****

**第三届全国高校软件定义网络（SDN）**

**创新应用大赛复赛技术报告**

**QoS over LLDP——基于 LLDP 协议的捎带**

**式服务质量采集机制**

学 校：西南民族大学

队伍名称：网 麟

队 员：周 成 吴俊磊 彭延阳

徐昊宸 邱 夏 谢 梅

指导老师：陈 曦

联 系 人：周 成

联系电话: 18482104565

目录

[一、简介 3](#_Toc459241454)

[内容 3](#_Toc459241455)

[拟解决问题 4](#_Toc459241456)

[设计思路 4](#_Toc459241457)

[二、应用场景介绍 4](#_Toc459241458)

[场景一 4](#_Toc459241459)

[场景二 5](#_Toc459241460)

[三、方案特色和创新 6](#_Toc459241461)

[四、具体设计 7](#_Toc459241462)

[（一）整体思路 7](#_Toc459241463)

[（二）解决方案 8](#_Toc459241464)

[（三）预期目标 9](#_Toc459241465)

[五、实现过程 9](#_Toc459241466)

[（一）程序流程 9](#_Toc459241467)

[（二）预期目标实现 11](#_Toc459241468)

[1.自发式实现 11](#_Toc459241469)

[2.触发式实现 14](#_Toc459241470)

[（三）实验设计 16](#_Toc459241471)

[1.性能实验 16](#_Toc459241472)

[2.应用实验 18](#_Toc459241473)

[附录： 21](#_Toc459241474)

[Open vSwitch部分 21](#_Toc459241475)

[Floodlight部分 23](#_Toc459241476)

[参考文献 25](#_Toc459241477)

# 一、简介

## 内容

QoS 在网络中利用各种技术为网络通信提供更好的服务能力,解决网络延迟和阻塞等问题,确保重要的业务量不受延迟或者丢弃。而要SDN部署 QoS 策略之前要先对网络中 QoS 信息进行采集,目前广泛使用的Floodlight 控制器上并没有采集服务质量信息的机制。

本项目拟通过 TLV (Type/Length/Value)扩展标准的LLDP协议,使南向接口具备服务质量采集、传递的能力。标准的LLDP协议可以通过交换机流表匹配失败后发送的Packet\_in,让控制器快速发现SDN网络的拓扑结构,从而掌握网络的全局视图。这一过程在SDN网络的初始化阶段广受支持。若能利用这一流程,将QoS信息通过TLV写入到LLDP分组中,则可让控制器快速获取带服务质量信息的全局网络拓扑结构,从而提供良好的QoS支持。本项目将这一过程称之为QoS over LLDP。QoS over LLDP的优势在于,在拓扑结构发现的过程中,捎带式地进行了服务质量信息的发现,减少了专用QoS发送带来的网络流量,并同拓扑结构关联起来,构成可感知服务质量的拓扑结构。由于一般的交换机均支持LLDP协议,也提高了软件的兼容性和可移植性。

## 拟解决问题

现亟需解决的问题：一是在Floodlight控制器中设置好LLDP报文中TLV（Type/Length/Value）的结构，利用其进行QoS信息的捎带;二是修改Open vSwitch源代码使之能够对收到的LLDP数据包，填写自身的QoS信息。由于采用了LLDP 作为 QoS 采集的承载协议,我们将实现上述二方面功能的方案称之为 QoS over LLDP。

## 设计思路

控制器向交换机下发包含 LLDP 报文的 Packet\_out 消息,交换机先将对应端口的QoS信息写入LLDP报文中设计好的TLV字段，然后在相应的端口进行转发,相邻的交换机收到 LLDP 报文后由于没有相应的流表项,就将报文装在 Packet\_in 消息中发给控制器,实现交换机邻接状态信息的获取,同时也控制器获取了交换机相应端口上的QoS信息。

# 二、应用场景介绍

## 场景一

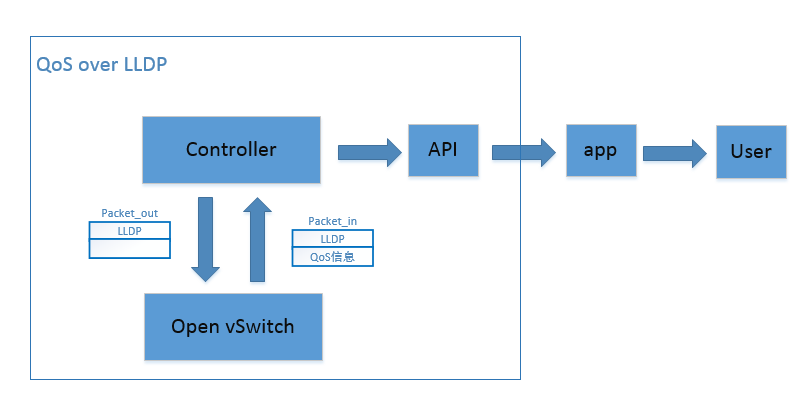
使用者在SDN网络中要想实时获得每个交换机端口的带宽、时延、抖动、丢包率等QoS信息时，可以在应用中对QoS over LLDP接口调用，从而使得控制器和上层应用（app）知悉整个网络的全局拓扑和QoS状态。如图1。

图 1 用户调用QoS over LLDP的API查询QoS信息

## 场景二

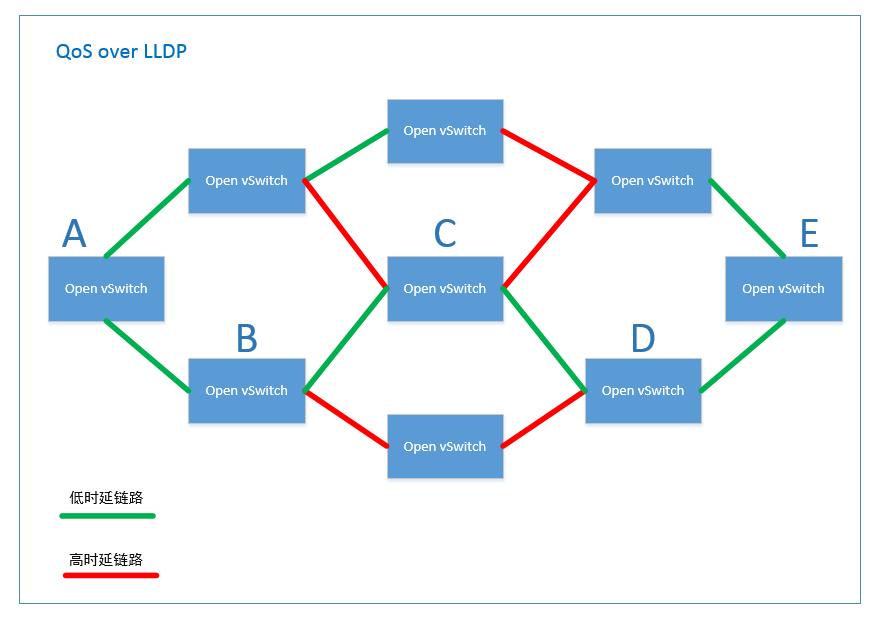
在SDN网络中实现基于QoS的最优路径发现和路由。如图2，A到E有多条路径可选，我们先从QoS over LLDP中获取交换机的时延信息，红线表示高时延链路，绿线表示低时延链路，这样一来我们就可以获取一条总时延比较低的路径A→B→C→D→E。上述应用场景的阐述同样适用于基于其他服务质量指标（如带宽等）的最优路径发现和路由。

图 2 在QoS over LLDP中部署基于QoS的路由

# 三、方案特色和创新

（一）目前,在广泛使用的 Floodlight 控制器中并没有 QoS 信息采集的机制，开发和研究人员难以在控制器上基于QoS信息部署 网络策略。若控制器在部署策略之前就可以知道整个网络任意局部的QoS信息，那么基于QoS信息部署网络策略时,就有实时数据可以进行参考，这样一来 SDN 网络部署 QoS 策略就显得更简便,更有效率。

（二） 在使用 LLDP 报文传递 QoS 信息时, 仅仅在原来的 LLDP 数据包中增加了几十个字节用来记录网络每个接口的QoS信息，对原有网络的开销没有显著的影响。

（三）OpenFlow 协议包含 LLDP 数据包处理模块,从而 QoS

over LLDP 可方便地移植到各种控制器的环境中(不局限于 Floodlight)。如果在没有对 QoS 处理模块的控制器中，最坏的情况仅是缺乏的 QoS的 TLV(Type/Length/Value) 的处理机制,但不会与控制器的基本工作流程冲突，因为所有控制器都支持 LLDP 处理。

（四）我们添加了处理QoS TLV 的模块,使该控制器可以从定制的 LLDP 报文中获取 QoS信息。由于处理的流程模块化，这样一来方便其他开发者调用和改进这个模块。

# 四、具体设计

## （一）整体思路

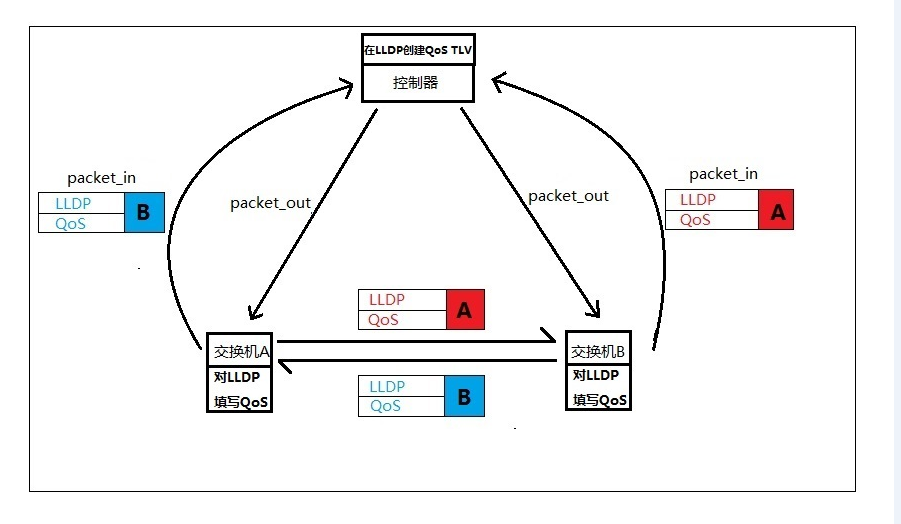
Floodlight控制器中设置好LLDP报文中TLV(Type/Length/Value)的结构——用于存放带宽、时延等QoS信息，控制器向交换机下发带有 LLDP 报文的 packet\_out 消息时,交换机将收到的 LLDP 报文的TLV字段填入自身的 QoS 信息,再将其发送到相邻的交换机,相邻的交换机收到 LLDP报文由于没有相应的流表进行匹配,就向控制器发送包含 LLDP 报文的 packet\_in 消息，这样控制器就获得了交换机的 QoS信息。整体框架如图3

图 3 整体框架

## （二）解决方案

因为LLDP 数据信息都是放在 TLV(Type/Length/Value)中的,因此我们可以把QoS 信息也放在包含LLDP 数据包中的 TLV中。为此，我们先在控制器中设置好TLV字段的格式，修改Open vSwitch源代码，实现Open vSwitch对收到Packet\_out消息后对相应LLDP报文的TLV字段写入QoS信息的。

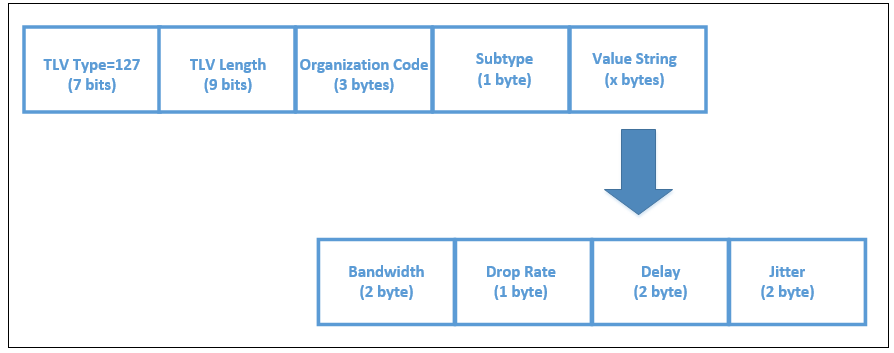
在 TLV Type 字段中,9~126 是自定义的 TLV，我们将带宽、时延、抖动等QoS信息设置成不同的TLV Type。TLV Length 字段指定该 TLV 可变长度值。 Organization Code 字段表示这是设计者定制的 TLV。 Subtype字段指定了详细的子类型,这里设定子类型有带宽、 丢包率、时延和抖动。 Value String 字段给出了真正的值。为了让接收器可以方便地解析不同的 QoS 信息,我们使用预定义的属性顺序和长度，如图4。

图 4 存放QoS信息的TLV字段

## （三）预期目标

在实现整体的设计以后，我们将创建拓扑进行抓包试验，此时抓到的带有LLDP的packet\_out包中会看到我们已经设计好的TLV字段格式且初始值固定；抓到的带有LLDP的packet\_in包中会看到TLV的值发生了改变。若此时的TLV值与我们为交换机配置的带宽、时延等值相对应，则实现了我们的设想。

# 五、实现过程

## （一）程序流程

程序流程图如图5。



图 5 程序流程图

## （二）预期目标实现

### 1.自发式实现

我们使用一个简单的拓扑进行测试，拓扑如图6，在给每个交换机配置QoS时使用随机函数，带宽、时延和抖动的值都是随机生成的，部分代码如图7。启动Floodlight控制器并建立好拓扑，控制器自动下发带有LLDP数据包Packet\_out消息，交换机填入QoS信息后发送给相邻交换机，相邻交换机向控制器发回Packet\_in消息，这个过程自动完成。

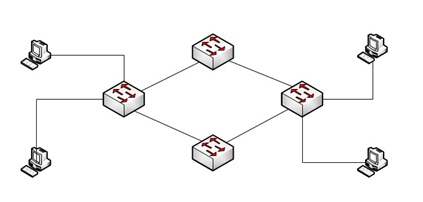


图 6 测试拓扑

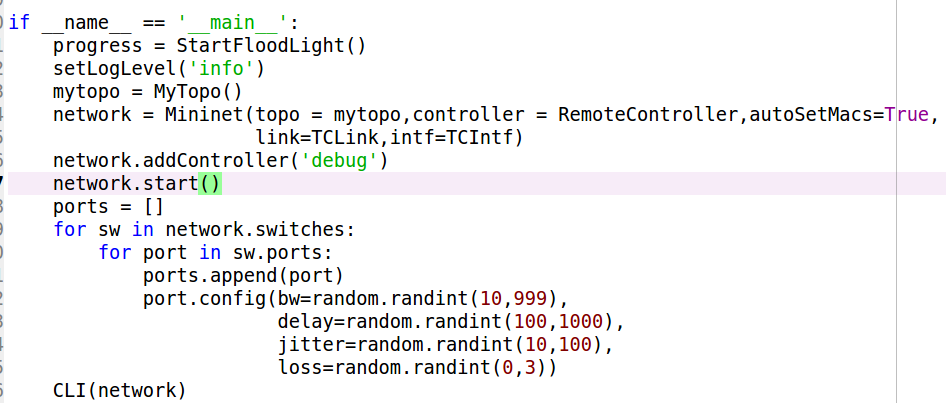


图 7 测试程序部分代码

这里我们查看的是交换机s1的3号端口s1-eth3相关的Packet\_in和Packet\_out数据包。生成拓扑后我们在Mininet下面调外部tc命令查看s1-eth3的QoS信息，如图8。

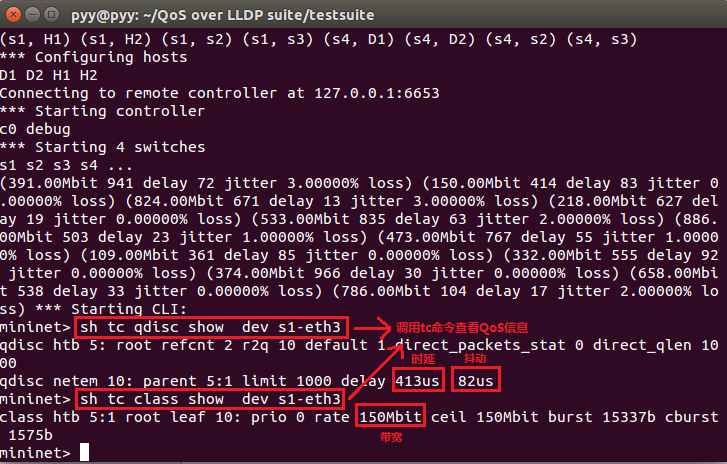


图 8 调用tc命令查看s1-eth3的QoS信息

使用Wireshark进行抓包，Packet\_out里边的LLDP包已经设定好TLV Type100~103分别表示带宽、时延、抖动和丢包率，它们的值都是默认的十六进制全f，代表尚未填写具体的QoS信息。之所以不使用全0作为空值，是考虑到某些QoS指标可能取值为0，如图9所示。packet\_in里边的LLDP包的QoS值发生了变化，表明已经利用LLDP采集并承载了相应交换机端口的服务质量信息，如图10所示。我们将变化后的值与调用tc命令查到的值进行比对，发现是一致的，其他端口的测试也是如此，故实现预期目标。

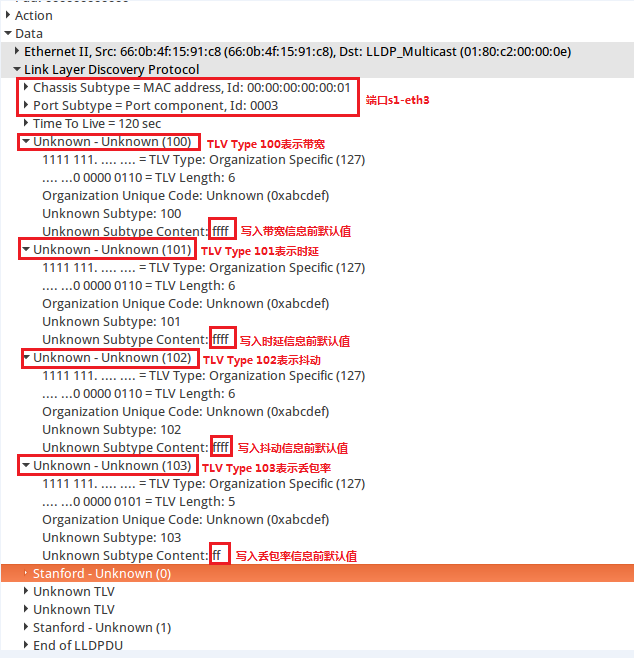


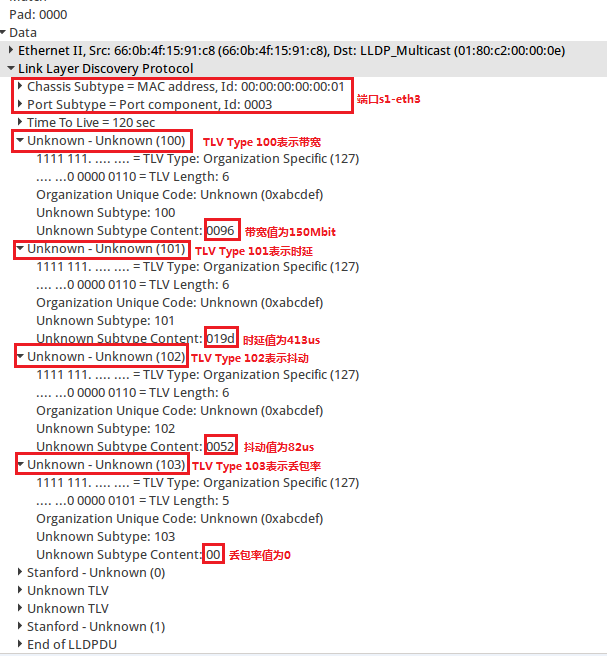
图 9 自定义TLV字段的Packet\_out

图 10 TLV中填入QoS的Packet\_in

### 2.触发式实现

上述的实现是在建立好网络以后自发实现的，我们在控制器上写好一个接口，调用这个接口以后控制器立即下发Packet\_out消息，最后LLDP包以Packet\_in消息回到控制器，整个流程与自发式一致，不同之处在于后者是调用接口触发式地实现控制器获取QoS信息，这也增加了QoS over LLDP的灵活性。调用API获取QoS信息流程如下：

先将Wireshark中将LLDP数据包清空 ，如图11 。

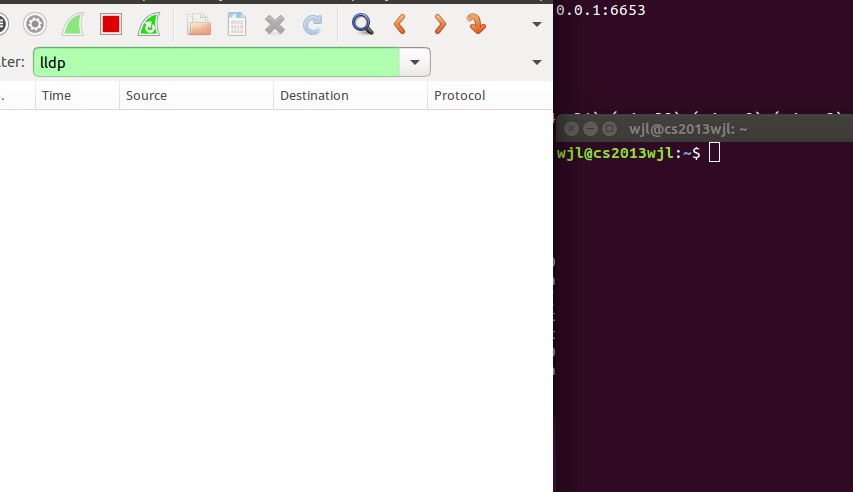


图 11 在Wireshark中将原有的LLDP包清空

在终端输入http://127.0.0.1:8080/wm/lldp/trigger/json ，控制器开始下发Packet\_out消息，可以看到Wireshark出现了LLDP数据包，终端中QoS信息也显示出来，如图12。

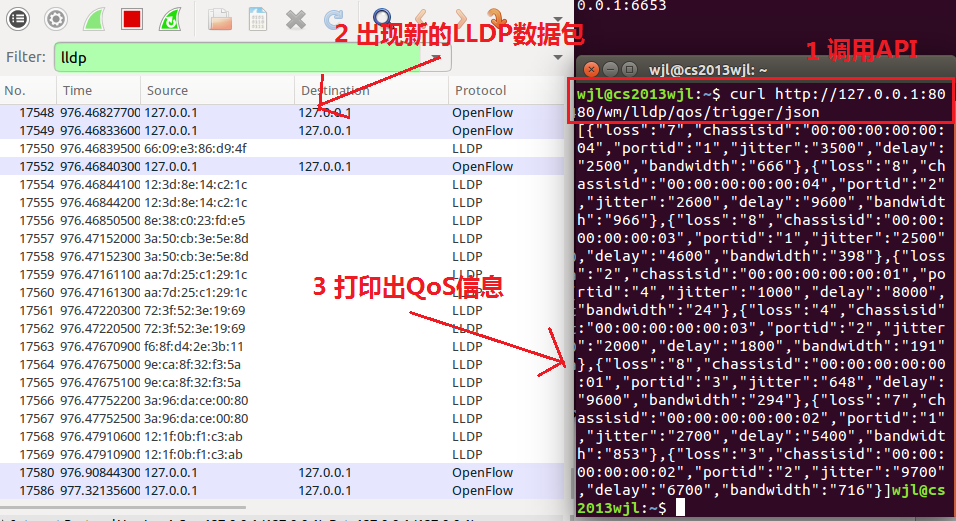


图 12 调用API

## （三）实验设计

### 1.性能实验

我们在Ubuntu15.10部署了QoS over LLDP的SDN环境，这里我们安装编译好已经修改过源代码的Floodlight控制器和Open vSwitch2.5.0，并设计三组拓扑进行统计LLDP数据包的开销，同时也在没有部署QoS over LLDP（Floodlight控制器和Open vSwitch2.5.0源代码没有进行修改）而仅使用标准LLDP的SDN环境中设计相同的拓扑统计LLDP数据包的开销。三组拓扑如下图。

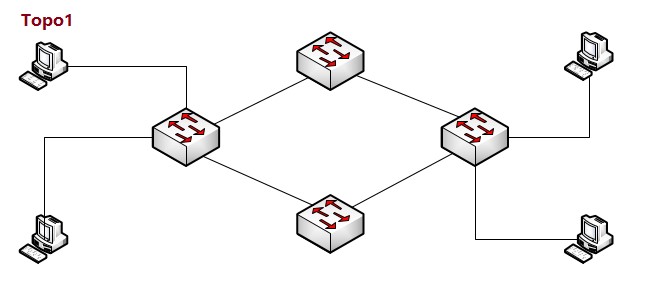


图 13 性能试验拓扑一

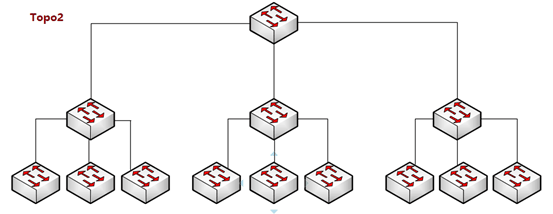


图 14 性能试验拓扑二

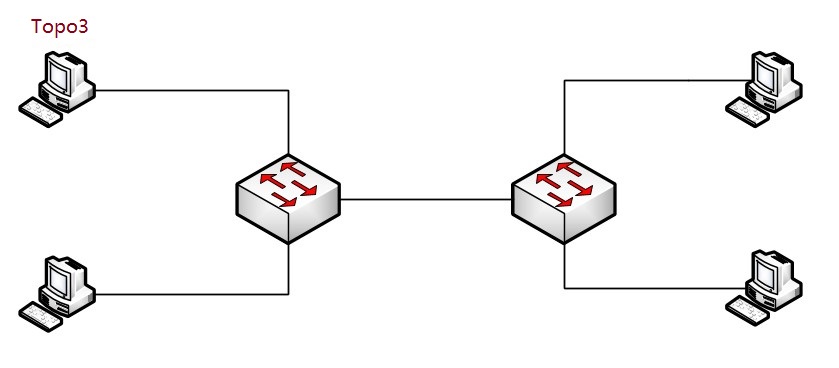


图 15 性能试验拓扑三

实验数据如图14，我们使用Wireshark进行抓包，分别统计了1分钟、10分钟和30分钟LLDP包的流量和经Openflow协议的总流量如图15、16，从实验数据我们很难看出修改过的LLDP数据包对整体经Openflow协议流量的影响。正如我们所说，只是增加了LLDP数据包中TLV字段的二十几个字节，网络开销比较小。

图 16 试验数据

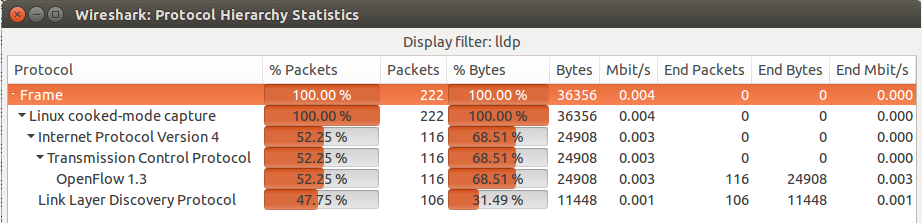


图 17 过滤出来的LLDP流量

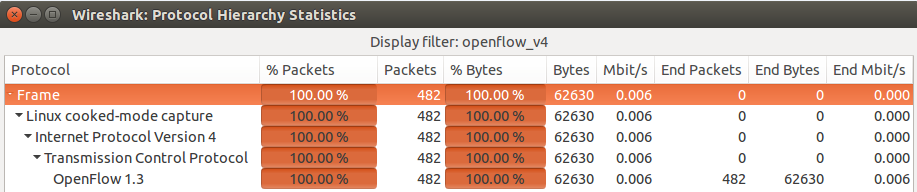


图 18 过滤出来的Openflow协议流量

### 2.应用实验

我们在部署了QoS over LLDP的环境里。设置一个拓扑，将所有的QoS信息设置成随机值，然后我们基于QoS信息找到一条最优路径。先获取所有链路的QoS信息，每条路径以带宽、时延、抖动、和丢包的瓶颈值作为参考标准，使用层次分析模型对每一指标进行评定，最后任意两点之间的每一条路径进行打分，最高者为最优路径。整体算法流程如下：

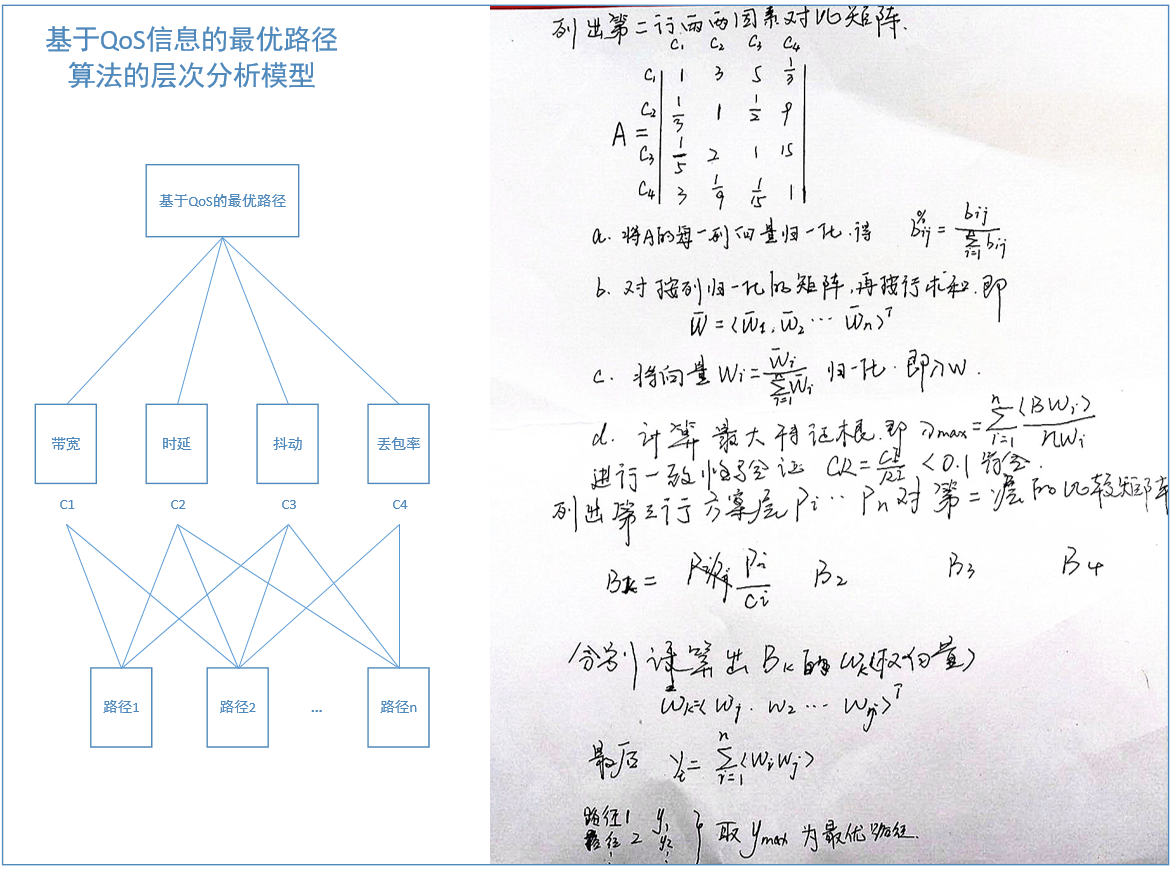


图 19 计算最优路径流程

拓扑图如下所示：

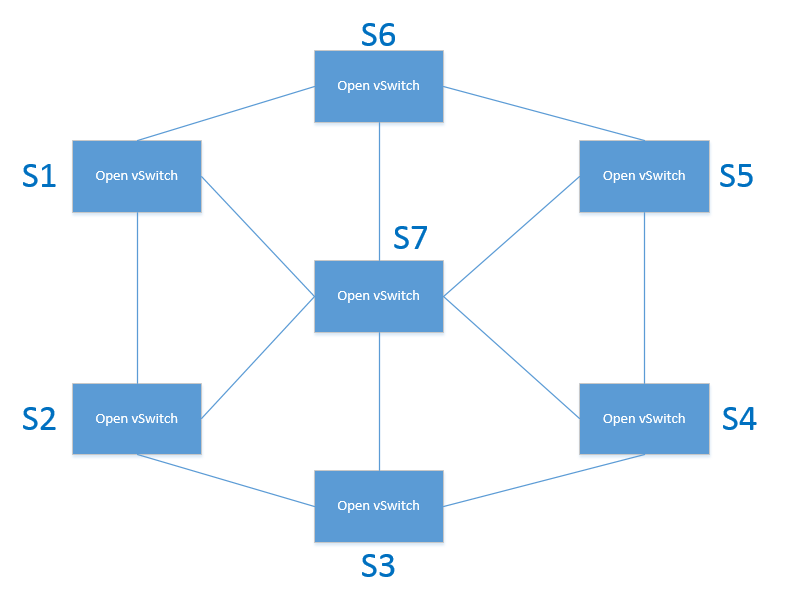
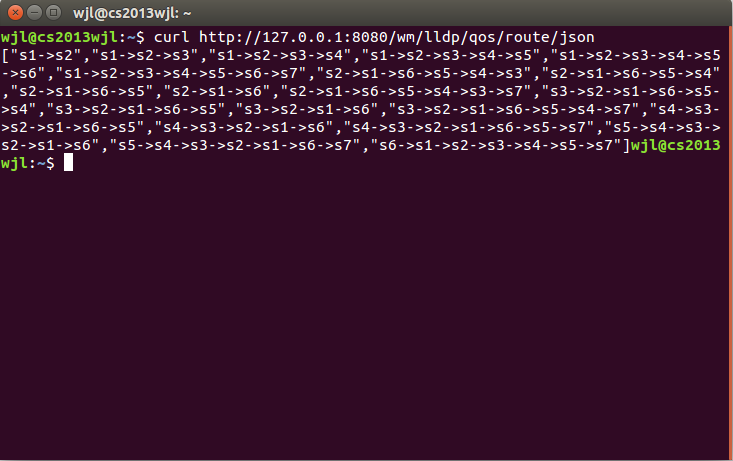


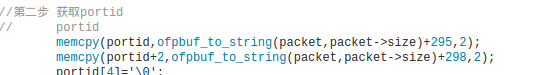
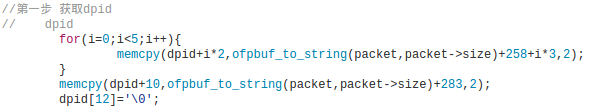
图 20 拓扑示意图

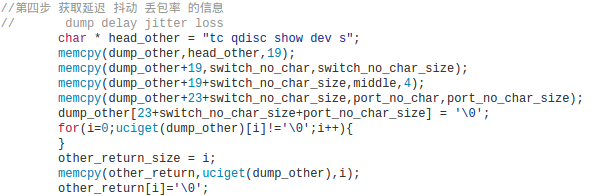
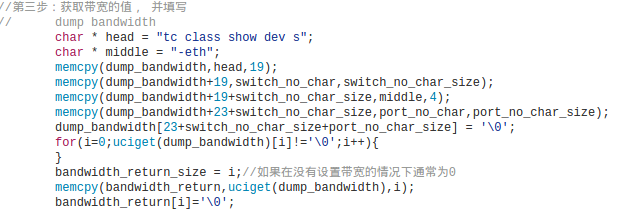
打印出任意两点之间的最优路径如下图：

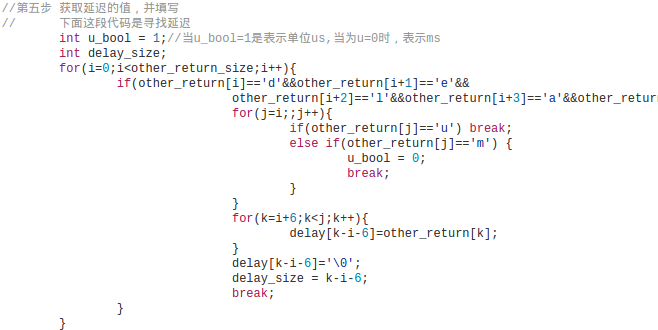


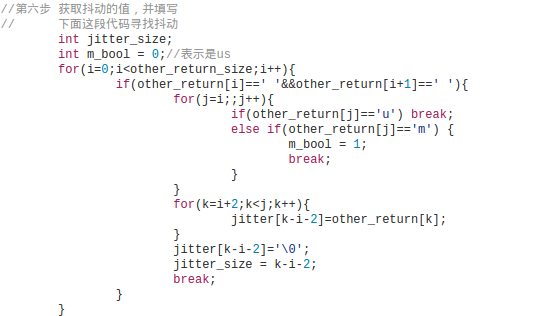
# 附录：

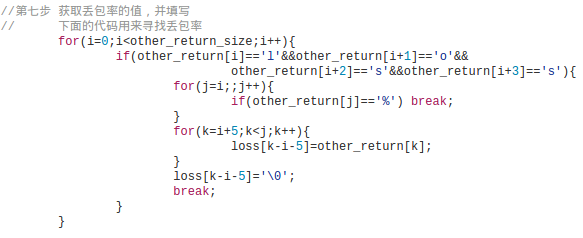
## Open vSwitch部分



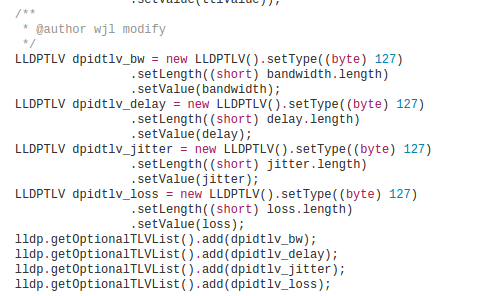
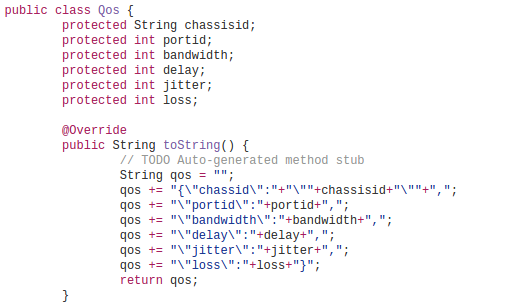








## Floodlight部分



# 参考文献

**[1]Thomas D.Nadeau&Ken Gray.SDN** 与 **OpenFlow** 解析**[M].**北京**:**  
人民邮电出版社**, 2014.5**  
**[2]Magnus Lie Hetland.Python** 基础教程（第 **2** 版**.**修订版） **[M].**北京**:**  
人民邮电出版社， **2014**  
**[3]**黄韬**,**刘江**,**魏亮**,**张娇**,**刘韵洁**.** 软件定义网络核心原理与应用实践  
**[M].** 北京**:**人民邮电出版社**,2014.9**  
**[4]Siamak Azodolmolky.**软件定义网络基于 **OpenFlow**的 **SDN**技术揭  
秘**[M].**北京**:**机械工业出版社**, 2014.6**  
**[5]**华为技术有限公司**. LLDP** 技术白皮书**[Z]. 01, 2012.10.31**  
**[6]**雷葆华**,** 王峰**,** 王茜**,** 王和宇**. SDN** 核心技术剖析和实战指南**[M].**  
北京**:** 电子工业出版社**, 2013.9**  
**[7] Weslsy J.Chun. Python** 核心编程**(**第二版**)[M].** 北京**:** 人民邮电出  
版社