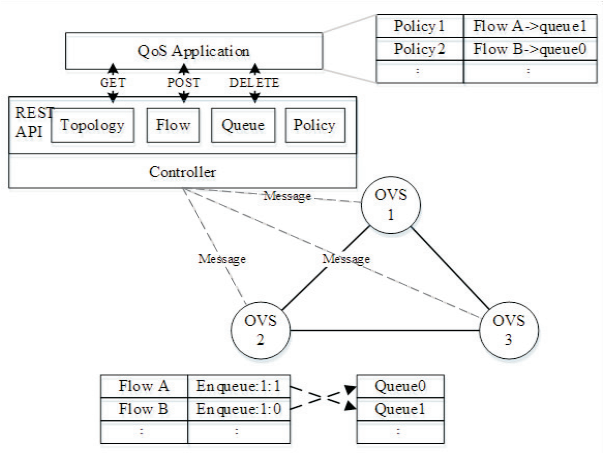


图 1　QoS 应用可编程接口

由于 REST 是基于资源的网络服务,文中根据第 2 节提出的 QoS 应用接口几类关键问题,将问题相关的控制器服务抽象为资源。QoS 应用可编程接口需要操作的网络资源包括:

(1)拓扑管理资源;

(2)QoS 资源(包括交换机流表、队列、计量表);

(3)策略解析资源。

3.1　资源 URI 定义

根据 REST 的技术规范,所有的网络服务被统一抽象为资源,并赋予统一的 URI 表示。文中按照分层结构,为服务质量相关的三类资源定义 URI,如图 2 所示。

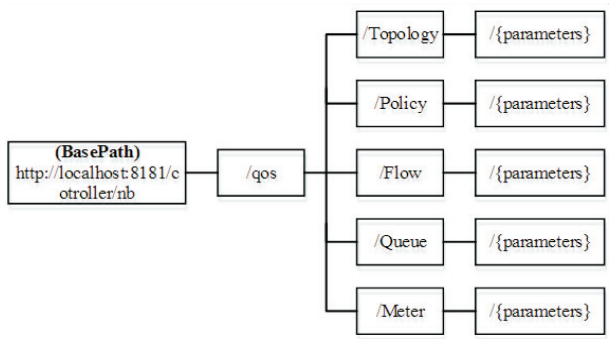


图 2　URI 分层结构

图 2 中对 QoS 相关资源进行定义,如 QoS 策略资源的 URI 为:

http://locaihost:　8181/controller/nb/qos/Policy/{parameters}

{parameters}表示操作的参数。

3.2　资源操作定义

根据 REST 的实现标准,利用 HTTP 协议的 GET/POST/DELETE 方法,实现资源的查询、添加和修改、删

除操作。文中为应用接口中的各类资源提供的操作方法如表 2 所示。

表 2 QoS 接口操作定义

资源类型	客户端请求
拓扑	GET:返回指定节点上端口的 QoS 相关信息
策略	GET:获取节点上添加的 QoS 策略 POST:增加或修改节点上的 QoS 策略 DELETE:删除节点上的 QoS 策略
流表	GET:返回指定节点上流表中流表项信息 POST:增加或修改节点上流表中的流表项 DELETE:删除节点上流表中的流表项
队列	GET:返回指定节点上端口的队列相关信息
计量表	GET:返回指定节点上端口的计量表信息 POST:增加或者修改节点上计量表的配置 DELETE:删除节点上匹配的计量表的配置

4 应用接口实现

4.1 应用接口 JAVA 类实现

编写 QoSNorthbound 类,作为接口实现的基础类,实现接口的控制逻辑。分别编写 TopologyManager、PolicyManager、FlowManager、QueueManager、MeterManager 类实现 QoS 资源的管理。资源管理类中分别实现与资源操作相对应的 get( )、post( )、delete( ) 方法,实现对资源的具体操作逻辑。

4.2 URI 实现

文中使用 JAX-RS 技术实现应用接口资源与 URI 的关联。JAX-RS 使用了 Java SE5 引入的 Java 标注,简化了 Web 服务的客户端和服务端的开发和部署。JAX-RS 提供了一些标注,将一个资源类封装为 Web 资源。JAX-RS 提供的标注类型如表 3 所示。

表 3 JAX-RS 标注类型

标注类型	标注功能
@ Path	标注资源类或者方法的相对路径
@ GET @ PUT @ POS @ DELETE	标注方法对应 HTTP 请求的类型
@ Produces	标注返回的 MIME 媒体类型
@ Consumes	标注可接受请求的 MIME 媒体类型
@ PathParam	标注来自 URL 路径的参数
@ QueryParam	标注来自数据段的参数
@ HeaderParam	标注来自 HTTP 请求的头部信息

PolicyManager 类中 post( ) 方法与 URI 关联实现如下。

```
@Path("/qos/Policy/")
@POST
@Produces({MediaType.APPLICATION_JSON})
@TypeHint(PolicyConfig.class)
@StatusCodes({@ResponseCode(code = 200, condition = "
Operation successful")
……//其他响应状态})
```

@ StatusCodes 用于说明响应状态码和响应信息。

@ Produces 指明了该应用接口采用 JSON 格式的数据类型。

4.3 JSON 数据

基于 REST 风格的 Web 服务常用的数据传输类型有 JSON 和 XML 类型。

XML 格式统一,容易与其他系统进行远程交互,数据共享比较方便,但 XML 文件比较庞大,传输占带宽比较大,服务器端和客户端需要花费大量的时间解析 XML。JSON 是一种轻量级的数据交换格式,具有良好的可读性和快速编写的特性。由于 JSON 数据格式都是压缩的,所以对于带宽需求较小,并且 JSON 格式数据能够被客户端和服务器端直接使用,大大减小了时延。

基于 JSON 数据轻便、易于读写的特点,文中选用 JSON 数据作为客户端和服务器交互的数据类型。不同的网络资源具有不同的数据内容,生成的 JSON 数据格式字段也各不相同。例如,QoS 策略 JSON 数据格式定义如下:

```
{
"node":{
"id":"00:00:2c:53:4a:00:bb:ec ",
"type":"OF"
},
"policys":{    //QoS 策略列表
"policy":{ //QoS 策略
"flow":{
"id":""//流匹配字段
},
"queue":{
"id":"" //队列}}}}
```

5 基于接口的 QoS 服务实现

区分服务的基本思想是在网络入口处为分组标记一个码点,码点用于指示分组在网络转发过程中应该被处理的方式。区分服务的处理流程包括分流和标记、分析与策略、队列和调度。文中针对 SDN 网络中端到端的流媒体数据传输场景,基于区分服务的思想,利用控制器接口提供的 QoS 资源控制操作,实现 SDN 网络中的区分服务,为流媒体传输提供不同的服务质量保证[14]。

(1)分流和标记。

当前网络中用于区分服务的分类标识一般是基于 IP 报文头部中的 DSCP(Differentiated Services Code Point,区分服务码点)。DSCP 优先级由 ToS 字节的 6

位组成,共表示 0 ~ 63 级服务优先级。OpenFlow 交换机流表项中的匹配字段能够实现网络报文的各个层次的匹配,因此,通过 API 在指定交换机上添加流表,即可实现数据的分流操作。

(2)分析与策略。

对于已经分类的数据流,根据其服务质量要求的不同,应提供不同的处理策略。此策略可以是基于一定的策略生成算法,也可由网络管理员根据网络状况进行调配。文中针对端到端的流媒体传输场景,通过接口对网络进行调整,提供不同的带宽服务。服务质量要求越高,高优先级的视频流将获得更大的带宽。

(3)队列和调度。

接口中提供的队列和计量表都能够实现对数据流传输速率的限制。文中为增加调用的灵活性,只采用交换机队列限制数据流速率。OpenFlow 流表项动作列表包含 enqueue 操作,可将已经分类的数据流添加到指定的队列中。

通过在应用层调用接口资源可以实现上述的处理流程,对不同的服务要求的数据流提供不同的带宽服务,得到一定的 QoS 保证效果。

6　测　试

文中的实验环境,使用一台服务器和若干台主机搭建小型的 SDN 模拟网络。首先在一台 DELL 服务器上部署 OpenDayLight 控制器基础版本,作为控制层面主要设备;在三台 DELL 台式机上部署 Open vSwitch 软件交换机,构建三个交换节点。所有交换机和控制器均运行 32 位 Ubuntu 12.04 桌面版系统。其他用于通信的主机节点均运行 Windows 系统。

6.1　测试方案

(1)测试系统提供的应用接口可用性,首先按照如表 4 所示的测试用例,测试接口 GET/POST/DELETE 方法可用性以及接口的快速响应能力。

表 4　接口方法测试用例

项目	方法及 URI
拓扑获取	GET/controller/nb/qos/Topology
队列查询	GET/controller/nb/qos/Queue / 00:00:2c:53:4a:00:bb:ec
流表添加	POST/controller/nb/qos/Flow/ 00:00:2c:53:4a:00:bb:ec
流表删除	DELETE/controller/nb/qos/Flow/ 00:00:2c:53:4a:00:bb:ec
异常情况	PUT/controller/nb/qos/Flow/ 00:00:2c:53:4a:00:bb:ec

(2)测试基于接口的 QoS 网络应用可用性。实验分别在三台通信主机上安装 iperf 客户端和服务器端,不间断发送 UDP 数据包,模拟两名用户同时观看不同清晰度视频的应用场景,对两名用户提供不同的服务质量保证。统计客户端实时带宽、丢包率、抖动情况,衡量客户端的服务质量。

6.2　测试结果分析

(1)经测试,在 OpenDaylight 控制器中部署的应用接口,能够对 HTTP 请求做出较快的响应,而且,对于错误的请求,也能及时反馈错误信息,具有良好的容错性。

上述测试用例的正常结果返回如图 3(a)所示,异常情况的结果返回如图 3(b)所示。

```
-<list>
 -<flowConfig>
    <installInHw>true</installInHw>
    <name>flow1</name>
   -<node>
      <id>00:00:2c:53:4a:00:bb:ec</id>
      <type>OF</type>
    </node>
    <priority>750</priority>
    <etherType>0x800</etherType>
    <nwDst>10.0.0.102</nwDst>
    <protocol>TCP</protocol>
    <tosBits>0</tosBits>
    <tpSrc>8080</tpSrc>
    <actions>OUTPUT=3</actions>
  </flowConfig>
</list>
```

(a) 正常查询

(b) 异常结果

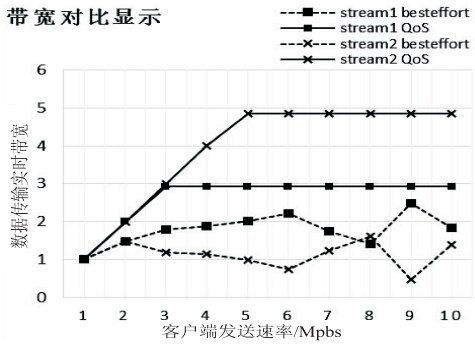
图 3　接口方法测试

(2)两名用户在具有服务质量保证和未有服务质量保证的前后两次测试过程中,链路传输的实时带宽、抖动、丢包率统计结果对比如图 4 所示。

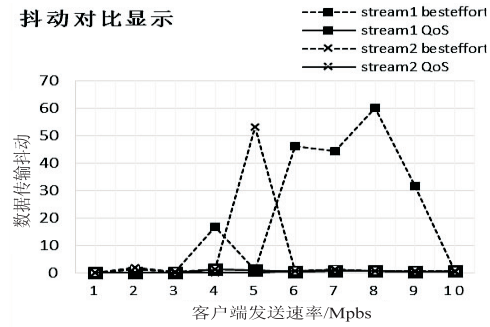
从结果可以看出,前后两次的带宽、抖动、丢包率情况具有明显的差异。通过应用接口添加服务质量保证后,链路传输明显更加稳定。这充分说明,基于应用接口编写的网络应用,能够提供良好的服务效果,文中设计的接口具有良好的 QoS 性能和可编程特性。

7　结束语

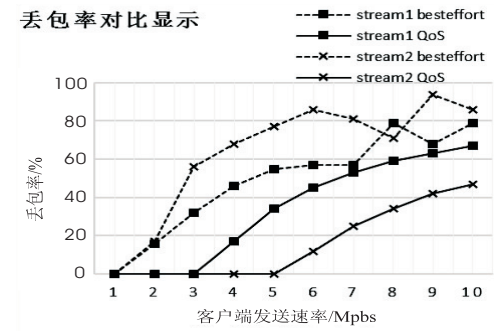
在 SDN 标准化制定过程中遇到了不少困难,尤其是控制器面向应用层尚未形成统一的接口标准。文中针对在全网领域难以形成统一的应用层接口标准,控制器缺少完整的服务质量保证解决方案的现状,基于 OpenFlow 协议流表和队列等服务质量支撑技术,采用 REST 风格 Web 服务实现方式,设计并实现控制器 QoS 应用可编程接口。按照区分服务的思想,提出基



（a）带宽



（b）抖动



（c）丢包率

图 4 结果对比

于应用可编程接口的 QoS 服务方案。经实验测试，该方案对 SDN 网络中的视频流传输，具有服务质量保证的能力。文中的实验场景比较简单，仍需设计更加完善的测试用例，发现系统不足，加以改进。

参考文献：

[1] Fundation O N. Software-defined networking: the new norm for networks[EB/OL]. 2012. https://www.opennetworking.org/images/stories/downloads/sdn-resources/white-papers.

[2] Tennenhouse D L, Wetherall D J. Towards an active network architecture[J]. ACM SIGCOMM Computer Communication Review, 2007, 37(5): 81-94.

[3] Open networking foundation[EB/OL]. 2012. https://www.opennetworking.org/.

[4] Gude N, Koponen T, Pettit J, et al. NOX: towards an operating system for networks[J]. ACM SIGCOMM Computer Communication Review, 2008, 38(3): 105-110.

[5] OpenFlow Consortium. OpenFlow switch specification, version 1.0.0[EB/OL]. 2009. http://www.openflow.org/documents/openflow-spec-v1.1.0.pdf.

[6] Open Networking Foundation. OpenFlow switch specification, version1.3[EB/OL]. 2012. https://www.opennetworking.org/images/stories/downloads/specification/openflow-spec-v1.3.0.pdf.

[7] Floodlight[EB/OL]. 2015. http://www.projectfloodlight.org/floodlight/.

[8] OpenDaylight[EB/OL]. 2012. http://www.opendaylight.org.

[9] Snir Y, Ramberg Y. Policy Quality of Service (QoS) information model, IETF RFC 2610[S/OL]. 2003. http://www.rfc-editor.org/rfc/pdfrfc/rfc3644.txt.

[10] Egilmez H E, Civanlar S, Tekalp A M. An optimization framework for QoS-enabled adaptive video streaming over OpenFlow networks[J]. IEEE Transactions on Multimedia, 2013, 15(3): 710-715.

[11] Egilmez H E, Dane S T, Gorkemli B, et al. OpenQoS: OpenFlow controller design and test network for multimedia delivery with quality of service[C]//Proc of networked and electronic media summit. Istanbul, Turkey: [s.n.], 2012: 22-27.

[12] Raumer D, Schwaighofer L, Carle G. MonSamp: a distributed SDN application for QoS monitoring[C]//Proc of federated conference on computer science and information systems. [s.l.]: IEEE, 2014: 961-968.

[13] 曲延盛，李 伟，罗军舟，等. 可信可控网络中的 QoS 资源控制模型[J]. 软件学报，2011，22(11)：2782-2794.

[14] 纪其进，陈 晞，董育宁. 区分服务模型[J]. 中国数据通信，2001，3(11)：33-37.

一种基于SDN的QoS应用编程接口设计方案

作者： 付健， 沈苏彬， FU Jian， SHEN Su-bin
作者单位： 南京邮电大学 计算机学院,江苏 南京,210003
刊名： 计算机技术与发展 ISTIC
英文刊名： Computer Technology and Development
年，卷(期)： 2015,25(11)