

# Research on QoS Technology Based on SDN Network

Siping Wang, Xiangquan Shi

School of Computer Science and Technology, National University of Defense Technology, Changsha Hunan  
Email: 15114512670@163.com

Received: Jun. 3<sup>rd</sup>, 2016; accepted: Jun. 19<sup>th</sup>, 2016; published: Jun. 22<sup>nd</sup>, 2016

Copyright © 2016 by authors and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

---

## Abstract

The Quality of Service (QoS) of IP network has been a hot issue concerned by many researchers and they put forward a series of theories and solutions. QoS provides guarantee of bandwidth, delay, packet loss rate, etc. for a variety of network applications and it is essential for multimedia applications. This paper analyzes the problems of traditional QoS technical and realizes the QoS technology based on SDN network. To enhance the quality of service QoS based OpenFlow protocol some key technologies were introduced. This article is a guideline of using SDN network to realize QoS technology; simultaneously it provides important practical reference in implementation of internet service based on the SDN framework.

## Keywords

Software-Defined Networking, Quality of Service, QoS Technology

---

# 基于SDN网络的QoS技术研究

王嗣平, 时向泉

国防科学技术大学计算机学院, 湖南 长沙  
Email: 15114512670@163.com

收稿日期: 2016年6月3日; 录用日期: 2016年6月19日; 发布日期: 2016年6月22日

## 摘要

一直以来, 互联网IP网络的服务质量(Quality of Service, QoS)是研究人员关注的热点问题, 提出了一系列理论和解决方案。QoS为各种网络应用提供了带宽、延迟、丢包率等的保证, 对多媒体应用至关重要。本文分析了传统QoS技术存在的问题, 并提出了利用SDN技术提升网络QoS服务质量的观点。对基于OpenFlow协议提升QoS服务质量的一些关键技术进行了介绍。本文是利用SDN网络实现QoS技术的指南, 同时也为其他网络服务在SDN框架下的实现提供了重要的实际借鉴意义。

## 关键词

软件定义网络, 服务质量, QoS技术

## 1. 引言

据中国互联网络信息中心(CNNIC)发布的第36次全国互联网发展统计报告, 截至2015年6月, 我国网民总数已达6.68亿人, 互联网普及率为48.8% [1], 说明以互联网为代表的网络技术已经成为人们日常生活中不可却缺少的一部分。互联网技术在短短二十年的商业化浪潮中, 以前所未有的速度在全球范围掀起一场影响人类所有层面的深刻变革, 引发各产业生产方式、生产关系、生产要素的重新组合、建构, 在很多方面影响和改变了人们的生活方式。但随着网络技术的发展, 越来越多的分布式应用和不同类型的网络技术被部署到网络上, 基于传统IP的网络体系结构正面临越来越多的问题, 传统的优势正逐渐成为制约网络技术发展的瓶颈。由于互联网的多提供商特性(即Internet的骨干网结构是由多个ISP互连组成), 部署一种新的体系结构或者对现有的体系结构进行修改都需要得到所有ISP的同意。因此, 对Internet体系结构的更新和部署一些新的网络技术正变得越来越艰难。同时, 随着网络技术的发展, 网络应用范围的进一步扩大, 网络已经由单一的数据传输网络发展为需要支持数据、视频、语音以及实时多媒体的多样化网络, 使得当前单一的网络基础设施很难同时满足上述多样化的网络应用需求。同时, 电信、广播、电视运营商开始考虑将互联网作为其业务的主要承载平台, 三网融合的趋势已经越来越明显, 以秒拍、IPTV、在线视频播放、网盘及网络直播等为代表的新型流媒体应用不断涌现, 这将会进一步加剧网络应用需求的多样化和复杂性[2]。由于话音、电视和数据三大类的业务对服务质量的需求差别很大, 需要设计一种有效的网络架构和管控方式, 保证其不同业务类型的服务质量, 并从更小的粒度保证每种大业务类别中不同子类的服务质量。研究表明, 很难设计出一套可以满足上述所有需求的网络体系结构。

## 2. QoS 技术

在现有的互联网体系架构中, 尽管在报文头部中包含了服务类型(Type of Service)字段, 但在路由器处理中通常被忽略, 网络不提供相应的服务, 而采取尽力而为(Best Effort)的传输转发方式, 只要把报文转发至目的节点即可, 提供端到端的通信, 几乎不提供服务质量, 其端到端的可靠性传输通过端到端的TCP协议中出错重传机制来保证。

没有QoS的网络被称为尽力服务网络。在尽力服务网络设计中, 所有的分组都被看成是同等重要。但通常网络中存在着对网络资源的大量竞争, 以及网络需要提供给不同用户不同等级的服务。要从常常有争夺发生的稀缺资源的网络中获得适当服务级别, 必须应用QoS的基本概念, 即全部的报文将不平等。为了解决竞争, 必须有可管理的不平等来对待报文转发, 在网络的基础设施中定义和部署管理策略, 以确保每个节点对单个分组(或者流)的相对重要性做出一致的判断。

QoS (Quality of Service, 服务品质保证)是指一个网络能够利用各种技术解决网络延迟和阻塞等问题,向选定的网络通信提供更好服务的一种安全机制。有关于如何发展互联网,以便为多媒体业务提供 QoS 是一个持续不断的争论。目前,全球范围内没有任何的 QoS 体系结构是得到广泛应用的。一些研究人员认为,根本性的变革应该是怎样做才能充分保证 QoS,而另一些人认为只要轻微的变化,就会有足够的保证,在很大程度上提供所请求的 QoS。

目前,互联网工程任务组(IETF)已经探索了一些服务质量(Quality of Service, QoS)的体系结构[3],包括以下几种:综合/集成服务模型(IntServ) [4]、区分服务模型(DiffServ) [5]、MPLS (Multiple Protocol Label Switching) [6]、流量工程模型(Traffic Engineering, TE)等。其中 IntServ、DiffServ、MPLS 是目前主流的 QoS 模型。

IntServ 的主要框架是在传送数据之前,为 IP 保留了端到端的 QoS 语义。关键端点是指这些发送和接收应用,他们强烈需求网络服务提供一组流,其中流由源地址、目标地址、传输协议、源端口以及目的地端口等定义。其协议为 RSVP (Resource Reservation Protocol, 资源预约协议) [7], RSVP 面向流在传输路径上为流预留并维护相应的资源,主机利用 RSVP 向网络为应用提出的 QoS 请求,网络设备(如路由器)建立和维持本地资源预留,并利用 RSVP 将 QoS 请求信息往目的路径的设备传送,从而形成整个流路径的资源预留。

DiffServ 的主要框架采用层次化结构,简化 DS 域内部路由器的功能,将复杂的 QoS 服务功能放在边界路由器上,在网络入口或在网络边界上,在 IP 头中设置一个字段(IPP 或 DSCP),使用这个字段来决定网络内转发分组的节点,根据每个“类别”或服务的要求在网络边界上对被标记的分组进行调解,在网络核心只保留简单的 DSCP 与每跳行为(PHB)的对应机制,根据 DSCP 值对数据包进行相应优先级转发控制[8]。目前标示报文优先级的方法有三种:IEEE802.1p 使用的二层报文的 CoS 字段、IP 报文的 ToS 字段和基于 IP 层的 DSCP 字段。

基于 MPLS 的 QoS 映射模型的基本内涵是在 MPLS 增加的 2.5 层头部内部,将流标记域与 MPLS 的标签中的 EXP 域进行映射,并且利用 MPLS 提供的标记交换技术来实现报文的高速转发。基于 MPLS 的 QoS 映射模型主要优点描述如下:利用 MPLS 实现流量工程的特性,可充分利用全网的拓扑资源,从而实现全网流量的负载均衡,也就能更好地为每条流提供 QoS 服务。其缺点描述如下:尽管 MPLS 提出了通过其高速交换能力的部分解决方案,但属于静态性的、预先分配性的实现,缺少实时地配置 QoS 的能力和动态的适应能力。

以上这三种架构都没有得到真正的成功和在全球范围内广泛部署。这主要是因为这些 QoS 体系结构,如 IntServ 和 DiffServ 是建立在目前互联网完全分布式的、逐跳路由式的体系结构之上,缺乏整体网络资源分布这种统一的全局视图。

为了解决上述问题,并且还要为了满足网络应用的多样化 QoS 的需求,迫切需要从网络控制架构的本身改变来解决从全局角度,统一解决当前网络部署 QoS 服务存在的各种问题。

### 3. 软件定义网络

软件定义网络(Software Defined Networking)最初是由斯坦福大学 Nick McKeown 教授等人在 2009 年左右提出的新型网络架构理念。通过实现网络设备中控制与转发的功能分离,将控制逻辑集中到一个统一的智能计算设备上,并实现控制逻辑的可编程。这种控制的迁移提供底层网络驻留在上层的应用程序的抽象,使他们能够把网络作为一个逻辑或虚拟实体,网络管理应用与网络服务从基础设施中抽象出来,将原来封闭的网络设备“黑盒子”系统转变为灵活开放的网络架构,由原来针对不同的、独立的、运行各种分布式协议的网络设备的控制变为更加简单的工作过程,即快速简单的转发硬件和智能的控制器决

策。这种将控制逻辑从转发硬件分离的分层架构使得网络使用者能够快速部署网络新协议和网络应用, 将所有的网络资源描绘成统一的资源池, 为不同的网络管理应用提供全局视图, 以更好地实现网络管理和

服务。  
图 1 描述了 SDN 体系结构的逻辑架构图, 底层的网络基础设施由 SDN 交换机等网络设备组成, 这些网络设备只需要接收执行上层控制器发来的指令, 而不再需要理解和处理成千上万的协议标准。这些指令封装在所谓的南向接口中(如 OpenFlow、NETCONF、BGP 等), 实现了上层控制器与底层网络基础设施层之间的信息交互。根据不同的硬件形态和性能需求, 基础设施层可由多种网络设备组成, 采用不同的硬件。例如图 1 中所示的网络处理器(Network Processor, NP)、硬件可编程 FPGA (Field Programmable Gate Array)、通用处理器(CPU)、网络处理芯片 ASIC (Application Specific Integrated Circuits)等。网络的智能行为由基于软件的控制层负责, 控制层通过标准的南向接口与底层的网络设备交互, 并维护了一个网络的全局视图。这个控制层也通常被称为网络操作系统。目前研究比较火热、研究人员开发使用比较多的是 Trema、Floodlight、OpenDayLight、ONOS 等网络操作系统。在全局网络视图的基础上, 控制层为上层应用提供了充分的编程接口。这些应用包括负载均衡、防火墙、移动终端管理、传统的 L2/L3 转发、代理等各种应用, 它们实现了相应的网络管理操作和服务。

SDN 最本质的特点就是网络控制与转发相分离, 并为网络管理人员提供了充分的可编程管理接口。这种控制与转发分离的一个重要功能在于, 它使得网络控制逻辑能够以一个全局网络的视野去设计、理解和管理网络, 降低了网络系统的复杂度, 提升了网络管理应用的控制能力; 另一方面, 控制转发的分离使得统一各式各样的网络设备成为可能, 打破了网络设备的封闭性, 改善了网络产业的生态链。而可编程的控制平面则能够让网络管理人员灵活地配置网络功能, 并实时地改变网络行为, 在增加网络灵活性的同时, 降低了网络技术革新的难度。

OpenFlow 是 SDN 中第一个也是目前唯一一个定义在控制层和基础设施层之间的标准通信接口, 主要由开放网络基金会 ONF (Open Network Foundation)组织负责维护[9]。上文提到的其他南向接口均为各个厂商私有的接口, 并未被标准化和受到所有厂家的认可。通过 SDN 控制器在数据平台中配置的规则, OpenFlow 使用流(Flow)的概念对网络进行识别, 这些规则被称为流表(Flow Table)。除了匹配规则, OpenFlow 还规定了流表中的动作, 包括转发报文和修改报文头内容等。通过流表, OpenFlow 可以在每个流的基础上对网络进行编程, 在实现对报文流精细控制的同时简化了报文处理复杂度。OpenFlow 协议发展过程如图 2 所示。

ONF 于 2009 年 12 月发布第一个版本 OpenFlow 协议的规范, 即 OpenFlow v1.0。其流表由匹配域、计数器、动作域三部分组成。匹配域中包含了报文头中 L2-L4 内 12 个字段的信息。随后, ONF 根据网络处理的需求, 引入了 IPv6、多级流表、组表、多控制器等概念, 不断地对协议本身进行修订。目前版本已更新到 OpenFlow v1.5, 流表包含匹配域、优先级、计数器、指令集、生存时间和 cookie 六部分, 匹配域采用更加灵活的基于 TLV (Type Length Value)的方式。其中匹配域包含了 40 多个不同协议类型的报文头域, 指令集中包含了 10 多个特定的协议相关报文处理动作。OpenFlow 协议规范从最初对单流表以及基本的二层、三层协议匹配域的支持, 演进到对多级流表架构和多协议匹配域及操作的支持, 已适用于大部分的网络应用场景。但随着网络设备的进一步开放, 新型网络协议的不断引入, 网络管理用户对设备可编程性要求不断增加, OpenFlow 协议规范仍将持续发展和演进更新, 如有研究人员提出支持协议无感知转发的高级编程语言 P4 等。

SDN 的设计者认为 SDN 简化了网络架构, 因而减少操作和控制的开销。所以其具有以下几个方面的优势:

- 1) 底层设备标准化。OpenFlow 逐渐成为控制器与网络设备之间通信接口的标准, 底层转发设备只

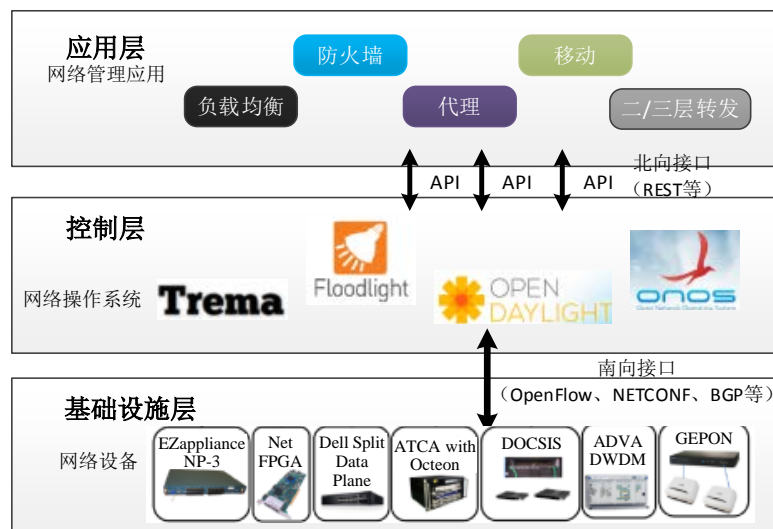


Figure 1. SDN architecture  
图 1. SDN 体系架构

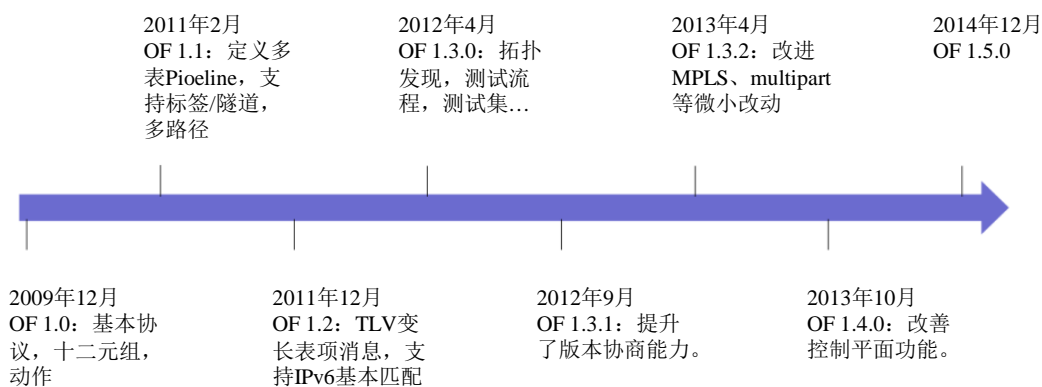


Figure 2. OpenFlow protocol development flow chart  
图 2. OpenFlow 协议发展流程图

需要通过 OpenFlow 一致性测试, 即可进行部署。网络管理员不需要考虑底层转发设备造成的内部实现差异。

2) 灵活的 QoS 服务策略选择。在 SDN 的工作模式下, 由于采用全局的带宽分配方式, 两种服务均可得到保障。网络管理员根据用户对传输流的需求, 灵活地选择相应的 QoS 策略。策略可以是类似于区分服务这种在局部区域按照权重竞争带宽资源, 可以是类似于集成服务这种保证某条流的绝对带宽。SVQoS 下灵活的 QoS 服务策略可以更高效地实现带宽分配, 充分利用全网的带宽资源。

3) 自动化的流量带宽配置。基于 OpenFlow 的 SDN 本身具有自动化网络配置的特性, 因此对于 QoS 来说, 可利用控制器中的接口, 实现自动化的流量带宽配置, 仅仅需要改变对集中控制中 QoS 参数输入的设置, 即可改变全局的 QoS 策略, 这将大大简化网络 QoS 管理过程中的操作复杂度, 复杂度从  $O(n)$  降为  $O(1)$ 。

4) QoS 策略的一致性。无论是集成服务还是区分服务模型, 网络管理员的 QoS 策略都是基于分布式的协议实现, 配置过程采用逐跳配置的形式, 这就必然导致策略更改带来的不一致情况。QoS 策略不一致的另一种情况是, 用户定义 QoS 策略带来的策略冲突以及可能的错误的资源分配。而本文提出的利用

SDN 集中控制的特性, 所有的 QoS 策略(无论是用户还是网络管理员)都通过集中控制器下发, 实现对所有网络设备以及全网流量的集中管控, 消除了以上两种情况下策略不一致带来的问题。

#### 4. OpenFlow 对 QoS 的支持

OpenFlow [10]是 SDN 中第一个也是目前唯一一个定义在控制层和基础设施层之间的标准通信接口, 主要由开放网络基金会组织负责维护, 目前已更新到版本 1.5.1, 而且有向 OpenFlow2.0 演进发展的趋势。一个 OpenFlow 交换机有以下几部分组成: 流表和 OpenFlow 通道。OpenFlow 通道用于交换机与控制器之间交互控制信息, 例如下发流表规则等等。流表是流规则的集合, 每条规则包含以下几个域: 匹配域, 以多级流表的形式实现, 每个匹配域定义了一种处理报文流的行为, 目前已包括达到 41 种报文头部; 动作执行的集合, 包括丢弃、转发、修改报文头域、转发至控制器等等; 计数器的集合, 记录了报文匹配的数目和字节数等等; 优先级, 用于确定规则在冲突情况下的执行顺序。基于 OpenFlow 的报文处理过程描述如下。交换机收到报文后, 通过查询流表得到最高优先级的匹配规则, 执行规则中相应的操作, 并且增加相关的计数器。通常来说, 如果是转发操作就转发至对应的端口。如果未命中, 则通过 OpenFlow 通道发送至控制器。

在 OpenFlow 的规范中, 主要定义了两种机制共同实现 QoS 流量调度: 计量表(meter)和队列设置操作(set queue) [11]。

计量表包含了计量表项, 定义了每条流的计量参数。每条流的计量使得 OpenFlow 实现了限速功能, 即将一条流的速率限定在一个固定大小的带宽内。每条流的计量也可以实现更复杂的 QoS 功能, 例如基于 DSCP 的计量可以把报文集合进行分类, 并且映射到不同的计量表上, 实现速率限制。计量表操作与队列设置相互独立, 互不干扰, 两者可以相互配合共同协作, 完成 QoS 的功能。本文为简化 OpenFlow 流表的规则的配置复杂性, 仅采用下面论述的队列设置操作实现 QoS 流调度。

在最新的 OpenFlow 规范中指出, OpenFlow 交换机通过实现简单的队列机制, 提供了 QoS 服务的支持。具体实现描述如下。一个端口中有一个或者多个队列。队列的配置由 OF-CONFIG 协议[12]实现, 是与 OpenFlow 协议无关的。OF-CONFIG 也是由 ONF 提出的, 用于对 OpenFlow 交换机进行远程的管理配置。与 OpenFlow 协议职责不同, OF-CONFIG 并非对流表的内容和数据转发行为进行配置, 而是对例如 OpenFlow 交换机的控制器 IP 地址、交换机端口的队列等进行配置。因此该协议对实时性没有太高的要求。OF-CONFIG 对 OpenFlow 交换机的队列属性配置包括队列的最小速率(Minrate)、最大速率(Maxrate)、自定义速率(Experiment) 3 个参数。

控制器通过 OpenFlow 消息中的 OFPMP\_QUEUE\_DESC 对 OpenFlow 交换机中配置的队列进行读取访问。OFPMP\_QUEUE\_DESC 消息中针对每个端口和每个队列提供了队列的描述。这个请求消息的结构体描述如下:

```
/*ofp_multipart_request 中 OFPMP_QUEUE_DESC 消息的结构体*/
struct ofp_queue_desc_request {
    uint32_t port_no; /* All ports if OFPP_ANY. */
    uint32_t queue_id; /* All queues if OFPQ_ALL. */
};
```

OFPMP\_QUEUE\_DESC 请求消息中包含了要查询的端口号(port\_no)和队列号(queue\_id)。宏定义 OFPP\_ANY 和 OFPQ\_ALL 分别表示查询所有端口和队列号。OpenFlow 交换机根据收到的 OFPMP\_QUEUE\_DESC 请求消息, 返回相应的应答消息。应答消息的数据结构如下所示:

```
/* OFPMP_QUEUE_DESC 请求对应回复消息的结构体*/
```

```

struct ofp_queue_desc {
    uint32_t port_no; /* Port this queue is attached to. */
    uint32_t queue_id; /* id for the specific queue. */
    uint16_t tlen; /* Length in bytes of this queue desc. */
    uint8_t pad[6]; /* 64-bit alignment. */
    struct ofp_queue_desc_prop_header properties[0]; /* List of properties. */
};

```

OFPMQ\_QUEUE\_DESC 应答消息中包含了队列相关的信息, 包括要查询的队列所在的端口号(port\_no)、队列号(queue\_id)、队列的长度(len)和队列的属性描述(ofp\_queue\_desc\_prop\_header properties)。

OpenFlow 中队列的配置根据不同的类型定义不同的属性, 共有以下三种属性:

```

/* 队列类型定义 */
enum ofp_queue_desc_prop_type {
    OFPMQDPT_MIN_RATE = 1, /* Minimum datarate guaranteed. */
    OFPMQDPT_MAX_RATE = 2, /* Maximum datarate. */
    OFPMQDPT_EXPERIMENTER = 0xffff /* Experimenter defined property. */
};

```

基于 OpenFlow 的 SDN 提供逻辑集中的控制平面, 为支持灵活的流量调度策略, 以满足不同网络应用的需求, 缓解网络中某些关键路径或者是热点交换机/路由器等网络设备的拥塞。现有的关于利用 SDN/OpenFlow 支持 QoS 的相关研究描述如下:

Kim 等人[13]提出了 QoS 网络控制框架, 允许对 QoS 参数的编程, 用高级切片描述的方式实现对网络中设备的 QoS 配置。然而, 文章中并没有提出基于 OpenFlow 相关的自动 QoS 控制框架。

Egilmez 等人[14]提出开放的流媒体控制体系架构 OpenQoS, 为多媒体视频流的传输提供 QoS 保证。利用基于多媒体视频流的报文头字段与其它类型的数据流中报文头字段存在的差异, OpenQoS 把针对于 OpenFlow 的匹配规则将传输的流量分为多媒体视频流和其他数据流两种类型。针对多媒体视频流, OpenQoS 结合传输路径在延迟、丢包测量得到的特性, 在 OpenFlow 控制器上增加了服务层, 为其选择一条满足 QoS 参数的传输路径, 其它数据流仍然维持在原有的最短路径上。该方案的不足是, 仅对多媒体视频流进行调度优化, 并没有考虑多个有 QoS 需求视频流同时进行调度的场景。

基于上述研究, Egilmez 等人在 OpenQoS 的基础上, 针对大规模的 SDN 网络部署, 提出了分布式的控制平面框架, 支持多运营商在多媒体业务流方面的 QoS 服务需求。由于分布式的控制平面包含多个控制器, 每个控制器在其管理的区域执行最优的 QoS 路由, 那么他们之间为了实现端到端的 QoS 服务就必须与其他区域共享聚合的 QoS 路由信息。所以该研究主要提出了跨多个域实现端到端 QoS 服务的优化框架, 设计了不同域间控制器交互 QoS 路由信息的信息机制。最后通过实验证明了这种分布式的 QoS 框架能达到接近最优的 QoS 保证。

Ishimori 等人[15]从 SDN 数据平面的角度, 即关注于 QoS 在底层 OpenFlow 设备中的实现, 提出了对每个报文进行 QoS 调度的控制机制, 也为每个服务提供不同的调度算法和流量整形。本文在数据平面实现时也借鉴该文章的一些方法。

Sonkoly 等人[16]对 Ofelia Control Framework (OCF)提出 QoS 形式化, 并在其实验床上实现细粒度的 QoS 控制。实现的扩展模块包括 OCF Expedient, Opt-In Manager, FlowVisor, 和 OF datapath 等, 主要是为了保证实验床的各方资源保证。该文章的不足是没有对其方法进行评估。

Ongar 等人[17]提出一个在 SDN 网络中的集中式管理与编排框架, 为实时多媒体应用实现区分的网

络服务,以达到服务等级约束(Service Level Agreements, SLA)。他们的主要贡献包括以下几个方面。首先,该文定义了一个扩展的 QoS 架构完成了标准 SDN 形式与其解决方案的无缝整合。其次,他们利用 SDN 能力在有线与无线的整合环境下为多媒体应用提供有质量的网络服务支撑。

以上的相关研究表明,目前关于 QoS 的研究仅仅针对于某条路径或者某条流量,没有从全局的角度实现流量 QoS 调度。

## 5. 关键技术分析

### 5.1. 链路状态收集技术

为了实现 QoS 策略和流量感知的路由,一个重要的前提是需要实时地收集全局网络的流量状态信息,例如传输延迟、流的带宽、每条链路的报文丢包率。任何一个 QoS 路由算法的性能是与网络状态信息获取的精确程度相关,即流量状况信息越精确,QoS 路由算法实现网络流量调度越精确。这种方法在传统的互联网中是不可行的,因为互联网采用的是完全分布式的体系架构。在分布式的流量感知路由计算过程中,附加的报文被用于告知相邻的网络设备其交换机和链路的状态,这对交换机来说是会增加较大的开销,而且这些附加的分布式链路状态通告协议也会占用部分链路带宽。此外,状态的收敛过程也相对比较缓慢。因此当前的路由协议对链路的流量感知能力较差。

在 SDN 架构下,集中的控制器可实现对全局网络的流量状态信息的获取。由原来的所有交换机共享所有的链路状态信息,转变为每个交换机向控制器发送流量状态信息。因此链路状态消息传递的复杂度从  $O(n^2)$ 变为  $O(n)$ 。但是随着网络规模的扩大,流量信息收集越发困难。根据下面的分析可知,在一个中等规模的网络中,获取全局流量信息是不可能的,会消耗大量宝贵的交换机——控制器的通信带宽,必须在提高流量收集精度与减少流量收集开销之间找到折中。

另外,由于集中的控制器需要实时地访问交换机的统计信息,因此网络状态的精确程度与信息收集的频率相关。如果部分长流占据了大部分传输的流量,那么集中控制器只需要每隔几秒钟读取一次交换机的状态信息。如果不是在高性能计算的网路环境下,流量特征显示长流持续时间很短,只有几秒钟,因此要想获得比较精确的网络状态,必须更加频繁地读取交换机的状态信息。

OpenFlow 支持三类每条流的计数器,分别是报文数、字节数、持续时间。OpenFlow 为控制器提供了两类获取交换机状态信息的方式,基于推的“Push-based”和基于拉的“Pull-based”,描述如下。

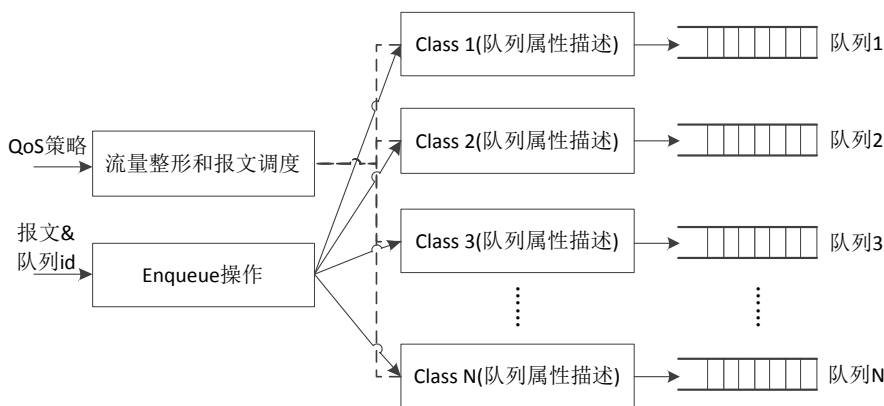
1) 基于推的收集方法:控制器通过在流建立的过程中进行决策,学习了流的状态信息。OpenFlow 协议中规定了控制器可以请求一个异步通知消息 OFPT\_FLOW\_REMOVED,当交换机删除一条流表项时,控制器可得知该消息。因此控制器可指定某条特定的流对应流规则超时。OpenFlow 既支持流表项的空闲超时机制,也支持流表项的硬超时机制。但是这种基于推的 OpenFlow 流规则删除机制不支持在表项超时之前通告控制器流的行为(例如状态等),因此基于推的状态信息统计不适合于集中的流量调度。

2) 基于拉的收集方法:控制器发送 Read-State 消息获取某条匹配流的状态统计信息。但是 DevoFlow 中 Curtis 等人指出,在理想设置下,依据 HP 的 5406zl OpenFlow 交换机,读取 16K 条精确匹配规则和 1500 条通配规则总计返回 1.3 MB 的数据,如果是每秒钟收集两次,那么会超出 17 Mbit/sec 的交换机 CPU 与控制器通信带宽的上限。所以,通常情况下,Read-State 消息只用来读取特定流或者多条通配流的统计信息,减少控制器与交换机交互的带宽。但这也阻碍了控制器获取全局网络流量的能力。

### 5.2. QoS 路由优化技术

与传统分布式的模型不同,SDN 利用全局网络视图为每条流的 QoS 控制提供了 OpenFlow 配置引擎,这种模型更加简化和高效。QoS 路由过程描述如下。





**Figure 3. SDN data plane support for QoS optimization**  
**图 3. SDN 数据平面对 QoS 优化的支持**

首先根据管理员的 QoS 参数设置作为输入，定义对应的流量信息。输入可以通过 REST API 加入到控制器中，或者通过扩展的 OpenFlow 请求，由主机发起。例如，某个用户建立一个高优先级的视频流请求，对应的分类标识是 UDP 端口号为 4428，其余用户采用普通优先级。控制器接收到请求消息后转化为对应的队列调度编码。当建立的视频流以 Packet\_in 消息发送至控制器时，控制器分析报文头，确定对应报文的等级，同时根据网络实时的拓扑，计算最有效的转发路径。

OpenFlow 协议提供的排队机制，为每个特定类型的应用程序提供带宽保证。一个 OpenFlow 交换机通过一个简单的排队机制提供有限的 QoS 支持。一个或多个队列可以被连接到一个端口，然后通过它来转发数据包。数据包转发到特定的队列将根据该队列的配置(如最小速率或最大速率)进行处理。

图 3 中描述了在 SDN 环境下，OpenFlow 交换机如何支持控制平面的 QoS 策略，以及实现基于 QoS 的报文转发。流量整形和报文调度模块接收控制发来的 QoS 策略，并将其转发为不同队列的划分，并配置其队列的属性。Enqueue 操作将报文根据其队列 id 调度至对应类别的队列中。

## 6. 总结与展望

流媒体应用，例如网络电视、视频会议和视频点播要求可预测的、稳定的网络资源，几乎没有延迟的变化，没有报文丢失，但这对于标准的尽力而为的互联网来说是不可能的。因此，过去十年，IETF 提出了很多的 QoS 框架(例如集成服务、区分服务等)，但是没有可以真正被部署和全局使用。这是因为，这些已经提出的体系架构是建立在当前互联网分布式逐条路由的体系架构，缺少对全网资源更广阔的视图。尽管 MPLS 这种隧道方式提供了部分的解决方案，但是它缺少实时地可配置性以及动态适应网络流量的能力。

OpenFlow 作为一个新的可编程网络接口，允许所有的网络服务提供商提供创新的网络服务和网络虚拟化。SDN 作为一个基于 OpenFlow 的新型网络架构，其基本概念是将路由的转发控制进行分离。为此，SDN 能够提供一种有效动态的方式实现 QoS 重配置。

由于 SDN 提供了很大的灵活性和可编程能力，打破了原来网络封闭的体系架构模型，为网络创新注入强大的动力。因此，很多传统网络中难以解决的问题都可能在 SDN 架构下得到优化。本文也为其他网络服务在 SDN 框架下的实现提供了重要的实际借鉴意义。

## 基金项目

本文受 863 课题(2015AA015603)资助。

## 参考文献 (References)

- [1] 第 36 次中国互联网络发展状况统计报告[EB/OL]. <http://wx.shenchuang.com/article/2015-07-24/1116280.html>, 2015-07-24/2016-02-08.
- [2] 吴建平, 林嵩, 徐恪, 等. 可演进的新一代互联网体系结构研究进展[J]. 计算机学报, 2012, 35(6): 1094-1108.
- [3] 冯径, 马小骏, 顾冠群. 适应 QoS 路由机制的网络模型研究[J]. 计算机学报, 2000, 23(8): 799-805.
- [4] RFC1633. Integrated Services in the Internet Architecture: An Overview [S].
- [5] RFC2475. An Architecture for Differentiated Service [S].
- [6] RFC3031. Multiprotocol Label Switching Architecture [S].
- [7] RFC2205. Resource ReSerVation Protocol (RSVP)—Version 1 Functional Specification [S].
- [8] 罗章庆. 自适应路由与队列调度策略控制技术研究[学位论文]. 北京: 北京邮电大学, 2010.
- [9] Open Network Foundation, Software-Defined Networking: The New Norm for Networks [EB/OL]. <https://www.opennetworking.org>, 2012-04-13/2016-02-08.
- [10] Mckeown, N., Anderson, T., Balakrishnan, H., *et al.* (2008) OpenFlow: Enabling Innovation in Campus Networks. *AcmSigcomm Computer Communication Review*, **38**, 69-74. <http://dx.doi.org/10.1145/1355734.1355746>
- [11] OpenFlow 1.3 Release. [http://www.openflowswitch.org/wk/index.php/OpenFlow\\_v1.3](http://www.openflowswitch.org/wk/index.php/OpenFlow_v1.3)
- [12] Narisetty, R.R., Dane, L., Malishevskiy, A., *et al.* (2013) OpenFlow Configuration Protocol: Implementation for the of Management Plane. 2013 *Second GENI Research and Educational Experiment Workshop*, Salt Lake City, 20-22 March 2013, 66-67. <http://dx.doi.org/10.1109/GREE.2013.21>
- [13] Kim, W., Sharma, P., Lee, J., *et al.* (2010) Automated and Scalable QoS Control for Network Convergence. *Proceedings of Internet Network Management Workshop/Workshop on Research on Enterprise Networking*, 1.
- [14] Egilmez, H.E., Dane, S.T., Bagci, K.T., *et al.* (2012) Open QoS: An OpenFlow Controller Design for Multimedia Delivery with End-to-End Quality of Service over Software-Defined Networks. *Signal & Information Processing Association Summit and Conference*, Hollywood, 3-6 December 2012, 1-8.
- [15] Ishimori, A., Farias, F., Cerqueira, E., *et al.* (2013) Control of Multiple Packet Schedulers for Improving QoS on OpenFlow/SDN Networking. 2013 *Second European Workshop on Software Defined Networks (EWSDN)*, Berlin, 10-11 October 2013, 81-86. <http://dx.doi.org/10.1109/EWSDN.2013.20>
- [16] Sonkoly, B., Gulyas, A., Nemeth, F., *et al.* (2012) On QoS Support to Ofelia and OpenFlow. *European Workshop on Software Defined Networking*, Darmstadt, 25-26 October 2012, 109-113. <http://dx.doi.org/10.1109/EWSDN.2012.26>
- [17] Ongaro, F., Cerqueira, E., Foschini, L., *et al.* (2015) Enhancing the Quality Level Support for Real-Time Multimedia Applications in Software-Defined Networks. 2015 *International Conference on Computing, Networking and Communications (ICNC)*, Garden Grove, 16-19 February 2015, 505-509. <http://dx.doi.org/10.1109/ICNC.2015.7069395>

### 再次投稿您将享受以下服务:

1. 投稿前咨询服务 (QQ、微信、邮箱皆可)
2. 为您匹配最合适的期刊
3. 24 小时以内解答您的所有疑问
4. 友好的在线投稿界面
5. 专业的同行评审
6. 知网检索
7. 全网络覆盖式推广您的研究

投稿请点击: <http://www.hanspub.org/Submission.aspx>