

基于 OpenFlow 的自适应 QoS 流量控制方法设计与实现*

戴彬, 郑俊, 徐冠, 杨军
(华中科技大学 电子信息与通信学院, 武汉 430074)

摘要: 目前已有的 QoS 流量控制方法大多存在对网络资源的利用率低、可靠性差、粒度粗、实现困难、可扩展性差等问题。如何利用有限的网络资源进行有效的流量控制,以保障业务的服务质量(quality of service, QoS)已成为一个非常迫切的问题。为解决此类问题,借鉴软件定义网络(software defined network, SDN)提出的控制层与数据层分离新思想,提出了一种基于 OpenFlow 技术的 QoS 流量控制方法,利用自适应多约束 QoS 路由技术提高了 QoS 控制的灵活性与可靠性,实现了对网络资源的高效利用和业务流控制的细粒度。实验结果验证了其有效性。

关键词: 软件定义网络; 服务质量; 流量控制; 自适应; 多约束路由

中图分类号: TP393.07 **文献标志码:** A **文章编号:** 1001-3695(2016)09-2759-04

doi:10.3969/j.issn.1001-3695.2016.09.045

Design and implementation of QoS traffic control method based on OpenFlow

Dai Bin, Zheng Jun, Xu Guan, Yang Jun

(School of Electronics Information & Communications, Huazhong University of Science & Technology, Wuhan 430074, China)

Abstract: The existing QoS traffic control methods still have many issues such as utilization of network resource, reliability, scalability, control granularity, complexity of implementation. It is urgent to guarantee QoS of applications with effective flow control constrained by network resources. This paper proposed an OpenFlow based QoS traffic control method, referring to the new idea of separating controlling layer and data layer provided by SDN. It took advantage of adaptive multi-constrained QoS routing technology, which improved flexibility and reliability for QoS control, to achieve effective usage of network resources and fine-grained of flow control. The experimental results validate its effectiveness.

Key words: SDN; QoS; flow control; adaptive; multi-constrained routing

0 引言

视频通话、电话会议、VoIP(voice over Internet protocol)等新型业务的兴起,对网络服务质量需求日益增加,如实时多媒体需要高比特率的吞吐量,VoIP 要求低的延时,网络游戏要求低的延时抖动率。目前“尽力而为”的网络体系已经很难满足这些业务的 QoS 需求。如何有效地利用目前有限的网络资源,去满足这些业务的 QoS 需求已经成为 Internet 发展的关键难题。尽管已提出了一些 QoS 解决方案,但都存在不少问题,如综合服务模型(integrated services model, IntServ)主要着重于每条流的要求参数,其使用了资源预留协议(resource reservation protocol, RSVP)来对网络资源进行预留,这要求路径上的路由器运行相同的协议和高速缓存所有流的信息。因此 IntServ 需要节点维护大量的数据信息包括每条流的预留信息,自身的资源信息同时还必须对数据流进行实时的监控,造成了实现复杂、拓展性差等缺点。区分服务模型(differentiated services model, DiffServ)使用数据包中的服务等级字段来进行流的

分类、标记和汇聚。每条流的状态信息保存在边缘路由器中,网络内部路由器则只对流提供不同优先级的服务。DiffServ 主要为汇聚流提供了区分服务的保证,无法实现对流的细粒度控制,无法保证每条流的 QoS。另外包括 MPLS 利用它的快速交换能力实现了部分功能,但其缺乏实时配置和可适应性。MPLS-DiffServ 要求对 QoS 流静态的分类,并且需要安装特殊的软件和硬件。其他的 QoS 自动化部署机制如 MPLS 流量工程要求策略服务器和策略配置代理等。

SDN 为解决现有网络体系下 QoS 流量控制机制存在的实现复杂、可扩展性差、灵活性低、控制粒度粗等问题提供了新的思路,逐渐成为目前网络领域的研究热点。从 2007 年提出到现在,SDN 已经在硬件和软件支持方面取得了长足的研究发展。SDN 的主要思想是将控制层从设备中分离出来,集成到统一的控制器中,而设备只需要接收来自控制器的信令执行相应的转发。本文提出了基于 OpenFlow 技术的自适应 QoS-guarantee 流量控制方法,充分发挥了 SDN 的优势,实现了对 QoS 流的可靠传输和细粒度控制,达到了对网络资源的有效利用。

收稿日期: 2015-06-21; **修回日期:** 2015-08-03 **基金项目:** 国家科技支撑计划资助项目(2012BAH93F01); 华中科技大学自主创新研究基金资助项目(2015MS037); 国家自然科学基金资助项目(60803005)

作者简介: 戴彬(1977-),男,江苏镇江人,副教授,工学博士,主要研究方向为云计算、SDN、移动互联网(dai bin@hust.edu.cn); 郑俊(1994-),男,硕士研究生,主要研究方向为 SDN 云计算、SDN 流量优化; 徐冠(1986-),男,博士研究生,主要研究方向为网络编码、SDN 流量优化; 杨军(1982-),男,研究员博士(后),主要研究方向为网络编码、网络安全。

1 相关研究

OpenFlow 最先是由斯坦福大学提出用来实现校园网络创新实验的平台。OpenFlow 是在 TCP/IP 技术的基础上提出的一种新型网络体系架构,它在传统的网络体系架构的基础上进行了革命性的改革,包括新型的 OpenFlow 网络模型和应用协议。OpenFlow 是 SDN 中一种标准化协议,主要用来实现控制层与底层设备进行通信的接口。OpenFlow 网络主要由 OpenFlow 控制器和 OpenFlow 交换机组成。图 1 是一个典型的 OpenFlow 网络,网络中 OpenFlow 交换机只需要对数据包进行转发,而如何控制数据包的转发则由控制器通过安全通道来传递控制信息,通常由 TLS 会话来实现。

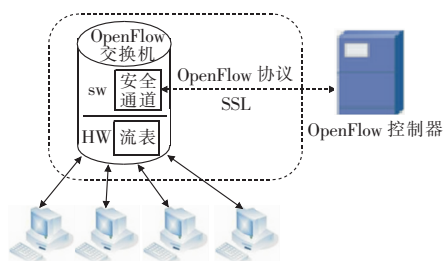


图1 OpenFlow网络

目前市场上 OpenFlow 交换机主要分为两类,分别基于硬件和基于软件实现的。本文实验中采用的是基于软件实现的虚拟 OpenFlow 交换机 open vswitch。OpenFlow 控制器是 OpenFlow 网络中用于整个网络的控制部件,可以控制交换机的一切行为,从而控制整个网络流量的行为。目前已经提出了一系列控制器,如 NOX、Beacon、Floodlight、SNAC、OpenDaylight 等。其中 OpenDaylight(简称 ODL)控制器功能强大、可扩展性强,易于设计和添加功能模块,开发者众多,是本文采用的控制器。

目前针对 SDN 中的 QoS 流量控制不少学者提出了相关的解决方案。Ishimori 等人^[1]通过扩展 OpenFlow 标准协议在 OpenFlow 数据通道中引入了多种队列调度机制来降低 QoS 数据包转发反应时间,有效地提高了视频流的 PNSR,其引入的队列调度算法包括 HTB、SFQ、RED 算法。Bueno 等人^[2]提出了 SDN 下对 QoS 策略进行灵活配置的框架,解决了提供端到端服务时的 QoS 配置限制,达到了对网络资源的有效利用,允许对网络按服务进行弹性动态配置,避免网络过度配置。Bari 等人^[3]提出了 QoS 策略自动化部署框架,主要是提出了 QoS 策略能否被准确执行,建立了策略违反机制保证策略的执行。Kim 等人^[4]针对汇聚流提出了自动化可扩展的 QoS 流量控制方法,其在交换机处采用了流量整形、速率限制、队列调度等技术来实现流量优先级的动态配置,通过调整流的优先级以尽可能降低对其他流传输的影响。Tomovic 等人^[5]提出了 QoS-aware 算法,主要通过 QoS 流与非 QoS 流之间进行不同的路由达到 QoS 流量控制的目的,其中 QoS 路由将带宽作为权重代入至 Dijkstra 算法中求得路径,当网络不能满足 QoS 流传输时,对非 QoS 进行了重路由以保证 QoS 流的传输,达到了 QoS 流量控制的目的。Egilmaz 等人^[6]针对视频流传输 QoS 要求提出了 QoS 路由算法,将多约束路由问题中的 CSP 问题应用到 SDN 中,利用控制器得到的链路和延时信息为 QoS 流计算出路径,得到了较好的服务质量。Egilmaz 等人^[7]进一步提出了自适应的视频流 QoS 传输机制,在视频流的增强层与基本层之间进行不同的 QoS 路由,以保证视频质量。Civanlar 等人^[8]

同样提出了可扩展的视频流 QoS 流量控制方法,尽可能降低 QoS 路径的长度,同时当链路拥塞时对视频流的基础层进行重路由以保证视频质量。

2 基于 OpenFlow 的 QoS 流量控制方法设计

尽管目前针对 SDN 的 QoS 流量控制提出的方法虽然可以在一定程度上实现 QoS 流量控制,但都存在一定的局限性。如文献[1,4]都只是从队列调度角度提出 QoS 的调度策略,没有考虑 QoS 路由。文献[2]提出的 QoS 策略自动配置框架并没有具体实现,且实现复杂度高。文献[5]提出的 QoS-aware 算法对 QoS 流没有进一步细致划分,因此运用相同的 QoS 算法没有充分考虑其他 QoS 要求,不能实现每条 QoS 流的细粒度保障。文献[6~8]都是针对视频流的特点设计了 QoS 路由算法,没有考虑资源的分配和其他 QoS 策略的下发,可扩展性差。而本文提出的 QoS-guarantee 方法则充分利用了 SDN 的流量控制能力,兼顾考虑了控制粒度、灵活性、扩展性等因素。

本文提出的 QoS-guarantee 流量控制方法主要是在 SDN 控制器进行设计与实现,具体框架如图 2 所示。

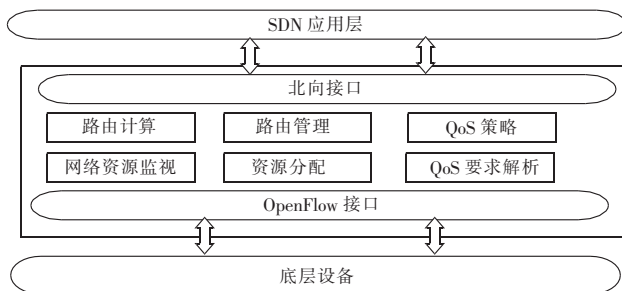


图2 QoS流量控制框架

QoS 流量控制框架主要包括网络资源监视、QoS 要求解析、QoS 策略库、QoS 路由计算与管理、资源分配,接入控制各模块。下面将分节对各个模块进行阐述。

2.1 网络资源监视模块

网络资源监视模块用来统计和监视当前网络链路状态信息,包括网络链路拥塞情况、链路的使用带宽、时延、抖动及丢包率信息,且基于 OpenFlow 协议实现。OpenFlow 协议中定义了计量表来统计流量的信息,包括计量带和计数器,其中计数器通过 OpenFlow 交换机计数器进行维护,可以面向流表项,面向端口、面向队列等进行统计,统计信息包括收发的字节数、传输错误情况、传输持续时间等。因此,假设单位时间 T 内统计某端口的字节数为 S ,则可以近似地将与该端口相连的链路使用带宽近似为 S/T 。链路延时考虑的主要是交换机转发时延,其通过在数据包添加时间戳方式,统计数据包的平均时延作为近似时延,假设某数据包入队时刻为 T_1 ,出端口转发时刻为 T_2 ,则转发时延为 $\Delta T = T_2 - T_1$,平均延迟 $\text{delay} = \sum \Delta T / N$, N 为周期统计的数据包数量。抖动作为延时的一阶导数得到;丢包率 $\text{loss} = \text{丢失的字节数} / \text{发送的字节数}$ 。

2.2 QoS 策略及要求解析模块

QoS 策略模块用于制定对 QoS 流的策略,包括针对特定应用或特定用户的策略,也可以是优先级策略、路由策略等。通过人为制定的策略对业务流进行控制,事先为对应数据流下发优先级高的流表。QoS 要求解析模块是通过对数据包中携带的 QoS 信息进行解析,用来获得业务流的带宽、延时或丢包率等 QoS 要求。目前 SDN 中控制器并没有对业务流 QoS 要求解

析机制。OpenFlow 定义的流表匹配字段包含了一系列匹配域,能够很好地支持 L2 ~ L4 层大部分基础协议。因此可以通过端口号或对报文的应用层协议解析识别,也可以在控制器与服务器之间建立 QoS 消息传递机制。本文是通过 ToS 字段进行解析,实现为不同 ToS 字段制定对应的 QoS 要求。

2.3 QoS 路由计算模块

QoS 路由计算模块通过得到网络中主机、交换机、拓扑、链路信息和指定的 QoS 路由算法计算出满足 QoS 传输要求的路径。其中指定的路由算法具体过程是:a)删除不符合传输要求带宽的链路;b)从剩下的网络拓扑中找到符合延时、抖动及丢包率要求的路径。其与多约束路由选择问题思想一致,当约束条件不小于 2 时是 NP-complete 问题,因此为了减少问题的复杂度将问题简化为 RSP(受限最短路径)问题。将 QoS 流的时延要求作为主要约束条件,抖动和丢包率作为代价参数, QoS 路由就变成从网络中找出满足时延约束,并且代价最小的路由。

网络拓扑可以用一个简单的有向图 $G(N, E)$ 来表示,其中 N 代表网络中交换机的集合, E 代表网络链路集合。 S 代表发送端, d 代表接收端; (i, j) 节点对代表从节点 i 到节点 j 之间的链路; r_{st} 代表发送与接收方之间的所有路由, r 代表路径。首先删除链路带宽不满足要求的链路。对于任何 $r \in r_{st}$ 的路由可以定义为

$$f_c(r) = \sum_{(i,j) \in r} c_{ij}, f_d(r) = \sum_{(i,j) \in r} d_{ij} \quad (1)$$

其中: d_{ij} 代表路由从节点 i 到 j 的时延, c_{ij} 代表路由从节点 i 到 j 的代价,其定义为

$$c_{ij} = (1 - \beta)g_{ij} + \beta p_{ij} \quad 0 \leq \beta \leq 1 \quad (2)$$

其中: g_{ij} 代表链路 i, j 之间的延时抖动; p_{ij} 代表链路 i, j 之间的丢包率。 β 值可变,对延迟抖动要求高的流可以将 β 取小,对丢包率要求更高的可以取大一些。因此寻找满足时延要求同时代价最低的路路由用数学公式可以表示为

$$r^* = \arg \min \{f_c(r) | r \in r_{st}, f_d(r) \leq d_{\max}\} \quad (3)$$

其中: d_{\max} 表示的是 QoS 流的延时要求。对于此多约束路由的求解问题,目前提出了很多数学算法如动态规划法、后向前向启发式算法、拉格朗日线性组合算法以及混合算法等^[10]。本文中采取的是拉格朗日松弛算法。拉格朗日松弛算法是解决数学中组合优化问题的经典算法,其基本思想是将优化问题转换为线性函数的求解问题,通过将约束条件转换为目标函数的参数,使得求解过程简单化,进而降低求解最优或较优解的时间^[11]。

2.4 QoS 路由管理及资源分配模块

QoS 路由管理模块用于 QoS 流资源分配流表的生成和 QoS 路由的状态包括延时、抖动、丢包率等的监视。一旦不满足 QoS 流传输要求则重新计算路由,通过直接删除对应路径交换机中的流表实现,流表是 OpenFlow 协议中针对交换机定义的转发规则。OpenFlow 的计量表中定义的计量带类型和流表中的指令动作集可以用来进行限速和队列优先级服务。因此资源分配模块从 QoS 路由管理模块获得路径及资源分配流表后,将流表信息传递给转发规则管理模块生成流表下发至底层设备,为 QoS 流创建高优先级的队列,保证传输带宽。另外在交换机上采用了 WRR 队列调度算法,其通过权重来分配带宽,既能保证高优先级队列的优先服务,又避免了低优先级“饿死”的情况。

3 QoS 流量控制方法实现

3.1 基于 OpenDaylight 的 QoS 流量控制实现框架

本文采用 OpenDaylight 控制器来实现所设计的 QoS 流量控制方法。OpenDaylight 控制层主要包括服务抽象层(SAL 层)和基本网络服务功能。SAL 服务层支持多种南向协议,屏蔽了协议间差异,为上层模块和应用提供一致性的服务。其提供的服务有数据包服务、拓扑服务、流编程服务、资源查询服务、连接服务、统计服务、清单服务等。基本服务功能模块包含拓扑管理模块、统计管理模块、交换机管理模块、转发规则管理模块和主机追踪模块^[9]。其中拓扑管理用于管理网络的拓扑信息主要包括交换机之间、交换机与主机间的链路信息。交换机管理用于维护管理底层的交换机信息,包括交换机位置信息、配置信息等。统计管理模块用于统计管理流过每个交换机的数据流、端口的统计信息,包括收发的数据包数量、字节数、丢弃包数量。转发规则管理负责管理流表规则,实现流表规则的添加、删除、更新、下发等操作。主机追踪模块负责追踪主机信息,记录主机的 IP、MAC、VLAN 以及连接交换机的节点和端口信息。基于 OpenDaylight 的模块实现框图如图 3 所示。

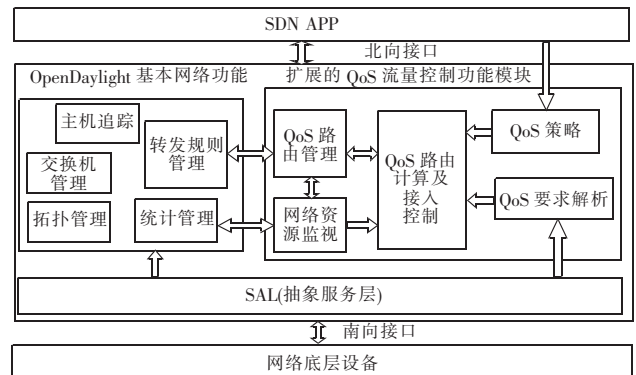


图3 基于 OpenDaylight 的模块实现框图

3.2 QoS 流量控制方法实现流程

通过对 OpenDaylight 中已实现框架和 OpenFlow 1.3 协议^[12]流程的分析,本文设计了如图 4 所示的 QoS 流量控制工作流程。流的建立过程包括:首先发送的第一个数据包会发送给控制器处理,控制器数据包服务可以提取数据包 ToS 字段,判断是否标记,若无则对非 QoS 流直接采用 Dijkstra 最短路径算法计算得到路径,下发表至底层设备;若标记则属于 QoS 流进行 QoS 路由,进一步解析 ToS 字段信息,得到预先制定的各 QoS 要求参数;同时控制器每隔 2 s 查询网络链路状态信息,这主要是从 OpenvSwitch 角度出发,在实验中计数器的数值从内核到用户空间所花的时间为 1 s。

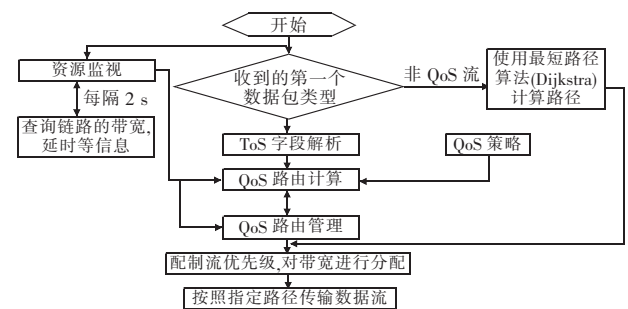


图4 QoS 流量控制实现流程

4 实验结果

4.1 实验环境

实验采用的拓扑如图5所示,使用 Mininet 软件来生成拓扑。Mininet 可以人为定制实验拓扑,定制每条链的带宽和延迟、通断等,并且包含 Open vSwitch。

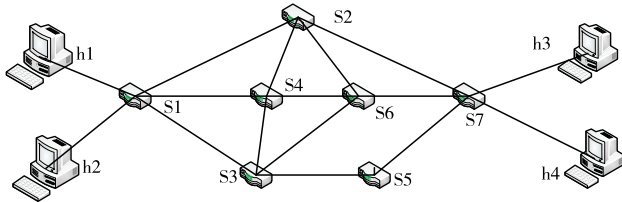


图5 实验拓扑

实验采用 Iperf 来产生不同类型的流量,Iperf 可以产生指定长度、指定时间、指定类型的数据流。实验开始前设置主要传输链路的带宽为 100 Mbps。实验的主要测试指标为吞吐量和丢包率。表1表示的是实验中所产生数据流的信息,包括速率、传输协议、ToS 字段、源目的主机及产生时刻。

表1 实验数据流

流序号	源主机-目的主机	传输协议	速率	ToS 字段	产生时刻
flow1	h1-h3	UDP	30 Mbps	4	0
flow2	h1-h4	UDP	40 Mbps	8	10 s
flow3	h1-h3	UDP	60 Mbps	0	30 s
flow4	h2-h4	UDP	60 Mbps	0	50 s
flow5	h2-h3	UDP	50 Mbps	12	70 s
flow6	h2-h4	UDP	60 Mbps	16	90 s

4.2 实验结果分析

通过吞吐量和丢包率实验结果对比可以看出,在未采取 QoS 流量控制时,每条流之间相互竞争带宽,随机分配有限的带宽,导致每条流吞吐量低,网络资源没有得到有效的利用,进而导致丢包率严重,如图6、7所示。采用 QoS-aware 算法后,QoS 流传输得到了一定的保证,丢包率也降低了不少,如图8、9所示。但在90 s后,flow5、flow6 流传输依然没有得到预先设计的吞吐量。这主要由于 QoS-aware 算法中路由只采用了改进的 Dijkstra 算法,仅将链路带宽权值加入到路由计算中,尽管通过对非 QoS 流进行重路由来保证 QoS 的传输要求一定程度上实现了负载均衡和 QoS 流保证,但对 QoS 流没有划分,导致了部分高 QoS 流不能获得其要求的吞吐量,影响了部分 QoS 流的可靠传输。而采用本文提出的 QoS-guarantee 流量控制方法后,控制器通过解析流的 ToS 字段得到 QoS 流要求带宽,给相应 QoS 流分配要求的带宽,吞吐量达到要求,使丢包率降到最低,网络资源得到有效的利用,如图10、11所示。尽管在30 s时由于 flow3 的传输导致 flow1、flow2 的吞吐量降低,丢包率增大,但经过2 s左右,吞吐量又恢复正常,丢包率降低,达到了 QoS 流的可靠稳定传输。综上所述,设计的基于 OpenFlow 的 QoS 流量控制方法实现了对资源的有效利用,对每条 QoS 流都实现了可靠传输的控制粒度,并且在受到干扰时能够及时调整路由,保证了 QoS 传输的高可靠性和自适应性。

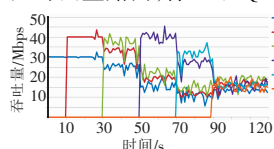


图6 未采用 QoS 算法的吞吐量

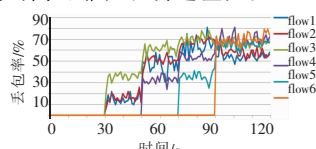


图7 未采用 QoS 算法的丢包率

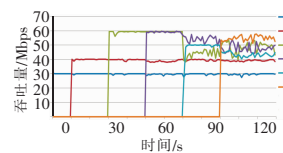


图8 采用 QoS-aware 算法的吞吐量

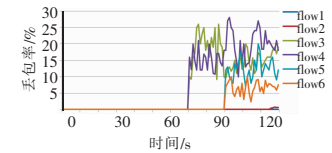


图9 采用 QoS-aware 算法的丢包率

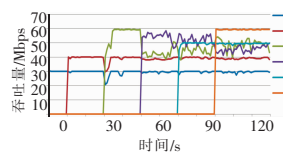


图10 采用 QoS-guarantee 算法的吞吐量

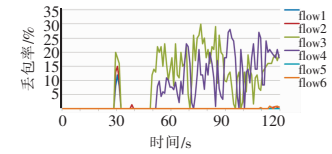


图11 采用 QoS-guarantee 算法的丢包率

5 结束语

本文利用 OpenFlow 网络中控制器能够获取整个网络链路、节点状态的实时精确信息的特点,提出了基于 OpenFlow 技术的 QoS-guarantee 流量控制方法,提出的自适应多约束 QoS 路由技术实现了对 QoS 业务流的可靠传输和网络资源的高效利用。与其他 SDN 下的 QoS 流量控制方法相比,本文提出的方法具有较强的灵活性、可靠性、适应性及可扩展性。在今后的工作中,笔者将在更复杂的网络环境中进行实验证明其有效性。

参考文献:

- [1] Ishimori A, Farias F, Cerqueira E, et al. Control of multiple packet schedulers for improving QoS on OpenFlow/SDN networking [C]//Proc of the 2nd European Workshop on Software Defined Networks. [S. l.]: IEEE Press, 2013: 81-86.
- [2] Bueno I, Aznar J I, Escalona E, et al. An OpenNaaS based SDN framework for dynamic QoS control [C]//Proc of IEEE SDN for Future Networks and Services Conference. [S. l.]: IEEE Press, 2013: 1-7.
- [3] Bari M F, Chowdhury S R, Ahmed R, et al. PolicyCop: an autonomic QoS policy enforcement framework for software defined networks [C]//Proc of IEEE SDN for Future Networks and Services. [S. l.]: IEEE Press, 2013: 1-7.
- [4] Kim W, Sharma P, Lee J, et al. Automated and scalable QoS control for network convergence [C]//Proc of Internet Network Management Workshop on Research on Enterprise Networking. [S. l.]: USENIX Press, 2010.
- [5] Tomovic S, Prasad N, Radusinovic I. SDN control framework for QoS provisioning [C]//Proc of the 22nd Telecommunications Forum Tel-for. [S. l.]: IEEE Press, 2014: 111-114.
- [6] Egilmez H E, Dane S T, Gorkemli B, et al. OpenQoS: an OpenFlow-controller design for multimedia delivery with end-to-end quality of service over software-defined networks [C]//Proc of Asia-Pacific Signal & Information Processing Association Annual Summit and Conference. [S. l.]: IEEE Press, 2012: 22-27.
- [7] Egilmez H E, Civanlar S. An optimization framework for QoS-enabled adaptive video streaming over OpenFlow networks [J]. IEEE Trans on Multimedia, 2013, 15(3): 710-715.
- [8] Civanlar S, Parlakisik M, Gorkemli B, et al. A QoS-enabled OpenFlow-environment for scalable video streaming [C]//Proc of IEEE GLOBECOM Workshops. [S. l.]: IEEE Press, 2010: 351-356.
- [9] OpenDaylight architecture overview [EB/OL]. (2015-03-07) [2015-06-01]. <https://www.opendaylight.org>.
- [10] 朱慧玲, 杭大明, 马正新, 等. QoS 路由选择: 问题与解决方法综述 [J]. 电子学报, 2003, 31(1): 109-116.
- [11] Juttner A, Szviatovski B, Mecs I, et al. Lagrange relaxation based method for the QoS routing problem [C]//Proc of the 20th Annual Joint Conference of IEEE Computer and Communications Societies. [S. l.]: IEEE Press, 2001: 859-868.
- [12] OpenFlow 1.3 specification overview [EB/OL]. (2015-01-24) [2015-06-03]. <http://sdnhub.org/tutorials/openflow-1-3>.