

# 一种基于SDN网络的QoS路由选择方案

孔祥彬 沈苏彬 李 莉

(南京邮电大学 计算机学院、软件学院 江苏 南京 210003)

**摘要:** 为了解决IP网络对业务流进行QoS路由选择时存在链路状态信息收集困难及算法复杂度较高等问题,利用SDN网络控制器对网络集中控制的优势,通过OpenFlow协议获取基础设施层的链路状态信息及SDN控制器提供集中式的QoS路由服务,为业务流选择具有QoS约束的路由。涉及到的QoS度量参数为带宽,QoS路由选择方案基于SDN网络设计链路上已用带宽的收集方法,为QoS路由算法提供准确的链路状态信息,并将业务流提出的带宽约束要求转换成链路代价。QoS路由方案根据链路代价使用基于Dijkstra算法的QoS路由算法为业务流选择满足带宽约束的QoS路由。在基于Floodlight控制器以及Mininet网络仿真工具搭建的实验平台上进行了验证,结果证明了该QoS路由方案的可行性。

**关键词:** 软件定义网络; 带宽; QoS路由; OpenFlow

中图分类号: TP393

文献标识码: A

文章编号: 1673-629X(2018)02-0102-05

doi: 10.3969/j.issn.1673-629X.2018.02.023

## A QoS Routing Scheme Based on Software-defined Networking

KONG Xiang-bin, SHEN Su-bin, LI Li

(School of Computer Science & Technology, School of Software, Nanjing University of  
Posts and Telecommunications, Nanjing 210003, China)

**Abstract:** In order to solve the problem that the link state information is difficult to collect and the complexity of the algorithm is high when the IP network selects a QoS routing for the service flow, we take advantage of the centralized control of the network by using SDN controller and obtain the link state information of the infrastructure layer through the OpenFlow protocol and provide the centralized QoS routing service to select the route with QoS constraints for the service flow. The QoS metric parameters are bandwidth. QoS routing scheme design the method of collecting bandwidth on the link to provide accurate link state information for QoS routing algorithm. Then, the bandwidth constraint requirements proposed by the service flow are converted into link cost. The QoS routing scheme uses the QoS routing algorithm based on the Dijkstra algorithm to select the QoS route satisfying the bandwidth constraint according to the link cost. The feasibility of QoS routing scheme is verified by the experimental platform based on the Floodlight controller and Mininet network simulation tool.

**Key words:** software-defined networking; bandwidth; QoS routing; OpenFlow

## 0 引言

在IP网络中,网络设备变得越来越难以配置和安装,从IP网络中添加或移动一个设备需要耗费许多成本,信息技术(information technology, IT)人员也需要重新配置交换机、路由器并更新访问控制列表(access control list, ACL)、虚拟局域网(virtual local area network, VLAN)和其他一些机制<sup>[1]</sup>。IP网络最初设计的目标只是为了实现端到端数据传送,现在已经很难满足企业或用户日益增加的复杂网络需求。此外,在提供和管理包含多种技术的大型多供应商网络所涉及的

运营成本也在不断增加,而运营的收入却在减少。

为了解决上述问题,工业界提出了软件定义网络(software defined networking, SDN)的概念<sup>[2]</sup>。SDN是一种分离网络控制与报文转发功能的、动态、可适配的体系结构<sup>[3]</sup>,这种逻辑集中式控制有助于网络资源的统一管理与分配,而转发层面只进行数据的转发。编程人员能够通过可编程接口实现对网络设备的管理和网络设备的报文转发控制,通过这种更加方便灵活的网络管理,从而达到简化网络运维、灵活管理调度的目标。因此,研究新型的网络架构软件定义网络具有

收稿日期: 2017-03-13

修回日期: 2017-07-20

网络出版时间: 2017-10-19

基金项目: 国家自然科学基金(61502246); 江苏省科技计划项目(未来网络前瞻性研究项目)(BY20130951108)

作者简介: 孔祥彬(1990-),男,硕士研究生,研究方向为计算机网络; 沈苏彬,博士生导师,CCF会员(E200005482S),研究方向为计算网络、下一代电信网及网络安全。

网络出版地址: <http://kns.cnki.net/kcms/detail/61.1450.TP.20171019.1626.078.html>

重要意义。

IP网络报文在进行信息传递时会涉及到路由选择的问题,为业务流选择满足QoS度量参数约束的路由一直是学术界和工业界研究的重点问题。IP网络的路由协议(如RIP、OSPF)只针对跳数计算最短路由<sup>[4]</sup>。随着业务对服务质量(quality of service, QoS)<sup>[5]</sup>的要求越来越高,需要同时考虑多个QoS度量参数进行路由选择。当涉及到两个及以上互相独立的QoS度量参数的路由时,传统网络出现的问题主要有<sup>[6]</sup>:路由信息不准确、算法复杂度高且选路效率较低,在分布式路由中还会产生回路问题。在具有集中式路由特点的SDN网络中,控制器通过全局网络拓扑为业务流选择路由,与IP网络源路由不同的是,控制器能够实时收集链路信息,保证了链路信息的准确性,为设计高效、快速的启发式算法提供了有力保证。

文中通过SDN网络控制器实时收集网络资源,在此基础上设计并实现了基于SDN网络的QoS路由算法,并对算法的有效性进行了仿真测试。

## 1 相关技术分析

QoS路由<sup>[7]</sup>是指根据网络中的可用资源和业务流对QoS的请求来决定流的路由的机制。QoS可用丢包率、延迟、延迟抖动、瓶颈带宽等基本性能指标来描述。随着IP网络的发展,视频会议、可视电话等多媒体业务不断增加,这类实时业务需要足够的带宽以保证其服务质量,因此,针对实时类业务流对带宽提出的约束问题进行QoS路由算法的设计。

在IP网络中,路由<sup>[8]</sup>一般由路由算法和路由协议构成,而路由协议用来收集网络状态信息,从而为QoS路由算法提供必要的状态信息。但是如果QoS路由算法<sup>[9]</sup>使用不准确的路由信息,会严重损害QoS路由算法的性能。因此,IP网络中期望在所有路由器维护最新的链路状态信息,随着网络规模的增大,频繁的链路状态更新需要消耗大量的信令开销。在SDN网络中,OpenFlow协议可以主动或被动地收集链路状态信息。文献[10]即是利用控制器与交换机之间的packet-in和Flow-Removed消息被动地收集链路信息,虽然控制器与交换机之间的通信开销较小,但是如果网络中存在持续时间较大的流量时,该方法则无法提供准确的链路信息。文献[11]中的OpenSample使用sFlow采样的方式收集基础设施层的链路信息,但数据的准确性受采样率的限制,如果为了提高准确率则同样会增加网络设备的CPU负载。为了能够提供准确的链路状态信息,文中选择OpenFlow协议以主动的方式采集链路状态信息,即使用StatisticsRequest消息以轮询的方式收集链路状态信息。

针对多个QoS度量参数提出约束要求称为多约束路径(multi-constrained path, MCP)路由问题。在MCP问题中,每条链路对应 $m$ 个约束条件,一条满足 $m$ 个约束条件的路径成为可行路径,网络拓扑中可能存在多条满足约束条件的路径,在多条可行路径中选择一条最小长度的路径称为多约束最优路径(multi-constrained optimal path, MCOP)问题。MCOP问题属于NP难度问题,MCOP问题中典型的一个子问题为延迟约束代价最小问题(delay constrained least cost, DCLC)。为了降低算法复杂度,根据求解问题类型和求解方法,QoS路由算法可分为<sup>[12]</sup>:多项式非启发类、伪多项式启发类、探测类、限定QoS度量类、花费函数和概率求解类。

文献[13]为了保证视频的服务质量,为业务流考虑了三个QoS度量参数,其中将延迟作为受限约束,将链路负载和延迟综合成链路代价,针对这个受限最短路径(restricted shortest path, RSP)问题采用拉格朗日松弛(Lagrangian relaxation)算法来为视频流选择QoS路由。文献[14]只考虑链路剩余带宽和延迟,将链路剩余带宽作为链路代价,延迟作为约束条件进行选路,当传输的业务流对延迟比较敏感时,采用拉格朗日算法计算路由。当传输的是数据类非实时业务流时,其只考虑链路剩余带宽,将链路剩余带宽作为链路代价并使用Dijkstra算法实现业务流的QoS路由选择。文献[15]基于SDN技术提出的HiQoS系统引入了多路径路由技术,其通过修改最短路径路由算法,使其能够计算满足QoS度量参数约束的多条路径并利用哈希表保存路径,控制器以一定的周期检查这些路径的链路状态,当需要为业务流选择路径时,控制器根据带宽使用情况和队列状态等选择一条最优的路径给相应的业务流。

## 2 问题分析与方案设计

### (1) 问题分析。

根据上一节对QoS度量参数收集、QoS路由算法的分析,可以看出IP网络信息收集需要耗费大量的信令开销来收集网络状态信息。当网络规模增大时,链路上的信令开销也会呈现指数级增大,并且由于信令延迟问题导致链路信息不准确,对QoS路由算法选路效率影响较大。因此,利用OpenFlow协议以主动的方式实时收集网络链路状态信息;此外,在IP网络中QoS路由算法实现起来相对困难,算法复杂度较高,很难为业务流选择端到端的QoS路由。而SDN网络可以聚集出基础设施层的网络拓扑且能够实时收集链路状态,控制器也可以提供集中式的路由服务。当前SDN网络中QoS路由方案是研究的热点问题,因此针

对业务流对带宽提出的约束要求,设计了 QoS 路由算法以满足业务流的 QoS 需求。

### (2) 基于 SDN 网络的 QoS 路由方案。

基于 SDN 网络的 QoS 路由方案的总体框架如图 1 所示。

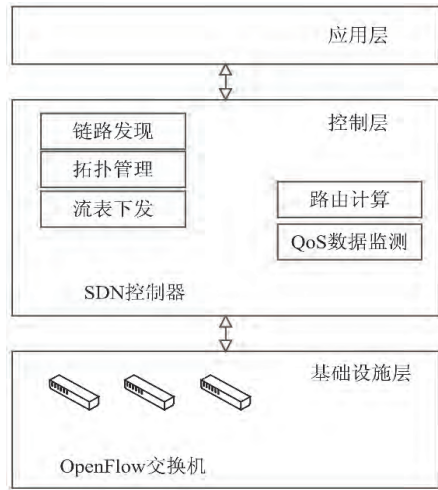


图 1 方案总体设计框架

该方案的设计依赖于 SDN 控制器提供的网络基础功能,如链路发现、拓扑管理、流表下发等基础功能,其中链路发现使用链路层发现协议(link layer discovery protocol, LLDP)实现底层网络链路的发现与更新,拓扑管理用于构建 SDN 网络的拓扑并实时更新网络拓扑,流表下发用于将路由策略下发给 OpenFlow 交换机。在此基础上详细阐述了 QoS 数据监测方法、QoS 路由算法,其中主要阐述了剩余带宽监测方法、基于 Dijkstra 的 QoS 路由算法,为业务流选择满足带宽约束要求的 QoS 路由。

### (3) QoS 数据监测设计。

QoS 数据监测是对链路上流量信息的收集以为 QoS 路由算法提供实时、准确的链路状态信息。在 SDN 网络中能够以被动方式或主动方式进行流量信息的收集。SDN 网络可以使用 OpenFlow 协议以主动的方式收集链路状态信息,保证流量收集的准确性。QoS 数据监测则是基于 SDN 网络,主要用于剩余带宽的收集。控制器通过解析交换机发送的 StatisticsReply 消息获取并解析端口流量的统计信息,通过不同时间端口接收字节数和发送字节数的不同来计算链路当前流量占用的带宽,最终通过端口最大带宽减去端口当前流量占用的带宽得到链路剩余带宽。

### (4) QoS 路由算法设计。

文中将带宽作为 QoS 度量参数,主要考虑业务流对带宽提出的约束要求,因此,将带宽转换为链路代价并用  $c_{ij}$  表示,  $B_{ij}$  表示链路  $ij$  的剩余带宽,  $B_{\min}$  表示业务流提出的最小带宽要求。剩余带宽越大,越能满足业务流的带宽要求,如果剩余带宽小于最小带宽要

求,则无法满足业务流提出的带宽约束要求。如果剩余带宽  $B_{ij}$  大于业务流提出的带宽约束  $B_{\min}$ ,则此时链路代价  $c_{ij}$  的值为链路剩余带宽与业务流提出约束要求带宽的差值;如果剩余带宽小于业务流提出的带宽约束要求,此时链路代价为负无穷。通过以上链路代价的初始化,从而为业务流选路时满足业务流对带宽提出的约束要求。

网络拓扑图用  $G(N, A)$  来描述,其中  $N$  表示网络中的节点集合,  $A$  表示网络中的链路集合。  $\text{Link}(i, j)$  表示一条从节点  $i$  到节点  $j$  的有向链路。  $P_{st}$  表示从源节点  $s$  到目的节点  $t$  的所有满足带宽约束的可行路径,对于其中的路径  $P \in P_{st}$ ,定义满足带宽约束的可行路径  $P$  的总代价  $f_c$ ,为路径  $P$  上每条链路代价的和。

基于以上的链路代价,使用 Dijkstra 算法为业务流选择满足带宽约束要求且代价最大的路径,算法的伪代码如下所示:

Dijkstra( $G, s, t, c$ ) //在图  $G$  中,  $s$  为源节点,  $t$  为目的节点,  $c$  为链路代价

任意的  $v$  属于节点集合  $N$ ,  $\text{dist}[v] := -\infty$ ,  $\text{prev} := \text{null}$ ; //初始化总代价为负无穷,前驱节点为空

$\text{dist}[s] := 0$ ; //设置源节点的总代价为 0

$S := \Phi$  //  $S$  表示保留了已知的所有  $\text{dist}[v]$  的值已经是最大代价路径的值的顶点,集合  $S$  初始状态为空

$Q := N \setminus Q$  //  $Q$  保留了其他所有顶点,其初始状态为图  $G$  中所有的顶点集合  $N$

while( $Q \neq \Phi$ ) //当节点集  $Q$  不为空时进行循环操作

$u := \text{Extract\_Max}(Q)$  //  $Q$  中  $\text{dist}[u]$  最大的顶点并将  $u$  从  $Q$  中删除

$S.append(u)$  //将节点  $u$  加入到  $S$  集合中

if( $u = t$ ) end; //如果节点  $u$  为目的节点,算法结束

For each edge outgoing from  $u$  as  $(u, v)$

$\text{alt} = \text{dist}[u] + c(u, v)$ ; //遍历  $u$  的所有相邻节点  $v$  并计算总代价

if( $\text{alt} > \text{dist}[v]$ ) //如果计算的总代价比原来代价大,则更新  $v$  的总代价

$\text{dist}[v] = \text{alt}$ ; //更新节点  $v$  的总代价

$\text{prev}[v] = u$ ; //更新  $v$  的前驱节点为  $u$

end //算法结束

以上基于 Dijkstra 算法的 QoS 路由算法根据链路代价为业务流选择 QoS 路由,而链路代价  $c_{ij}$  考虑到了业务流对带宽的最小约束要求。此外,基于 SDN 网络的 QoS 算法较 IP 网络 QoS 路由算法的优势为数据平面交换机之间不需要信令开销。相较于 IP 网络 QoS 源路由,利用 SDN 控制器能够实现实时的链路状态更新,聚集出有效的链路状态信息而不会产生陈旧信息,选路效率较高;算法能够基于聚集的链路状态信息利用 SDN 控制器进行集中式路由。因此利用 SDN 网络实现 QoS 路由选择技术使算法具有易扩展、适应不同

网络拓扑等特点。

### 3 方案实现

QoS路由选择的实现主要基于开源控制器 Floodlight, 其中依赖于 Floodlight 控制器实现了链路发现模块、拓扑管理模块、流规则下发模块。链路发现模块实现了底层网络链路的发现与更新, 拓扑管理模块实现了网络拓扑的构建及更新, 流规则下发模块实现了 SDN 网络控制器将路由策略下发给 OpenFlow 交换机。而 QoS 数据监测主要实现底层网络剩余带宽的收集。QoS 路由算法主要实现业务流的 QoS 路由选择。

#### (1) QoS 数据监测实现。

本节主要实现对剩余带宽的监测。基于 Floodlight 开源控制器设计了 DataMonitor 模块, 该模块需要调用 IFloodlightService、IFloodlightModule、IOFMessageListener 提供的服务。其中 IFloodlightService 是 Floodlight 能够提供服务的模块的基本接口, IFloodlightModule 定义了 Floodlight 能够加载的模块的接口。Floodlight 控制器通过 IOFMessageListener 来监听 OpenFlow 消息。

DataMonitor 模块中的主要类和接口如下:

IDataMonitor 接口为 DataMonitor 中的一个服务, 继承 module 模块即 Floodlight 能够提供服务的模块的基本接口 IFloodlightService, 主要方法为 getLink, 用于计算剩余带宽。通过 IDataMonitorService 接口向 QoS 路由模块提供剩余带宽信息。

DataMonitor 类主要实现了剩余带宽的计算, 实现了 IFloodlightModule、IOFMessageListener 两个接口中的 getModuleServices、getServiceImpls 等方法。IFloodlightModule 标识了 DataMonitor 模块是 Floodlight 的一个模块, 在 Floodlight 启动过程中, 该类以模块形式加载。IOFMessageListener 接口用于监听 Floodlight 控制器与 OpenFlow 交换机之间的消息, 如 Packet-in、Flow-Mod、Stat-Reply 等。DataMonitor 类除了实现 getLink 方法以外, 还实现了 receive 方法, 其主要是对 Stat-Reply 消息的处理。方法 Precess\_stat\_reply 功能是对 OpenFlow 消息 Stat-Reply 进行处理, 创建一个 Flow\_Table\_Entry, 并将这个流表项作为 getLink 方法的一个参数。

计算剩余带宽的主要方法为 getLink, 在 DataMonitor 中实现。首先通过 receive 方法处理 OpenFlow 消息, 然后通过 Precess\_stat\_reply 方法将统计的消息转化成一个流表项供 getLink 方法使用。getLink 方法通过获取的当前端口的状态信息, 计算出剩余带宽。

#### (2) QoS 路由算法实现。

QoS 路由选择的实现依赖于 Floodlight 控制器, 首先在 Eclipse 中导入 Floodlight 源码和依赖包, 然后在 Eclipse 中创建 QoS 路由选择模块, 之后在 resources META-INF/services/net. Floodlight. core. module. IFloodlightModule 中指定新增模块名, 添加模块配置信息。文中路由计算模块实现调用 TopologyManager 模块类以实现全局网络拓扑的维护, 并且根据 QoS 数据监测模块提供的服务, 为业务流选择 QoS 路由; 然后调用 Floodlight 控制器的 Forwarding 模块给交换机安装流表项, 最终实现业务流的转发。

下文对 QoSRouting 模块各个接口及类进行说明, 并重点阐述了与 QoS 路由算法实现相关的方法。

其中 QoSRouting 模块主要的接口和类为: IFloodlightModule、MasterTopologyInstance 类、TopologyManager 类、IQoSRoutingService, 各个类及接口之间的关系如图 2 所示。

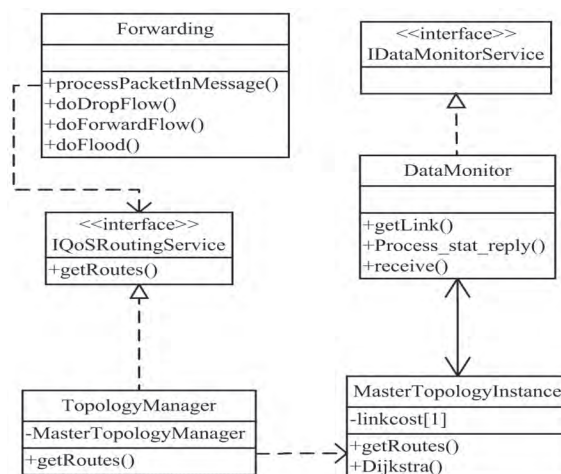


图2 QoSRouting 类及接口图

图中, DataMonitor 类实现剩余带宽的监测, MasterTopologyInstance 类实现 QoS 路由算法。其中 IQoSRoutingService 为 Floodlight 中其他模块提供服务的接口, 其中主要方法为 getRoutes。Forwarding 模块可以通过该方法获取 QoS 路由路径。TopologyManager 用于维护网络拓扑结构。MasterTopologyInstance 类为 QoS 路由模块的核心类, 完成路径计算, 其中主要方法为 Path Dijkstra( Cluster c, Map<DatapathId, set<link>> links, DatapathId src, DatapathId dst, Map<link, Integer> linkCost)。通过源节点和目的节点以及链路代价来计算具有链路代价的路径, 其中 Path 类保存了所计算路径的链路信息以及所计算出的代价, 如 Dijkstra 算法中返回的 Path 类型, 包含了路径上的链路信息和这条路径的总代价, Cluster 表示 OpenFlow 交换机簇。

### 4 实验环境搭建与结果分析

#### (1) 测试环境。



采用 Floodlight 控制器与网络仿真工具 Mininet 对 QoS 路由方案的有效性进行测试。首先使用 `java -jar target/floodlight.jar` 命令开启改进后的 Floodlight 控制器,此时 Floodlight 根据其配置文件信息加载基本模块、拓扑模块、链路发现模块在 Floodlight 控制器启动即开始监听事件,但是此时 Mininet 并没有开启, Floodlight 控制器的 UI 界面暂时没有网络设备。通过 Mininet 创建的自定义拓扑,连接 Floodlight 控制器、链路发现模块和拓扑管理模块处理 packetin 消息构建整个网络的拓扑。

实验中涉及两台物理主机,在其中一台物理主机(地址为 192.168.1.110)上安装 Floodlight 控制器作为控制层设备。在另一台物理主机(地址为 192.168.1.111)上安装 Mininet 网络仿真软件,用来构建网络拓扑。控制层和交换层物理主机上运行的环境为 ubuntu12.04,在 Mininet 上使用 iperf 网络性能测试工具发送 UDP 数据包。实验拓扑图如图 3 所示。使用 Mininet 自定义拓扑功能,将链路  $\langle s_1, s_5 \rangle$ 、 $\langle s_5, s_4 \rangle$ 、 $\langle s_4, s_3 \rangle$  上的带宽设置为 20 Mbits,其他交换机间链路带宽设置为 5 Mbits。

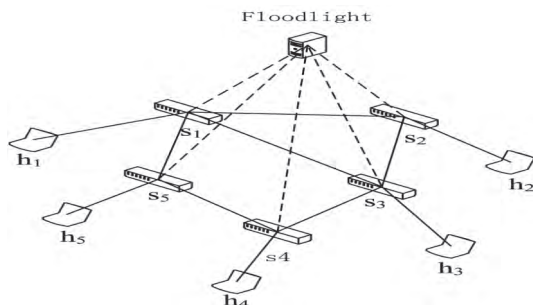


图 3 实验拓扑图

## (2) 测试步骤与结果分析。

根据实验拓扑图,设置业务流最小带宽约束为 6 Mbits,利用 iperf 网络性能测试工具进行发送 UDP 报文的测试,以验证算法的有效性。步骤如下:

① Mininet 连接最短路算法的 Floodlight 控制器,通过 Mininet 的 xterm 命令打开  $h_1$  和  $h_3$  虚拟终端,以  $h_1$  为 iperf 客户端,  $h_3$  为 iperf 服务端,  $h_1$  通过如下命令向  $h_3$  发送 UDP 报文:

```
iperf -c 10.0.0.3 -S 0x10 -u -i 5 -t 30 -b 6M
```

通过 Floodlight 控制器的 UI 界面查看交换机的流表项可知,业务流的传输路径为  $\langle s_1, s_3 \rangle$ ,并未考虑链路的任何代价,只进行最短路路由。

② 启动添加了 QoS 路由方案的 Floodlight 控制器,首先需要在 `floodlight.modules` 配置参数后添加 QoS Routing 模块名,以让模块在 Floodlight 启动时必须加载,然后再次运行 Ant 命令重新编译版本,最后执行 `floodlight.sh` 启动 Floodlight 控制器。通过 Mininet

的 xterm 命令开启  $h_1$  和  $h_3$  的虚拟终端,以  $h_1$  为 iperf 客户端,  $h_3$  为 iperf 服务端,  $h_1$  通过如下命令向  $h_3$  发送 UDP 报文:

```
iperf -c 10.0.0.3 -S 0x10 -u -i 5 -t 30 -b 6M
```

通过查看 Floodlight 控制器的 UI 界面中的交换机流表项,分析 UDP 报文的转发路径为  $\langle s_1, s_5, s_4, s_3 \rangle$ ,部分交换机的流表项如图 4 所示。

OpenFlow Version: OF\_13

**Ports (4)**

#	Link Status	TX Bytes	RX Bytes
3 (s5-eth3)	UP 10 Gbps FDX	0	0
1 (s5-eth1)	UP 10 Gbps FDX	0	0
2 (s5-eth2)	UP 10 Gbps FDX	0	0
local (s5)	UP	0	0

**Flows (5)**

Cookie	Table	Priority	Match	Apply Act
9007199254740992	0x0	1	in_port=1 eth_dst=42:10:6c:68:1b:46 eth_src=c6:e1:71:86:57:0d eth_type=0x0800 ip_proto=0x11 ipv4_src=10.0.0.1 ipv4_dst=10.0.0.3 udp_src=45687 udp_dst=5001	actions:out

图 4  $s_5$  交换机部分流表项

由自定义拓扑中链路的带宽状态可知,最短路路径  $\langle s_1, s_3 \rangle$  和次短路路径  $\langle s_1, s_2, s_3 \rangle$  的链路带宽无法满足业务流的最小带宽约束要求,而对于路径  $\langle s_1, s_5, s_4, s_3 \rangle$  的链路能够满足业务流的带宽约束要求, QoS 路由算法能够选择剩余带宽较大的路径。

## 5 结束语

基于 SDN 网络设计了 QoS 路由选择方案,利用 OpenFlow 协议收集网络状态信息,根据 SDN 控制器的集中控制优势,设计并实现了 QoS 路由算法,为有 QoS 要求的业务流选择 QoS 路由,最后对该方案进行了测试。结果验证了 QoS 路由算法的有效性。接下来的工作将对算法满足带宽约束进行更深入的测试,并且利用 SDN 网络控制器的集中控制优势,通过合理的资源分配满足业务流服务质量要求。

## 参考文献:

- [1] 左青云,陈鸣,赵广松,等.基于 OpenFlow 的 SDN 技术研究[J].软件学报,2013,24(5):1078-1097.
- [2] Open Networking Foundation. Software-defined networking: the new norm for networks [EB/OL]. 2012-04-01. <https://www.opennetworking.org/sdn-resources/sdn-library/white-papers>.
- [3] 沈苏彬.软件定义联网的建模与分析[J].南京邮电大学学报:自然科学版,2014,34(3):1-9.
- [4] 林玉侠,朱慧玲,马正新,等.QoS 路由度量参数的选择问

(下转第 134 页)

的准确率的变化的情况。实验参数的设置与实验 1 相同,结果如图 3 所示。

由图 3 可见,采用提出的信任度模型预测不诚实实体的准确率比 EigenTrust 信任度模型收敛的速度快。

## 4 结束语

分析了现有访问控制模型,在已有研究的基础上提出改进贝叶斯信任度计算及对模型构建域间访问控制模型。信任度计算方法有所改进,一定程度上提高了检测的收敛速度,有利于根据信任度的变化进行模型的调整和重构,具有效率高、速度快的特点。

## 参考文献:

- [1] 张宏, 贺也平, 石志国. 基于周期时间限制的自主访问控制委托模型[J]. 计算机学报, 2006, 29(8): 1427-1437.
- [2] 陈俊欣, 张凤荔, 刘渊. 基于角色的空间信息强制访问控制模型研究[J]. 计算机应用研究, 2016, 33(7): 2170-2174.
- [3] 熊厚仁, 陈性元, 杜学绘, 等. 基于角色的访问控制模型安全性分析研究综述[J]. 计算机应用研究, 2015, 32(11): 3201-3208.
- [4] 邓集波, 洪帆. 基于任务的访问控制模型[J]. 软件学报, 2003, 14(1): 76-82.
- [5] 熊金波, 姚志强, 马建峰, 等. 基于行为的结构化文档多级访问控制[J]. 计算机研究与发展, 2013, 50(7): 1399-1408.
- [6] JOSHI J B D. A generalized temporal role based access control model for developing secure systems [D]. USA: Purdue

University 2013.

- [7] 彭维平,刘雪贞,郭海儒,等.基于信任度的跨域安全访问控制模型研究[J].计算机应用研究,2016,33(6):1797-1796.
- [8] KATT B,ZHANG X W,BREU R,et al.A general obligation model and continuity: enhanced policy enforcement engine for usage control[C]//ACM symposium on access control models and technologies.Estes Park,CO,USA:ACM,2008:683-695.
- [9] 王小明,赵宗涛.基于角色的时态对象存取控制模型[J].电子学报,2005,33(9):1634-1638.
- [10] 刘庆云,沙泓州,李世明,等.一种基于量化用户和服务的大规模网络访问控制方法[J].计算机学报,2014,37(5):1195-1205.
- [11] YUAN E,TONG J.Attributed based access control (ABAC) for Web services[C]//IEEE international conference on web services.FL,USA:IEEE,2005:561-569.
- [12] 房梁,殷丽华,郭宇川,等.基于属性的访问控制关键技术研究综述[J].计算机学报,2017,40(7):1680-1698.
- [13] 黄保华,贾丰玮,王添晶.云存储平台下基于属性的数据库访问控制策略[J].计算机科学,2016,43(3):167-173.
- [14] 李凤华,王巍,马建峰,等.基于行为的访问控制模型及其行为管理[J].电子学报,2008,36(10):1881-1890.
- [15] 李凤华,王彦超,殷丽华,等.面向网络空间的访问控制模型[J].通信学报,2016,37(5):9-20.
- [16] 史姣丽,黄传河,王晶,等.云存储下多用户协同访问控制方案[J].通信学报,2016,37(1):88-99.
- [17] 王晶,黄传河,王金海.一种面向云存储的动态授权访问控制机制[J].计算机研究与发展,2016,53(4):904-920.

(上接第 106 页)

- 题研究[J].电信科学, 2003, 19(7): 22-27.
- [5] 杨胜文, 史美林. 一种支持 QoS 约束的 Web 服务发现模型[J]. 计算机学报, 2005, 28(4): 589-594.
- [6] KUIPERS F A. Quality of service routing in the internet: theory, complexity and algorithms [M]. [s.l.]: IOS Publisher, 2004.
- [7] 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.
- [8] 闵应骅. 计算机网络路由研究综述[J]. 计算机学报, 2003, 26(6): 641-649.
- [9] MASIP-BRUIN X, YANNUZZI M, DOMINGO-PASCUAL J, et al. Research challenges in QoS routing [J]. Computer Communications, 2006, 29(5): 563-581.
- [10] YU C, LUMEZANU C, ZHANG Y, et al. FlowSense: monitoring network utilization with zero measurement cost [C]//International conference on passive and active network measurement. Berlin: Springer, 2013: 31-41.
- [11] SUH J, KWON T T, DIXON C, et al. OpenSample: a low-latency

- tency sampling-based measurement platform for commodity SDN [C]//International conference on distributed computing systems. Madrid ,Spain: IEEE 2014: 228-237.
- [12] 崔 勇,吴建平,徐 恪,等.互联网络服务质量路由算法研究综述[J].软件学报 2002 ,13( 11) : 2065-2075.
- [13] EGILMEZ H E ,DANE S T ,BAGCI K T ,et al.OpenQoS: An OpenFlow controller design for multimedia delivery with end-to-end quality of service over software-defined networks [C]//Signal & information processing association summit and conference.Hollywood ,California ,USA: IEEE 2012.
- [14] CUI H Y ,ZHU Y ,YAO Y ,et al.Design of intelligent capabilities in SDN [C]//International conference on wireless communications , vehicular technology , information theory and aerospace & electronic systems. Aalborg ,Denmark [s. n. ] 2014: 1-5.
- [15] YAN Jinyao ,ZHANG Hailong ,SHUAI Qianjun.HiQoS: an SDN-based multipath QoS solution [J].China Communications 2015 ,12( 5) : 123-133.