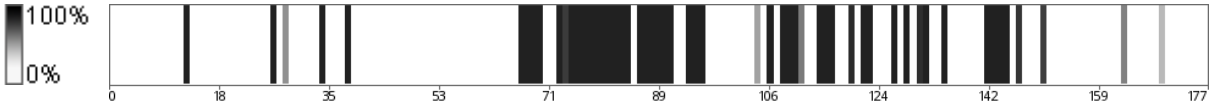
PaperPass旗舰版检测报告
简明打印版

比对结果(相似度):

总　体：43% (总体相似度是指本地库、互联网的综合对比结果)
本地库：20% (本地库相似度是指论文与学术期刊、学位论文、会议论文、图书数据库的对比结果)
期刊库： 7% (期刊库相似度是指论文与学术期刊库的比对结果)
学位库：18% (学位库相似度是指论文与学位论文库的比对结果)
会议库： 1% (会议库相似度是指论文与会议论文库的比对结果)
图书库： 2% (图书库相似度是指论文与图书库的比对结果)
互联网：42% (互联网相似度是指论文与互联网资源的比对结果)

编　号：5AD945732E87F2XY1
版　本：旗舰版
标　题：SDN实验设计
作　者：SDN实验设计
长　度：5303字符(不计空格)
句子数：177 句
时　间：2018-4-20 9:42:11
比对库：学术期刊、学位论文、会议论文、书籍数据、互联网资源
查真伪：http://www.paperpass.com/check

句子相似度分布图:



本地库相似资源列表(学术期刊、学位论文、会议论文、书籍数据):

1.相似度：18% 篇名：《数据中心网络架构及虚拟网络映射研究》
来源：学位论文 西安电子科技大学 2013
2.相似度： 2% 篇名：《数据中心网络的体系结构??》
来源：学术期刊 《软件学报》 2013年2期
3.相似度： 2% 篇名：《基于OpenFlow的网络测量研究》
来源：学术期刊 《网络安全技术与应用》 2017年8期
4.相似度： 1% 篇名：《面向云计算的数据中心网络的研究》
来源：学位论文 西安电子科技大学 2014
5.相似度： 1% 篇名：《数据中心网络拓扑结构研究》
来源：学术期刊 《电脑知识与技术》 2016年21期
6.相似度： 1% 篇名：《异构网络环境下物理拓扑自动发现算法研究》
来源：学位论文 复旦大学 2008
7.相似度： 1% 篇名：《网络拓扑结构研究与分析》
来源：学术期刊 《计算机光盘软件与应用》 2013年17期
8.相似度： 1% 篇名：《网络工程设计与实验教程》
来源：书籍数据 电子工业出版社 2010-02-01
9.相似度： 1% 篇名：《OpenFlow交换机流表转发设计与实现》
来源：学术期刊 《中国计量学院学报》 2015年3期
10.相似度： 1% 篇名：《网络工程设计与实验教程》

来源：书籍数据 电子工业出版社 2010-02-01
11.相似度： 1% 篇名：《OpenFlow协议分析与应用优化研究》
来源：学位论文 华中科技大学 2015
12.相似度： 1% 篇名：《基于Mininet的SDN仿真与性能分析》
来源：学术期刊 《信息通信》 2017年3期
13.相似度： 1% 篇名：《数据中心网络拓扑研究》
来源：学术期刊 《智能计算机与应用》 2014年5期
14.相似度： 1% 篇名：《基于组合导航系统应用的网络交换机设计研究》
来源：学位论文 哈尔滨工程大学 2008
15.相似度： 1% 篇名：《三层交换技术探究》
来源：学术期刊 《魅力中国》 2017年6期
16.相似度： 1% 篇名：《宽带接入网设备安装与维护》
来源：书籍数据 西安电子科技大学出版社 2010-10-01
17.相似度： 1% 篇名：《OpenFolw技术在SDN网络中的实现与优化》
来源：学位论文 山东大学 2015
18.相似度： 1% 篇名：《基于packet tracer的校园网络设计仿真》
来源：学术期刊 《中国电子商务》 2010年12期
19.相似度： 1% 篇名：《天安数码城园区网络的规划与设计》
来源：学位论文 华中师范大学 2016
20.相似度： 1% 篇名：《交换机MAC地址表的产生与管理》
来源：学术期刊 《计算机与网络》 2012年9期
21.相似度： 1% 篇名：《交换式以太网MAC地址学习功能的安全性研究》
来源：学术期刊 《科技创新导报》 2007年36期
22.相似度： 1% 篇名：《浅谈网络安全——IP地址与MAC地址绑定分析》
来源：会议论文 2006-03-01
23.相似度： 1% 篇名：《二层交换机的攻击分析与防范技术的研究》
来源：学术期刊 《网络安全技术与应用》 2005年7期
24.相似度： 1% 篇名：《计算机网络实验教程》
来源：书籍数据 清华大学出版社;北京交通大学出版社 2010-03-01
25.相似度： 1% 篇名：《云数据中心的虚拟机放置问题研究》
来源：学位论文 西安电子科技大学 2016
26.相似度： 1% 篇名：《OpenFlow交换机流缓存技术研究与实现》
来源：学位论文 国防科学技术大学 2014
27.相似度： 1% 篇名：《SDN网络中资源预留系统的设计》
来源：学位论文 电子科技大学 2015
28.相似度： 1% 篇名：《基于模型检测的OpenFlow多交换机数据包转发协议的分析与验证》
来源：学术期刊 《计算机科学》 2016年10期
29.相似度： 1% 篇名：《计算机网络实用教程》
来源：书籍数据 北京邮电大学出版社 2010-07-01
30.相似度： 1% 篇名：《DPClos:基于3级Clos结构的集装箱数据中心网络设计》
来源：学术期刊 《大连理工大学学报》 2016年6期
31.相似度： 1% 篇名：《基于VMware的虚拟网络管理系统设计》
来源：学位论文 中国计量大学 2016
32.相似度： 1% 篇名：《数据中心网络负载均衡与容错研究》
来源：学位论文 浙江大学 2014
33.相似度： 1% 篇名：《基于SDN的数据中心网络动态负载均衡研究》
来源：学位论文 湖南师范大学 2015
34.相似度： 1% 篇名：《下一代网络技术中流量和资源管理机制研究》
来源：学位论文 西北工业大学 2014
35.相似度： 1% 篇名：《ARP欺骗网络安全研究》
来源：学术期刊 《信息通信》 2017年1期
36.相似度： 1% 篇名：《数字化变电站的应用研究》
来源：学位论文 华北电力大学 2011
37.相似度： 1% 篇名：《基于软件定义网络的STP和RIP协议的研究与实现》
来源：学位论文 山东大学 2017

互联网相似资源列表：

1.相似度：24% 标题：《RYU控制器代码解析-简单交换机 - CSDN博...》
https://blog.csdn.net/manml/article/details/78059433
2.相似度：22% 标题：《FatTree胖树拓扑结构 - bryantin...》
https://www.cnblogs.com/zhuting/p/8880475.html
3.相似度：22% 标题：《数据中心拓扑总结.doc》
https://max.book118.com/html/2016/0823/52577657.shtm
4.相似度：22% 标题：《数据中心拓扑总结-第一范文网》
http://www.lanxicy.com/read/64616a84e9a5275759cdb6b8.html
5.相似度：22% 标题：《 数据中心拓扑总结_百度文库》
https://wenku.baidu.com/view/b8b6cd595727a5e9846a6164.html
6.相似度：17% 标题：《数据中心网络架构及虚拟网络映射分析.pdf》
https://max.book118.com/html/2017/0528/109850218.shtm
7.相似度： 8% 标题：《Mininet模拟多个数据中心的网络拓扑 S...》
https://www.sdnlab.com/mininet-topology/
8.相似度： 7% 标题：《 技术专栏 Mininet模拟不同数据中...》
https://www.douban.com/note/463588924/?type=like
9.相似度： 7% 标题：《mininet 模拟不同数据中心的网络拓扑》
http://www.docin.com/p-1153718749.html
10.相似度： 5% 标题：《Mininet模拟不同数据中心的网络拓扑 - C...》
https://blog.csdn.net/ka_ci_la714/article/details/50540495
11.相似度： 5% 标题：《mininet 简单的自定义拓扑_百度文库》
https://wenku.baidu.com/view/2ca3bff7866fb84ae55c8db3.html
12.相似度： 5% 标题：《Mininet模拟不同数据中心的网络拓扑-SDN...》
http://blog.itpub.net/29943020/viewspace-1368008/
13.相似度： 4% 标题：《Ubuntu16.04安装Mininet - C...》
https://blog.csdn.net/linyixiao88/article/details/65651390
14.相似度： 2% 标题：《SDN环境的配置(Mininet+Floodli...》
https://blog.csdn.net/ltt440888/article/details/78184457
15.相似度： 2% 标题：《 SDN学习之实现环路通信-布布扣-bubuko...》
http://www.bubuko.com/infodetail-2100196.html
16.相似度： 2% 标题：《SDN学习之实现环路通信 - 守功 - 博客园》
https://www.cnblogs.com/sgatbl/p/6861771.html
17.相似度： 2% 标题：《 SDN学习之实现环路通信》
http://www.mamicode.com/info-detail-1835705.html
18.相似度： 2% 标题：《SDN学习之RYU源码安装_其他分类_非百站新闻...》
http://www.genshuixue.com/i-cxy/p/15657510
19.相似度： 2% 标题：《原生支持OpenFlow1.3协议的Minine...》
http://www.lai18.com/content/2248977.html
20.相似度： 1% 标题：《Mininet使用源码安装以及简单使用 ---笔...》
https://weibo.com/p/230418b05e8c1f0102zam3

全文简明报告:

四、SDN虚拟环境搭建

4.1实验的网络技术

4.1.1 Mininet仿真平台

使用源码编译安装的方式安装mininet，安装步骤如下：

更新软件

apt-get update

apt-get upgrade

如果ubuntu没有安装git，需要安装

apt install git

从github上获取Mininet源码

git clone git: //github.com/mininet/mininet

{90%：获取源码后可以查看当前获取的Mininet版本，在~/mininet目录下，可以通过git tag 命令列出所有可用的Mininet版本}

cd mininet

git tag

或者你想安装的任意版本

git checkout -b 2.2.1 2.2.1

获取源码树并安装Mininet

#cat INSTALL

切换到mininet文件下：

cd mininet 注意： 如果前面进行了cd mininet此时则不需要再cd mininet

这里有多个安装选项： mininet/util/install.sh[options]

“-a”： 完整安装包括Mininet VM，还包括如Open vSwitch等依赖关系软件，以及像的OpenFlow Wireshark和POX。 {100%：默认情况下，这些工具将被安装在你的home目录中。} 完整安装命令：

{56%: # ./util/install.sh -a 我选择的是完整安装，装完这步可以跳到（4）了}

“-nfv”： 安装Mininet、基于OpenFlow的交换机和Open vSwitch。 命令：

./util/install.sh - nfv

“-s mydir” ： {100%：使用此选项可将源代码建立在一个指定的目录中，而不是在home目录中。}

#./util/install.sh -s mydir

另外，你只想安装OpenFlow1.3和Open vSwitch2.3.0，可以使用安装命令：

mininet/util/install.sh -n3V 2.3.0

{93%：安装完成后可通过简单的命令测试Mininet的基本功能}

sudo mn --test pingall

可以查看安装好的Mininet版本：

mn --version

4.1.2 Ryu控制器

安装RYU，具体步骤如下：

（1）安装前先更新一下系统：

sudo apt-get upgrade

（2）安装RYU，需要安装一些python的套件：

$ sudo apt-get install python-pip python-dev build-essential

$ sudo pip install --upgrade pip

$ sudo apt-get install python-eventlet

$ sudo apt-get install python-routes

$ sudo apt-get install python-webob

$ sudo apt-get install python-paramiko

（3）安装RYU，采用获取源码的方式，也可以采用pip安装方式（ pip install ryu 即可）

$ git clone git: //github.com/osrg/ryu.git

$ cd ryu

$ sudo pip install -r tools/pip-requires

$ sudo python setup.py install

测试

$ cd

$ ryu-manager

4.2实验的设计过程

4.2.1 拓扑结构设计

数据中心是对 SDN应用较快较广泛的地方，其中最迫切地需求是为服务器群组提供高效的双向带宽互联， 通常采用层次性多根网络拓扑，其中胖树（ fat- tree）结构因其简单易用而得到广泛应用，本实验基于胖树结构的 SDN网络进行仿真， {100%：胖树网络的好处是具有一个多层次的树状拓扑结构固有的容错能力。}

{90%：Fat-tree拓扑结构是由MIT的Al-Fares等人在改进传统树形结构性能的基础上提出的，属于switch-only型拓扑。} {92%：整个拓扑网络分为三个层次，自下而上分别为边缘层（ Edge）、汇聚层（ Aggregate）以及核心层（ Core），} {96%：其中汇聚层交换机与边缘层交换机构成一个 pod。} Fat-tree构建拓扑规则如下： FatTree拓扑中包含的 Pod数目为 k，每一个 pod连接的 sever数目为(〖 k/2)〗^2， {93%：每一个 pod内的边缘交换机及聚合交换机数量均为 k/2，} {82%：核心交换机数量为(〖 k/2)〗^2，网络中每一个交换机的} {93%：端口数量为 k，网络所能支持的服务器总数为 k^3/4}

{94%： FatTree结构采用水平扩展的方式，当拓扑中所包含的 pod数目增加，交换机的端口数目增加时， FatTree拓扑能够支持更多的服务器，} {96%：满足数据中心的扩展需求，如 k=48时， FatTree能够支持的服务器数目为27648。}

{97%：FatTree结构通过在核心层多条链路实现负载的及时处理，避免网络热点；} {100%：通过在pod内合理分流，避免过载问题。}

{98%：FatTree对分带宽随着网络规模的扩展而增大，因此能够为数据中心提供高吞吐传输服务；} {100%：不同pod之间的服务器间通信，源、目的节点之间具有多条并行路径，因此网络的容错性能良好，一般不会出现单点故障；} {100%：采用商用设备取代高性能交换设备，大幅度降低网络设备开销；} {100%：网络直径小，能够保证视频、在线会与等服务对网络实时性的要求；} {100%：拓扑结构规则、对称，利于网络布线及自动化配置、优化升级等。}

Fat-Tree结构也存在一定的缺陷： {100%：Fat-Tree结构的扩展规模在理论上受限于核心交换机的端口数目，不利于数据中心的长期发展要求；} {100%：对于Pod内部，Fat-Tree容错性能差，对底层交换设备故障非常敏感，当底层交换设备故障时，难以保证服务质量；} {100%：拓扑结构的特点决定了网络不能很好的支持one-to-all及all-to-all网络通信模式，不利于部署MapReduce、Dryad等现代高性能应用；} {100%：网络中交换机与服务器的比值较大，在一定程度上使得网络设备成本依然很高，不利于企业的经济发展。}

{91%：一个简单的基于两个数据中心的网络拓扑如下图：}

{100%：图中c1和c2为核心交换机，a1-a4为聚合交换机，e1-e4为边缘交换机，h1-h8为主机。}

Mininet提供了一个内置的可视化工具 miniedit，因此可以直接使用可视化工具通过拖拽图标与修改属性的方式搭建出理想的拓扑， 也可以通过编写 python文件来自定义拓扑结构。

{100%：使用Mininet设计模拟不同网络数据中心拓扑，可以用来分析网络的总流量，而且除此之外，还可以通过负载均衡策略来保证数据中心的可用性。} {100%：不同数据中心网络拓扑管理设计，主要是基于胖树（ Fat- Tree）拓扑创建网络，} {100%：胖树网络的好处是具有一个多层次的树状拓扑结构固有的容错能力。}

构造自定义拓扑主要使用以下三个函数：

增加主机

addHost(’ hostname’)

增加交换机

addSwitch(’ switchname’)

增加链路

addLink(node1，node2，node1_port，node2_port)

4.2.3 开发Ryu应用

{51%：Ryu是控制器，通过下发流表给交换机来进行数据包转发决策。} 首先，我们应该知道：

{100%：控制器和交换机握手完成之后，一条默认流表项应该被添加进流表。} 它的作用是为了让交换机发送Packet-In消息。 {93%：这条默认流表项被称为Table-Miss，不过Table-Miss不是默认存在的，他也是由控制器下发的。} {100%：我们在控制器代码里面定义的 MAC地址表是只有控制器知道的，对于交换机来说，只有流表才能帮助它转发，} {100%：所以控制器要给交换机下发流表，如果你不在交换机添加详细的流表项，那么你应该让交换机收到每一个数据包都上报控制器，} {63%：由控制器下发 Packet- Out报文进行处理，否则数据包会被丢弃，控制器下发的流表项包含了应该对数据包进行的动作。}

一、编写Ryu组件addtag.py，功能是实现打标签。

首先是继承Ryu的app_manager.RyuApp基类，然后定义以下函数：

{93%：switch_features_handler函数的作用就是下发一条Table miss流表，用于将不知如何转发的数据包上报控制器。} {100%：鉴于流表项的特殊性，我们在交换机向控制器发送完自身信息的时候让控制器下发这条流表项，这个时候修饰器会调用这个函数。} {100%：这条流表项需要具有以下特点：} 优先级最低，保证最后匹配； 必须要能匹配所有数据包。 {87%：所以我们指定了一个空的match，并且把这条流表项的优先级设置为0，将OFPCML_NO_BUFFER指定为max_len。} 调用的add_flow函数是我们自己定义的。

{94%：add_flow函数的参数表较多，因为这些在下发流表的时候调用的FlowMod类里面都要用到。} {96%：所以简要的介绍一下用到的参数（括号内是默认值）:} datapath：是传来数据的交换机，内有ID编号; priority(0)： {96%：指定流表项的优先级，影响匹配顺序，优先级越大越先匹配;} match(None)： {96%：流表项要匹配的内容，比如进入接口，目的地址等;} buffer_id（ofproto_v1_3.OFP_NO_BUFFER）： {86%：指定在OpenFlow交换机上缓冲的数据包的缓冲区ID。} {97%：当未指定缓冲区ID时，请设置OFP_NO_BUFFER;} instructions ([]): 交换机要执行的指令列表。 {93%：指定好所有的值最后就调用send_msg来发送流表项。}

delete_flow函数即负责删除流的函数

push_ tag_ action函数实际定义了一个操作数据包的动作，该动作负责添加 vlan或者修改 vlan， 本实验以链路号作为标签，把标签保存在 vlan中，因此 vlan的

值即为链路号的有序组合，在该函数中，需要编写代码来实现打标签的逻辑，比如需要判断链路号是否需要作为标签存入 vlan中（当该链路是唯一可达路径上的一部分时，该链路号不需要作为标签保存），需要判断前面是否已经存入了标签来决定保存该标签的方式等。

{98%：_ packet_ in_ handler函数要完成的功能比较多，首先它需要获取进入的接口，} {100%：源和目的 MAC地址，然后填充 MAC地址表，注意一个细节，我们需要根据交换机编号和源地址来指定接口，} {100%：因为同一个接口可能会对应不同的 MAC地址。} {100%：然后做一下判断，如果MAC地址表里面没有目的MAC地址，就设置为广播；}有地址就设置为出接口。 {84%：然后是添加流表项操作，我们只能在找到了目的地的情况下才添加流表项，} 对于本次实验来说，定义流表项的动作集时，填充的最关键的动作就是打标签的动作，调用 push_ tag_ action函数定义该动作。这里有个概念buffer_id需要理解，buffer_id: {81%： 指定openflow交换机上封包对应的缓冲区，若不需要，则指定为OFP_NO_BUFFER。} openflow中buffer_id分别在三类消息中定义，并且起到的作用均是不同的，第一类、Packetin消息： 用于标记缓存在交换机中的数据报文id，如报文被 action上送到控制器中 maxlen字段或者 table_ miss消息限制长度，而通过 bufferid将报文缓存在交换机中，以便被另外两种消息来调用； 第二类、Packetout消息： 用于控制器将原先buffer在交换机中的报文，通过Packetout个形式从交换机的某个物理口送出去； 第三类、Flowmod消息： 如果 flowmod中带有bufferid，那么说明这个 flowmod需要做两件事情，第一是正常下发一条 flow，其次是把交换机中先前 buffer的那个数据报文， Packetout到 table来匹配一次下的这条flow； 以上两个指令都是通过这个带有bufferid的消息执行的，不需要控制器另外下packet_out消息，这种设计思路是非常巧妙的。

二、编写Ryu组件getpath.py，功能是提取数据包的标签，回溯路径并记录当前时间。

该组件的目的不是定义数据包转发规则，而是读取数据包内容，因此_ packet_ in_ handler函数中主要需要实现的是，按照保存标签的逻辑提取出标签并组成队列，对队列中相邻的元素进行树的广度优先遍历来找出两个元素之间的唯一路径，{61%：最后将这些路径连接起来组成一条完整的路径。}

4.2.4 Mininet与Ryu连接

本实验定义了胖树网络拓扑， topo. py，在该文件中，使用 net. addController函数（该函数有两个参数，第一个参数为控制器的名称，第二个参数为控制器的监听端口）添加了控制该拓扑的两个控制器 con0（ 监听的端口号为6633）和con1（监听的端口号为6634）， {44%：让 con0与拓扑中的所有交换机相连， con1则只与边缘交换机相连。}

打开终端1，运行： sudo ryu-manager addtag.py --ofp-tcp-listen-port=6633

打开终端2，运行： sudo ryu-manager getpath.py --ofp-tcp-listen-port=6653

打开终端3，运行： sudo python build_ topo. py进入 mininet终端，在mininet的终端可进行数据包的 ping操作，如果在实现组件的时候有将结果展示到终端，那么可以在终端1，2看到相应的结果。

检测报告由PaperPass文献相似度检测系统生成