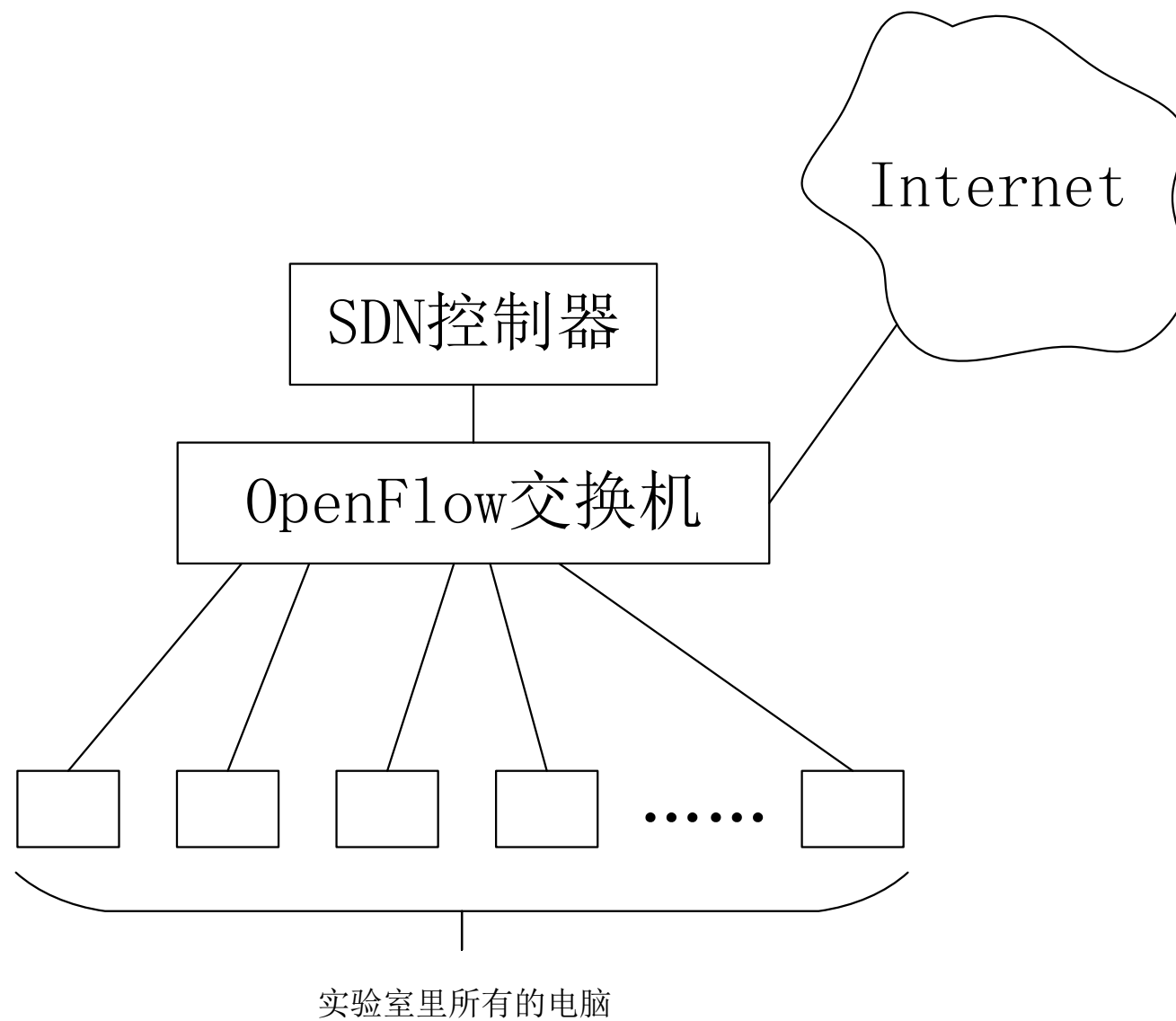


实验室搭建SDN

步骤

- 1.在一台电脑上安装开源SDN控制器。
- 2.连接SDN控制器和OpenFlow交换机。
- 3.将所有的电脑连接在OpenFlow交换机上。
- 4.将外网连接在OpenFlow交换机上。

示意图



不购需要买控制器

由于控制器选择开源的，所以不需要购买，只需要把一台电脑做成控制器。

需要购买一个OpenFlow交换机

形号Pica8 3297。

支持OpenFlow协议，有48个以太网接口，实验室基本够用。

价格35000元、36000元左右。

购买清单

1. Pica8 3297交换机，价格35000元、36000元左右。

总价格35000元、36000元左右

9月13日

自制OpenFlow交换机

方案1: 10个TP-LINK WR842N, 每个86, 一共860

方案2: 10个2手的TP-LINK各种形号, 每个20多, 一共200多。

openWRT是一种开源的路由器上的操作系统, 可以在上面安装OpenFlow 1.3。先在电脑上配置好openWRT和OpenFlow 1.3, 然后再将其安装到openWRT所支持的路由器上, 这样就制作好OpenFlow交换机了。网上有很多这个教程。

自制OpenFlow交换机

大致步骤：

- 1.在系统里下载openWRT源码并编译，选择与自己的路由器相应的选项
- 2.下载OpenFlow-openWRT的源码并编译，选择相应的选项
- 3.编译完成后会生成.bin文件，将其刷入路由器，TP-LINK WR740N的方法是登陆路由器管理界面192.168.x.x升级固件。

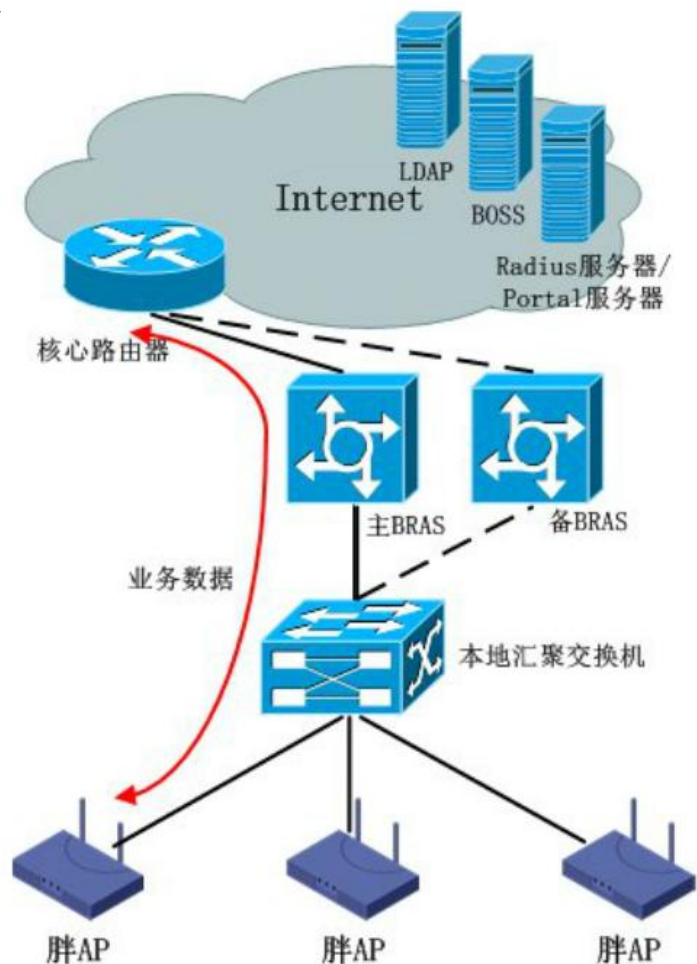
openWRT支持的路由器型号在这里查询：

<http://wiki.openwrt.org/toh/start>

基于SDN架构的WLAN组网方案

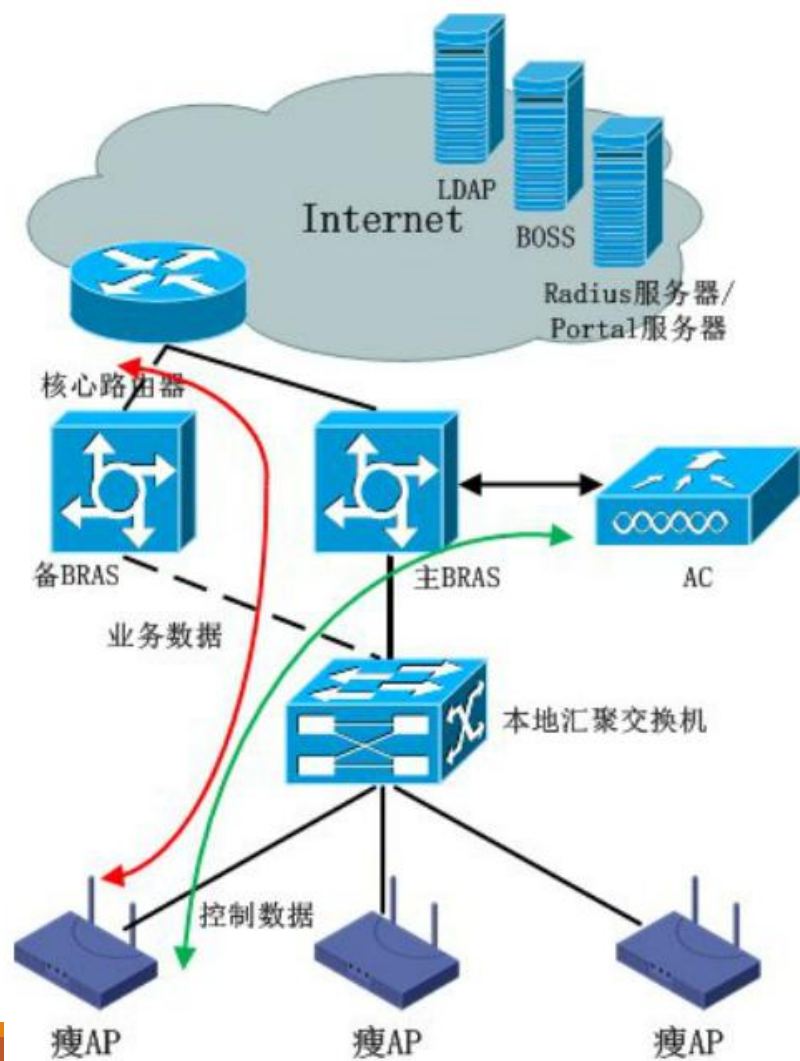
2016-9-13 彭建云

传统WLAN组网方案——胖AP



- 组网模式比较简单，可用于小规模组网
- AP之间独立工作
- AP本地存储大量配置信息，设备丢失会造成配置信息泄漏
- 难以进行无线网络状态数据的采集
- 没有统一管理，网络维护麻烦，软件升级工作量大
- 对终端漫游支持不足

传统WLAN组网方案——瘦AP+AC



- AC集中管理AP
- AP管理控制数据经过AC处理
- 用户接入控制数据由BRAS处理
- 业务数据由本地交换机完成转发
- AC所管理瘦AP的物理区域较大
- AC设备需要量少

WLAN网络演进路线

集中式



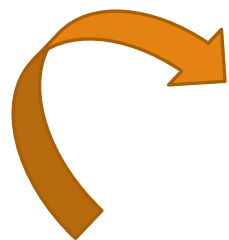
分布式



虚拟化

- | | | |
|---|--|--|
| <ul style="list-style-type: none">• 组网模式比较简单• 便于组建小型网络• 不便于大规模部署• 网络可管可控性差• 对漫游支持不足 | <ul style="list-style-type: none">• 便于大规模部署• 网络可管可控性好• 对漫游支持好• 网络资源利用率不高• 网络瓶颈明显• 不同品牌的AC和AP不兼容 | <ul style="list-style-type: none">• 硬件标准化• 组网成本更低• 网络资源利用率更高• 网络规划、设计和管
理更加灵活• 云计算广泛应用 |
|---|--|--|

组网设备的演进



胖AP的功能:

数据加密/频射管理/用户
认证/网络管理/QoS保障/
安全策略/支持漫游等

交换机的功能:

数据转发/路由/流量均衡
/QoS管理/安全机制等

瘦AP的功能:

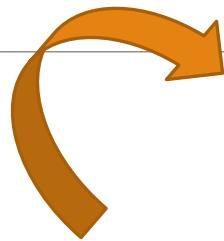
数据加密/频射管理/安全防御

交换机功能:

数据转发/路由/流量均衡/QoS
管理/安全机制等

AC的功能:

用户认证/网络管理/QoS保障/
安全策略/支持漫游/负载均衡



瘦AP的功能:

数据加密/频射管理/安全
防御

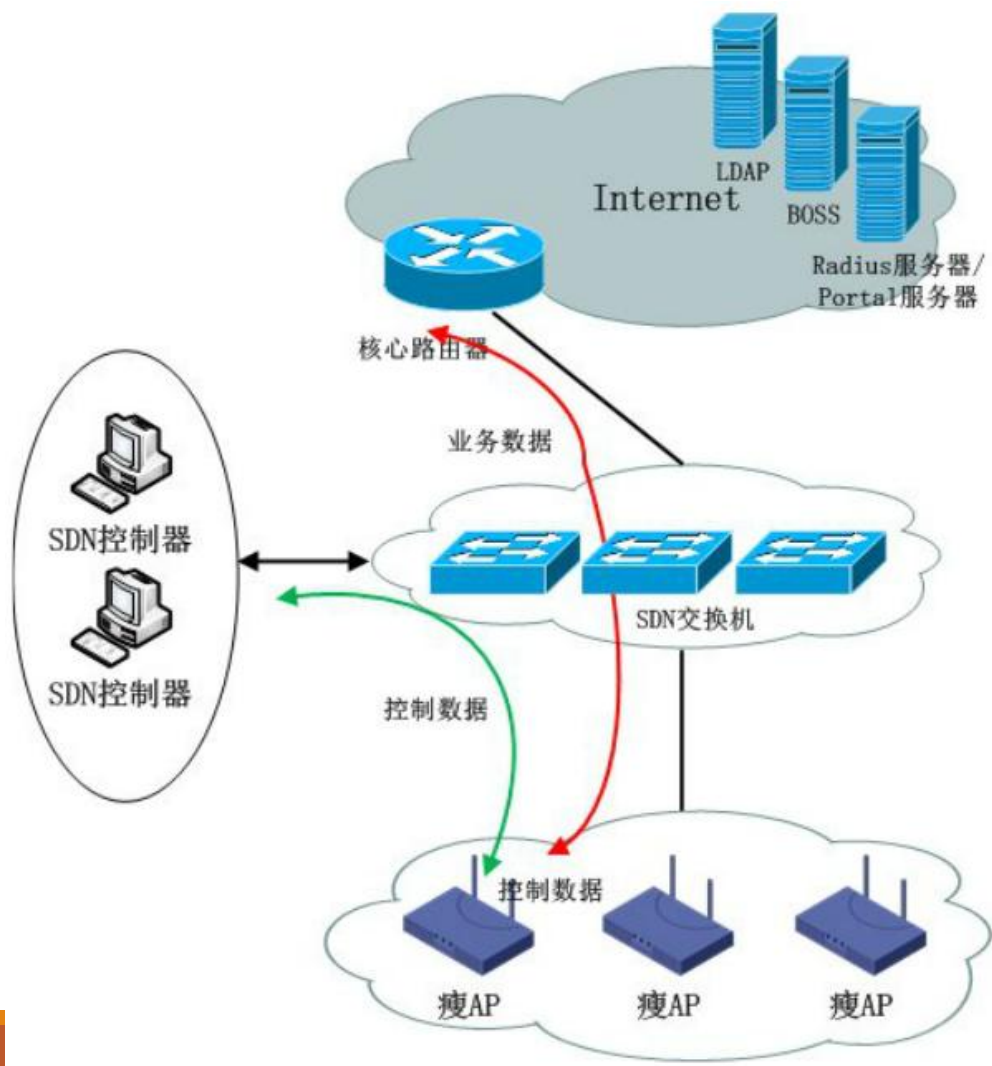
SDN交换机的功能:

数据转发/安全机制等

SDN控制器的功能:

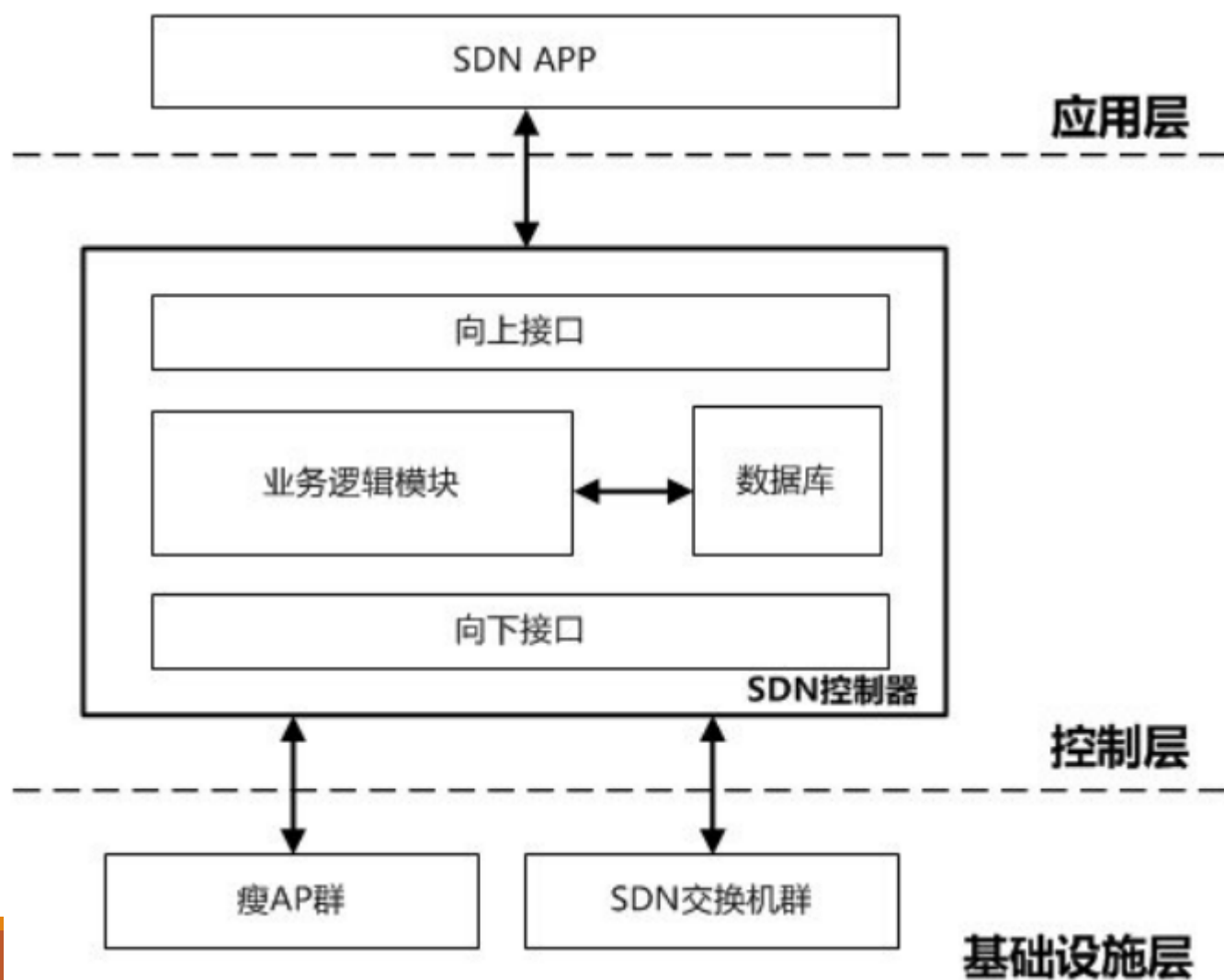
用户认证/网络管理/QoS保
障/安全策略/支持漫游/负
载均衡/流量均衡/路由等

SDN架构的WLAN组网方案

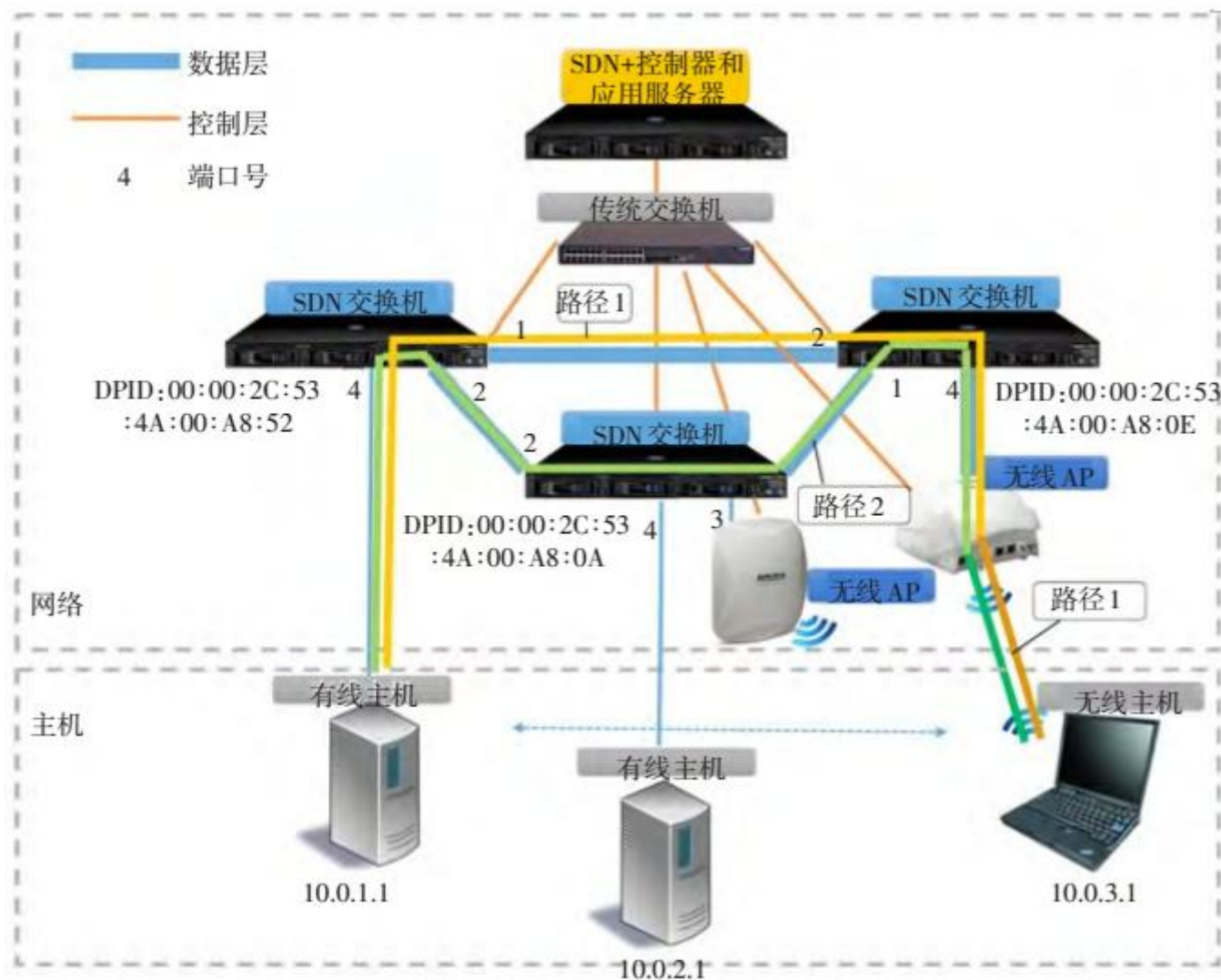


- 无线管理和接入控制都在SDN控制器上通过软件编程实现
- 用户业务数据转发通过SDN交换机直接进行本地转发，提高交换机的转发能力
- SDN控制根据网络实时动态，对数据进行流量均衡

基于SDN的WLAN系统架构



- 在AP和交换机中植入OpenFlow协议支持模块，使得SDN控制器可以通过相关协议对AP和交换机进行控制和管理
- SDN控制器通过 向下接口实现对数据转发层的AP和交换机的控制，解析AP和交换机的行为并进行管理
- 上层应用开发人员可通过SDN控制器的向上接口添加新的功能应用或策略模块
- 业务逻辑模块包含对各种数据的处理和管理
- 数据库包含从底层采集的各项数据信息



设备	数量	硬件参数	软件
SDN+控制器	1	CPU: Intel® Xeon(R) CPU E5405 @ 2.00 GHz × 4 MEM: 4 GB NIC: Intel Corporation	Floodlight 和对无线设备的扩展协议
SDN 交换机	3	82580 Gigabit Network	OvS2.0.1
无线 AP	2	Aruba AP Rukus AP	802.11a/b/g/n/ac 802.11a/b/g/n
主机	3	Servers Laptop	Iperf server Iperf client
应用服务器	1	与 SDN+控制器一致	Floodlight Module & Python script

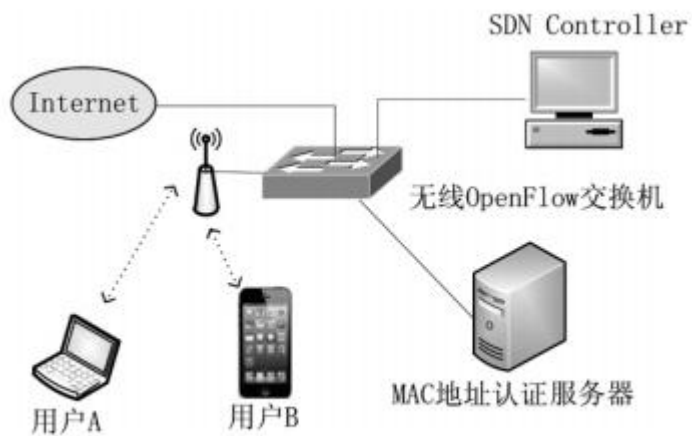


图 3 无线 SDN 网络

无线OpenFlow交换机：
TP-Link 841n无线路由器+OpenFlow模块
(OpenWrt系统)

SDN

与

传统网络

传统网络

传统网络从一开始就是一个分布式的网络，没有中心的控制节点，网路中的各个设备之间通过口口相传的方式学习网络的可达信息，由每台设备自己决定要如何转发，这直接导致了没有整体观念，不能从整个网络的角度对流量进行调控。由于是口口相传，就必须使用大家都会的语言，这就是网络协议。各个设备供应商不能自己随便开发协议，否则不同厂商各执己见，网络还是不通。这样全球性的组织就诞生了，比如IETF，而RFC就是网络协议的法律，相当于国际法，各个设备供应商遵从国际法行事，就基本保证了整个网络世界的正常运行。

传统网络的协议为了适应各种不同的需求场景，发展也越来越复杂。

从传统网络的部署方式能看出来，传统网络新业务部署是补丁式的，因为网络中新老设备并存，新业务必须兼容老业务，新的协议基本上都是在原有协议上进行扩展，这就大大限制的设备提供商的想象力。

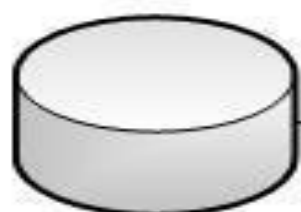
简单回顾一下什么是SDN

软件定义网络（Software Defined Network, SDN），是由美国斯坦福大学clean slate研究组提出的一种新型网络创新架构，SDN技术就相当于把每人家里路由器的管理设置系统和路由器剥离开。以前我们每台路由器都有自己的管理系统，而有了SDN之后，一个管理系统可用在所有品牌的路由器上。SDN的设计理念是将网络的控制面与数据转发面进行分离，并实现可编程化控制。

SDN的典型架构共分三层，最上层为应用层，包括各种不同的业务和应用；中间的控制层主要负责处理数据平面资源的编排，维护网络拓扑、状态信息等；最底层的基础设施层负责基于流表的数据处理、转发和状态收集。

应用层

基于SDN应用软件



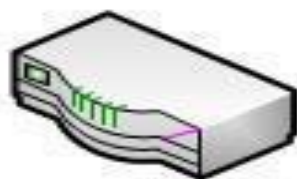
APP

应用层：通过控制层提供的开放API控制设备的转发功能

API接口

控制层

SDN 控制系统



SDN 控制器

控制层：由SDN控制器组成，与下层可用OpenFlow协议通信

OpenFlow协议

基础设施层



OpenFlow交换机



OpenFlow交换机



OpenFlow交换机

基础设施层：由转发设备组成

SDN的好处

SDN本质上具有“控制和转发分离”、“有开放的编程接口”和“集中式的控制”三大特性，这带来了一系列的好处。

第一，设备硬件归一化，硬件只关注转发和存储能力，与业务特性解耦，可以采用相对廉价的商用的架构来实现。

第二，网络的智能性全部由软件实现，网络设备的种类及功能由软件配置而定，对网络的操作控制和运行由服务器作为网络操作系统（NOS）来完成。

第三，对业务响应相对更快，可以定制各种网络参数，如路由、安全、策略、QoS、流量工程等，并实时配置到网络中，开通具体业务的时间将缩短。

总结

如果说传统网络是一个不断打补丁的木桶，SDN就是把木桶打破，重新建一个水池，甚至水库。

SDN将转发和控制分离，集中控制起来，就意味着原来各自为政的交换机和路由器，从原来的散兵游勇变成了一支军队，SDN能整合所有的资源，能做的事情会发生质变。

所以，即使现在的SDN实现基本上是试验性的，但已减少网络管理经费50%，减少IP地址使用60%。谷歌把它连接全世界数据中心的G-Scale WAN换成OpenFlow网络，使预留空间的利用率提高95%，不损失任何东西。

而我们有理由相信，这只是SDN的开始，它为我们打开了新世界的大门。

番外：有趣的SDN定义

盛科张卫峰

第一类是狭义SDN(等同于Openflow),

第二类是广义SDN(控制与转发分离),

第三类是超广义SDN(管理与控制分离)。

而且我认为，第二类定义中的SDN，是最通用，最有价值的一种。

第四类，SDN的本质定义就是软件定义网络，也就是说希望应用软件可以参与对网络的控制管理，满足上层业务需求，通过自动化业务部署简化网络运维，这是SDN的核心诉求。换句话说，控制与转发分离只是为了满足SDN的核心诉求的一种手段，如果某些场景中有别的手段可以满足，那也可以，比如管理与控制分离。

集线器、中继器、交换机、路由器相互之间的区别？可否互换？

答：

集线器===N口中继器：带宽共享

交换机：带宽独享，交换机内部核心处有一个交换矩阵，为任意两端口间的通信提供通路，或是一个快速交换总线，以使由任意端口接收的数据帧从其他端口送出。

路由器：带宽共享

结论：交换机能注重转发平面,而路由器更注重控制平面.

三层交换机与路由器的区别：

1、交换机网络打通速度很慢，即线速转发，比如交换机的带宽是1Gbps，带在没有达到线速转发之前，速度是很低的，并且达到线速转发的时间可能要很久，以IPv6为例，有些设备需要十几分钟才能达到线速转发，而路由器是不需要消耗这么久的，路由器的路由表计算是走CPU的，任何时间都是线速转发的（当然如果CPU负载太重则例外）——这是交换机最致命的弱点。

2、交换机抗网络震荡能力很弱，前面说了线速转发的问题，同样，网络一旦开始震荡，路由表不稳定的时候，交换机就又不能线速转发了，而路由器则基本不受影响

48口百兆交换机：

华为(Huawei)非管理交换机S1700-52R-2T2P-AC 48口百兆交换机

华为以行践言！产品咨询电话：13366266860



天天比价·品牌热卖 爆品低价

比价

¥1146.00 降价通知

比苏宁低82.00元，比价时间：2016-09-12 17:37:17(比价仅供参考)▼

48口百兆路由器：

华为 (HUAWEI) AR2201-48FE-S 48口百兆全能路由器

在线支付，顺丰包邮，5分好评后，联系客服金额500以下送20话费，500以上送30话费

¥5800.00 降价通知

实验室交换机换成路由器会怎么样？

路由器是共享带宽，而且，需要解包到第三层，所以在转换大数据量时，路由器承受大。

宿舍里的路由器换成交换机会怎么样？

不能互换

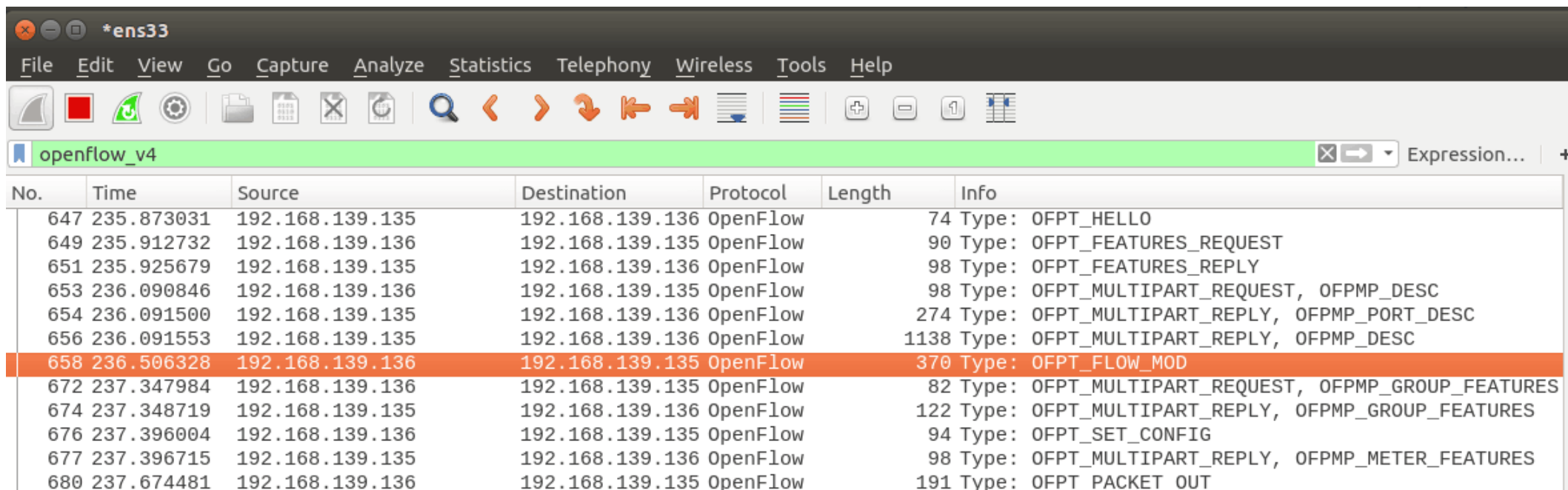
如果你的外线只有一个帐号或者一个地址，那么你内部电脑就只能通过扩展地址才能上网，就以你只有一个地址为例（就算是一个帐号，拨号之后会获取到一个地址），那么需要将内部的多个地址做NAT（Network Address Translation，网络地址转换）之后，转换成运营商提供给你的那个地址才能出去上网，这个功能只有路由器、防火墙、上网行为管理有，任何交换机都不具备这个功能。



无线路由器作为无线交换机？

无线路由器可以作为无线交换机使用

抓包讲解OpenFlow协议



*ens33

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

openflow_v4

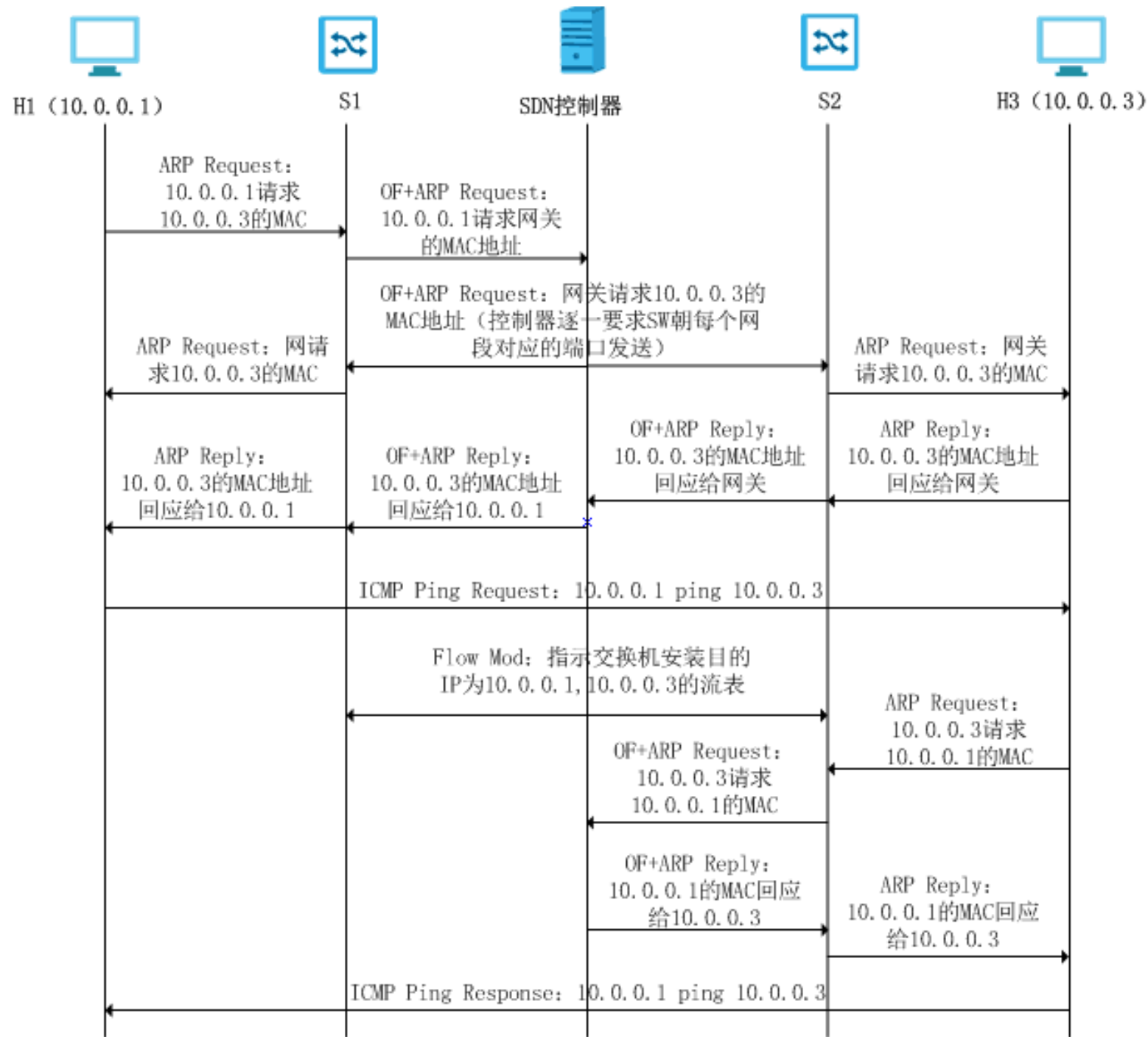
No.	Time	Source	Destination	Protocol	Length	Info
647	235.873031	192.168.139.135	192.168.139.136	OpenFlow	74	Type: OFPT_HELLO
649	235.912732	192.168.139.136	192.168.139.135	OpenFlow	90	Type: OFPT_FEATURES_REQUEST
651	235.925679	192.168.139.135	192.168.139.136	OpenFlow	98	Type: OFPT_FEATURES_REPLY
653	236.090846	192.168.139.136	192.168.139.135	OpenFlow	98	Type: OFPT_MULTIPART_REQUEST, OFPMP_DESC
654	236.091500	192.168.139.135	192.168.139.136	OpenFlow	274	Type: OFPT_MULTIPART_REPLY, OFPMP_PORT_DESC
656	236.091553	192.168.139.135	192.168.139.136	OpenFlow	1138	Type: OFPT_MULTIPART_REPLY, OFPMP_DESC
658	236.506328	192.168.139.136	192.168.139.135	OpenFlow	370	Type: OFPT_FLOW_MOD
672	237.347984	192.168.139.136	192.168.139.135	OpenFlow	82	Type: OFPT_MULTIPART_REQUEST, OFPMP_GROUP_FEATURES
674	237.348719	192.168.139.135	192.168.139.136	OpenFlow	122	Type: OFPT_MULTIPART_REPLY, OFPMP_GROUP_FEATURES
676	237.396004	192.168.139.136	192.168.139.135	OpenFlow	94	Type: OFPT_SET_CONFIG
677	237.396715	192.168.139.135	192.168.139.136	OpenFlow	98	Type: OFPT_MULTIPART_REPLY, OFPMP_METER_FEATURES
680	237.674481	192.168.139.136	192.168.139.135	OpenFlow	191	Type: OFPT_PACKET_OUT

抓包讲解OpenFlow协议

No.	Time	Source	Destination	Protocol	Length	Info
658	236.506328	192.168.139.136	192.168.139.135	OpenFlow	370	Type: OFPT_FLOW_MOD
672	237.347984	192.168.139.136	192.168.139.135	OpenFlow	82	Type: OFPT_MULTIPART_REQUEST, OFPMP_GROUP_FEATURES
674	237.348719	192.168.139.135	192.168.139.136	OpenFlow	122	Type: OFPT_MULTIPART_REPLY, OFPMP_GROUP_FEATURES
676	237.396004	192.168.139.136	192.168.139.135	OpenFlow	94	Type: OFPT_SET_CONFIG
677	237.396715	192.168.139.135	192.168.139.136	OpenFlow	98	Type: OFPT_MULTIPART_REPLY, OFPMP_METER_FEATURES
680	237.674481	192.168.139.136	192.168.139.135	OpenFlow	191	Type: OFPT_PACKET_OUT
681	237.678106	192.168.139.136	192.168.139.135	OpenFlow	191	Type: OFPT_PACKET_OUT
690	240.130598	192.168.139.136	192.168.139.135	OpenFlow	274	Type: OFPT_FLOW_MOD
692	240.191899	192.168.139.136	192.168.139.135	OpenFlow	316	Type: OFPT_PACKET_OUT
694	240.870439	192.168.139.135	192.168.139.136	OpenFlow	178	Type: OFPT_PACKET_IN

抓包讲解OpenFlow协议

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	96:d8:56:61:bb:a1	CayeeCom_00:00:01	LLDP	85	TTL = 4919 System Name = openflow:1
2	5.000302945	96:d8:56:61:bb:a1	CayeeCom_00:00:01	LLDP	85	TTL = 4919 System Name = openflow:1
→ 3	9.358641578	10.0.0.2	10.0.0.1	ICMP	98	Echo (ping) request id=0x18a2, seq=1/256, ttl=64 (r
← 4	9.358663354	10.0.0.1	10.0.0.2	ICMP	98	Echo (ping) reply id=0x18a2, seq=1/256, ttl=64 (r
5	9.999660054	96:d8:56:61:bb:a1	CayeeCom_00:00:01	LLDP	85	TTL = 4919 System Name = openflow:1
6	10.357845398	10.0.0.2	10.0.0.1	ICMP	98	Echo (ping) request id=0x18a2, seq=2/512, ttl=64 (r
7	10.357866624	10.0.0.1	10.0.0.2	ICMP	98	Echo (ping) reply id=0x18a2, seq=2/512, ttl=64 (r
8	11.357893136	10.0.0.2	10.0.0.1	ICMP	98	Echo (ping) request id=0x18a2, seq=3/768, ttl=64 (r
9	11.357912322	10.0.0.1	10.0.0.2	ICMP	98	Echo (ping) reply id=0x18a2, seq=3/768, ttl=64 (r
10	12.357959967	10.0.0.2	10.0.0.1	ICMP	98	Echo (ping) request id=0x18a2, seq=4/1024, ttl=64 (
11	12.357981476	10.0.0.1	10.0.0.2	ICMP	98	Echo (ping) reply id=0x18a2, seq=4/1024, ttl=64 (
12	13.358057572	10.0.0.2	10.0.0.1	ICMP	98	Echo (ping) request id=0x18a2, seq=5/1280, ttl=64 (
13	13.358079008	10.0.0.1	10.0.0.2	ICMP	98	Echo (ping) reply id=0x18a2, seq=5/1280, ttl=64 (
14	14.358084691	10.0.0.2	10.0.0.1	ICMP	98	Echo (ping) request id=0x18a2, seq=6/1536, ttl=64 (
15	14.358106216	10.0.0.1	10.0.0.2	ICMP	98	Echo (ping) reply id=0x18a2, seq=6/1536, ttl=64 (



9月22日

OpenFlow的OpenWRT刷机部署

环境

- 1.Ubuntu系统，编译
- 2.Windows系统，刷机
- 3.支持OpenWRT的路由器

可以在这个网址查看支持OpenWRT的路由器型号：

<http://wiki.openwrt.org/toh/start#supported.hardware-.router.type>

我选用的是华为的HG255D。

安装依赖

```
$ sudo apt-get update
```

```
$ sudo apt-get install build-essential binutils flex bison autoconf gettext texinfo sharutils  
subversion libncurses5-dev ncurses-term zlib1g-dev gawk git-core unzip
```

下载源码编译

```
$ git clone http://git.openwrt.org/12.09/openwrt.git
```

（这一步有12.09， 14.07， 15.05三个版本， 12.09选不到HG255D。）

```
$ cd openwrt
```

```
$ mv feeds.conf.default feeds.conf
```

```
$ ./scripts/feeds update -a
```

```
$ ./scripts/feeds install -a
```

```
$ ./scripts/feeds install -a luci
```

```
$ ./scripts/feeds install -a luci
```

make menuconfig

对于hg255d，要做如下修改：

打开target/linux/ramips/image/目录下的Makefile文件。搜索HG255D，把下列2行前面的#去掉。

否则在make menuconfig的时候选不到hg255d。

make menuconfig

1、选择CPU型号

Target System——Ralink RT288x/RT3xxx。

2、选择路由型号

Target Profile——huawei hg255d。

3、添加luci

LuCI—>Collections—— <*> luci

4、添加luci的中文语言包

LuCI—>Translations—— <*> luci-i18n-chinese

make menuconfig

5、添加DDNS

LuCI—>Applications —> <*>luci-app-ddns

6、添加UTF8编码

Kernel modules —> Native Language Support —> <*> kmod-nls-utf8

添加OpenFlow

```
$ git clone https://github.com/CPqD/openflow-openwrt.git
```

建立软链接

```
$ cd .....openwrt/package/
```

```
$ ln -s .....openwrt/openflow-openwrt/openflow-1.3/
```

```
$ cd .....openwrt/
```

```
$ ln -s .....openwrt/openflow-openwrt/openflow-1.3/files
```

make menuconfig

选择Network -> openflow..... Open Flow 1.3 Switch Userspace Package

make

编译完成后，会在 `/openwrt/bin/` 目录下产生多个 `binary` 文件
找到 `*sysupgrade.bin` 文件进行刷机。

刷机

在Windows下进行。

1. 准备固件1102-0x20000_hg255d-squashfs-tftp.checksum2.bin，以及刚才生成的*sysupgrade.bin。
2. 用网线连接电脑和HG255D的lan口，设置固定IP:192.168.1.2，子网掩码：255.255.255.0
3. 打开IE并在地址栏输入http://192.168.1.1/upload.html
4. 用牙签顶住HG255D的reset键后打开电源，开机后一直保持按住reset键。
5. 回到IE按回车，再以每秒一次的速度刷新网页直到出现升级页面，就会出现刷机画面，这时候可以松开reset键。
6. 出现刷机画面后，马上上传1102-0x20000_hg255d-squashfs-tftp.checksum2.bin，快速按Update software。

刷机

7. 一开始进入刷机模式，路由器有五六个灯是亮的，等它自动刷好，会自动重启，这是电源灯一闪一闪，最后只有两三个灯亮，就可以确定刷机完成了。
8. 输入192.168.1.1进入管理页，账号root，密码admin，在“备份|刷机”界面中选择刚才生成的*sysupgrade.bin固件再次刷机。
9. 大概等四五分钟，这期间两三个灯常亮，刷完后会自动重启，电源灯会一闪一闪，之后常亮重启完，再次进入输入192.168.1.1进入管理页，账号root，密码admin，可以设置了。

SDN与传统网络 的未来

SDN现在面临的一些问题

网络设备:网络设备的芯片支持的流表数量很有限，报文匹配和动作都不够灵活，无法支持多级的流表等等

安全问题:控制器

协议不成熟: Openflow协议还处于发展初期。不过，正如我们的研发成果呈现的，现有的Openflow协议已经足够支撑很多网络应用的开发了。

但这些并不是关键的问题

关于SDN的不完美和问题可以列出一大堆，这也是各种网络技术研讨会议、论坛激烈讨论的议题，理论派、实战派、应用派等各方面的人士纷纷抛出各种见解，让这个市场异常热闹。如果说SDN搭建的是一个全新的网络大舞台，那么各种各色的人物都开始在这个舞台上尽力地表演着，舞台上节目越来越多，也越来越精彩，都想成为台下观看节目的数据中心用户心中最受欢迎的人物和作品。正是因为SDN有很多的不足，才会有这么多的议论，最终必将逐渐形成业界统一的标准，而现在正在这个变化的过程中。早在三十年前，网络协议也不是只有以太网协议，IBM，Apple都曾提出过自己的网络协议，但是在发展的过程中，逐渐由3com提出的Ethernet得到了用户的青睐，以至于今天发展成为唯一的局域网技术，SDN也正在经历着这个蜕变的过程。

SDN的应用场景

数据中心网络

政企网络

运营商网络

个人、家庭？

SDN 对用户的技术能力和创新意识要求比较高有一定关系。传统的用户可能愿意等待厂商把一切都做好了，没有意识到未来的网络其实就应该是用户自己来定义和实现（这个转变如同从软件的使用者转变到软件的编程者）。

现在很多厂商已经开始技术转型，把设备和网络开放给用户来控制；但用户是否做好了技术转型的心理和技术准备 有较高技术开发能力和创新意识的公司（如Google 和微软这类创新意识很强的互联网公司以及具有开发能力的各类网络用户，特别是行业专网和大型企业网的用户），将是 **SDN** 新技术的最先的使用者。

然后随着用户获得收益出现了更多普通用户的需求，会逐渐形成一个新的产业链，即出现提供 **SDN** 专项服务（例如配套的软件、工具和接口，应用设计，技术服务等）的新型企业，帮助那些尚不具备自主创新能力的普通用户来应用**SDN**。当然，出现这个新的产业链需要较长的时期

我们要正确地认识SDN，善于发现SDN的缺陷之美。同时要冷静头脑，不要过高估计SDN的能力。短期看，SDN技术不会取代传统网络，甚至看不到它有占据垄断地位的可能，但是它肯定会是现有网络的一个强力补充。而在未来，像数据中心等大型网络会被SDN取代，但是对于小型网络（特别是家庭网络），用户是否选择SDN还尚未可知。

综上所述，我的观点是，在未来，SDN和传统网络是并存。

9月30日

编译带有OpenFlow的OpenWRT并刷入华为HG255D

1. 安装依赖

```
ubuntu@ubuntu: ~/mine
ubuntu@ubuntu:~/mine$ sudo apt-get update
ubuntu@ubuntu:~/mine$ sudo apt-get install build-essential binutils flex bison a
utoconf gettext texinfo sharutils subversion libncurses5-dev ncurses-term zlib1g
-dev gawk git-core unzip
```

编译带有OpenFlow的OpenWRT并刷入华为HG255D

2. 下载OpenWRT版本为15.05的源码，并安装一些模块

```
ubuntu@ubuntu:~/mine$ git clone http://git.openwrt.org/15.05/openwrt.git
ubuntu@ubuntu:~/mine$ cd openwrt
ubuntu@ubuntu:~/mine/openwrt$ mv feeds.conf.default feeds.conf
ubuntu@ubuntu:~/mine/openwrt$ ./scripts/feeds update -a
ubuntu@ubuntu:~/mine/openwrt$ ./scripts/feeds install -a
ubuntu@ubuntu:~/mine/openwrt$ ./scripts/feeds install -a luci
```

编译带有OpenFlow的OpenWRT并刷入华为HG255D

3. 找到target/linux/ramips/image/目录下的Makefile文件。搜索HG255D，把下列2行前面的#去掉

```
ubuntu@ubuntu:~/mine/openwrt$ vim target/linux/ramips/image/Makefile
```

```
#Image/Build/Profile/HG255D=$(call BuildFirmware/Default16M/$(1),$(1),hg255d,HG255D)
```

```
Image/Build/Profile/HG255D=$(call BuildFirmware/Default16M/$(1),$(1),hg255d,HG255D)
```

编译带有OpenFlow的OpenWRT并刷入华为HG255D

4. 补充安装以下依赖包

```
ubuntu@ubuntu:~/mine/openwrt$ sudo apt-get install libp11-kit-dev
```

```
ubuntu@ubuntu:~/mine/openwrt$ sudo apt-get install libssl-dev
```

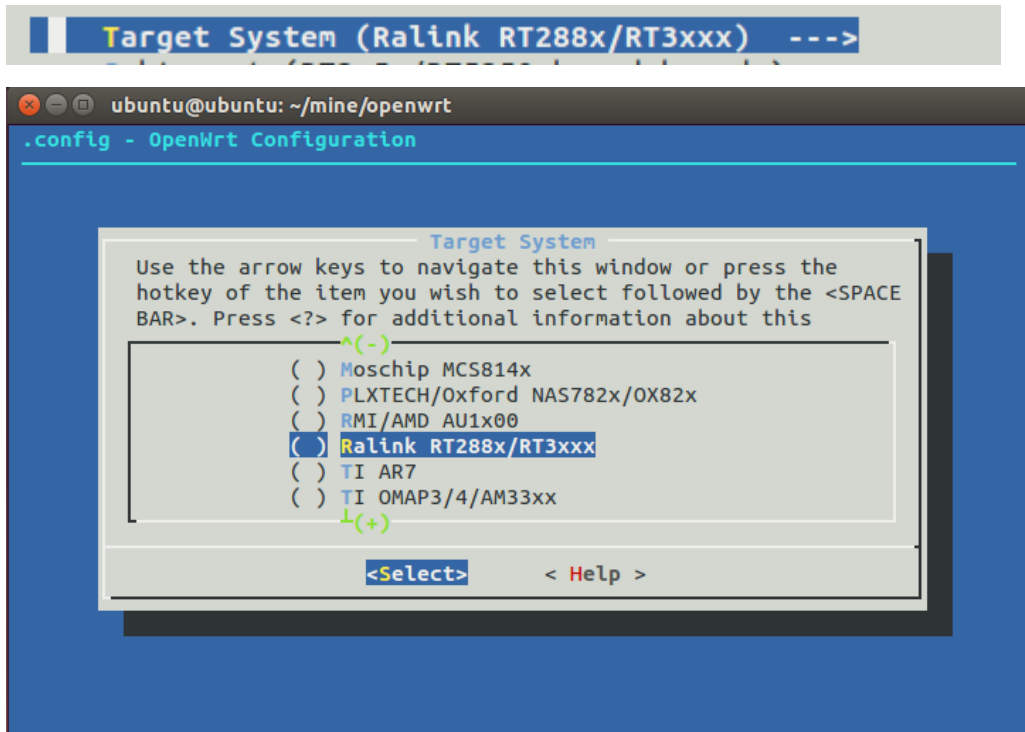

编译带有OpenFlow的OpenWRT并刷入华为HG255D

5. make menuconfig

```
ubuntu@ubuntu:~/mine/openwrt$ make menuconfig
```

make menuconfig

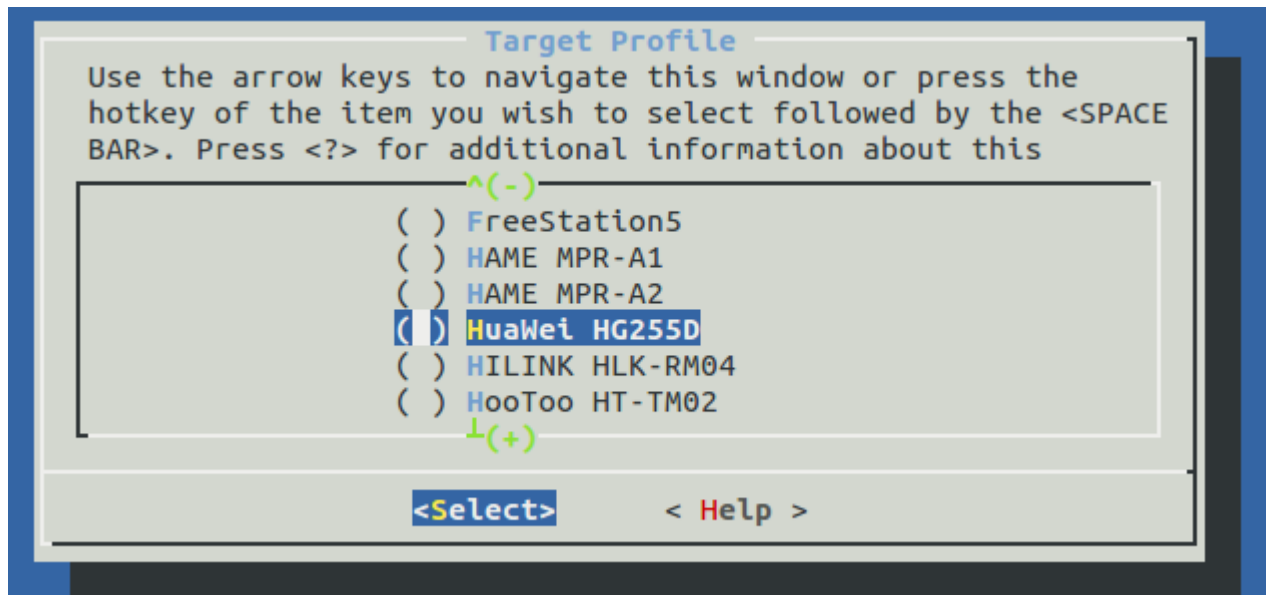
(1) Target System :



make menuconfig

(2) Target Profile:

Target Profile (HuaWei HG255D) --->



make menuconfig

(3) LuCI—>Collections— <*> luci,

LuCI—>Translations— <*> luci-i18n-Chinese

LuCI—>Applications —> <*>luci-app-ddns

Kernel modules —> Native Language Support —> <*> kmod-nls-utf8

编译带有OpenFlow的OpenWRT并刷入华为HG255D

6. make

```
ubuntu@ubuntu:~/mine/openwrt$ make
```

编译带有OpenFlow的OpenWRT并刷入华为HG255D

7. 下载OpenFlow的源码

```
ubuntu@ubuntu:~/mine/openwrt$ git clone https://github.com/CPqD/openflow-openwrt  
.git
```

编译带有OpenFlow的OpenWRT并刷入华为HG255D

8. 建立软链接

```
ubuntu@ubuntu:~/mine/openwrt$ cd package/
```

```
ubuntu@ubuntu:~/mine/openwrt/package$ ln -s ~/mine/openwrt/openflow-openwrt/openflow-1.3/
```

```
ubuntu@ubuntu:~/mine/openwrt/package$ cd ..
```

```
ubuntu@ubuntu:~/mine/openwrt$ ln -s ~/mine/openwrt/openflow-openwrt/openflow-1.3/files/
```

编译带有OpenFlow的OpenWRT并刷入华为HG255D

9. make menuconfig 选择Network→penflow

```
Mail --->  
Multimedia --->  
Network --->  
Sound --->
```

```
<*> odhcpd..... OpenWrt DHCP/DHCPv6(-PD)/RA Se  
(0) CER-ID Extension ID (0 = disabled)  
<*> openflow..... Open Flow 1.3 Switch Users  
< > openldap-server..... LDAP implementat  
< > openvswitch..... Open vSwitch Users  
... ..
```


编译带有OpenFlow的OpenWRT并刷入华为HG255D

10. make kernel_menuconfig, 选择Networking Support->Networking options->**Hierarchical Token Bucket (HTB)**

编译带有OpenFlow的OpenWRT并刷入华为HG255D

11. 用编译后产生的bin目录下openwrt-ramips-rt305x-hg255d-initramfs-ulmage.bin文件刷机



openwrt-ramips-rt305x-hg255d-initramfs-ulmage.bin

2016/9/1

刷机

在Windows下进行。

- (1) 用网线连接电脑和HG255D的lan口，设置固定IP:192.168.1.2，子网掩码：255.255.255.0
- (2) 打开IE并在地址栏输入http://192.168.1.1/upload.html
- (3) 用牙签顶住HG255D的reset键后打开电源，开机后一直保持按住reset键。
- (4) 回到IE按回车，再以每秒一次的速度刷新网页直到出现升级页面，就会出现刷机画面，这时候可以松开reset键。
- (5) 出现刷机画面后，马上上传openwrt-ramips-rt305x-hg255d-initramfs-ulmage.bin，快速按Update software。

刷机

(6) 一开始进入刷机模式，路由器有五六个灯是亮的，等它自动刷好，会自动重启，这是电源灯一闪一闪，最后只有两三个灯亮，就可以确定刷机完成了。

←

→

↻

192.168.1.1/upload.html

Update Software

Step 1: Obtain an updated software image file from your ISP.

Step 2: Enter the path to the image file location in the box below or click the "Browse" button to locate the image file.

Step 3: Click the "Update Software" button once to upload the new image file.

NOTE: The update process takes about 2 minutes to complete, and your DSL Router will reboot.

Software File Name:

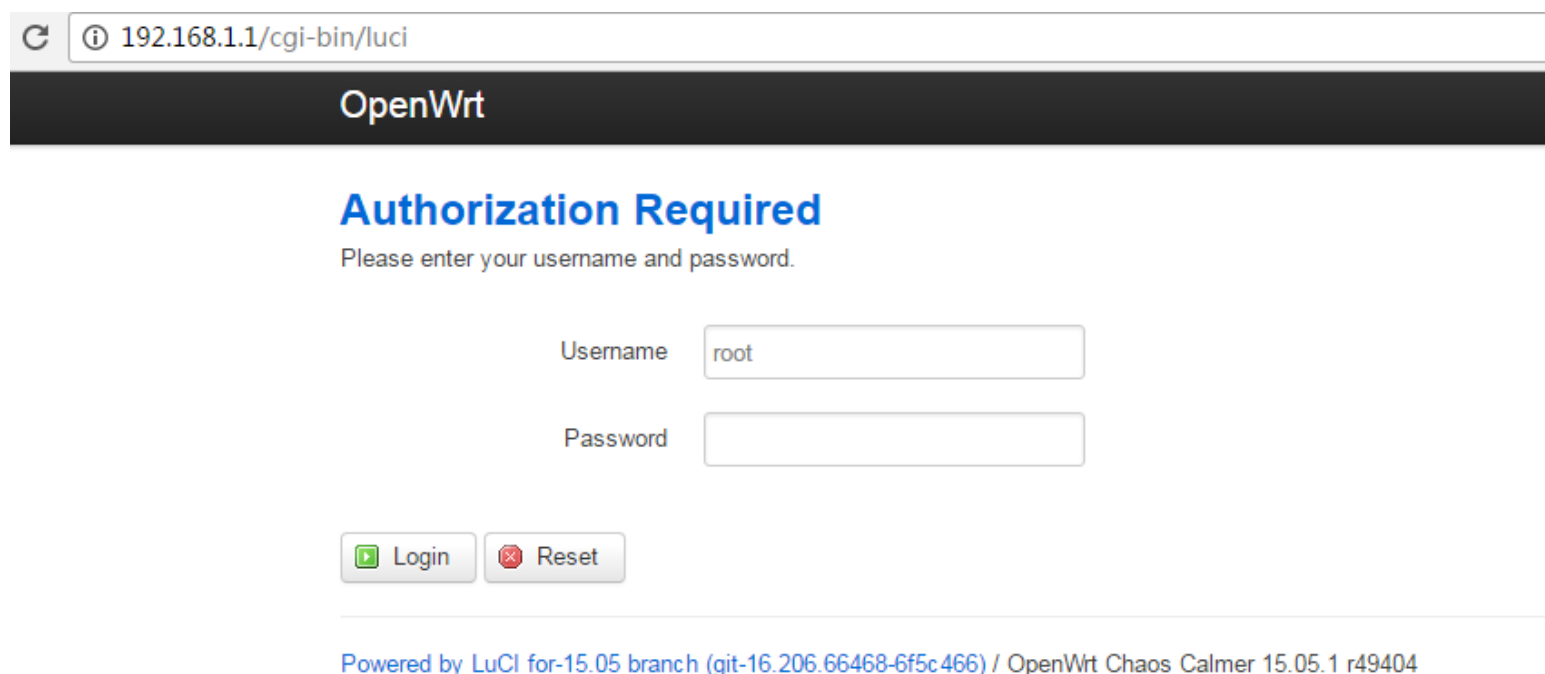
选择文件

未选择任何文件

Update Software

编译带有OpenFlow的OpenWRT并刷入华为HG255D

12. 刷机后的192.168.1.1登陆界面



192.168.1.1/cgi-bin/luci

OpenWrt

Authorization Required

Please enter your username and password.

Username

Password

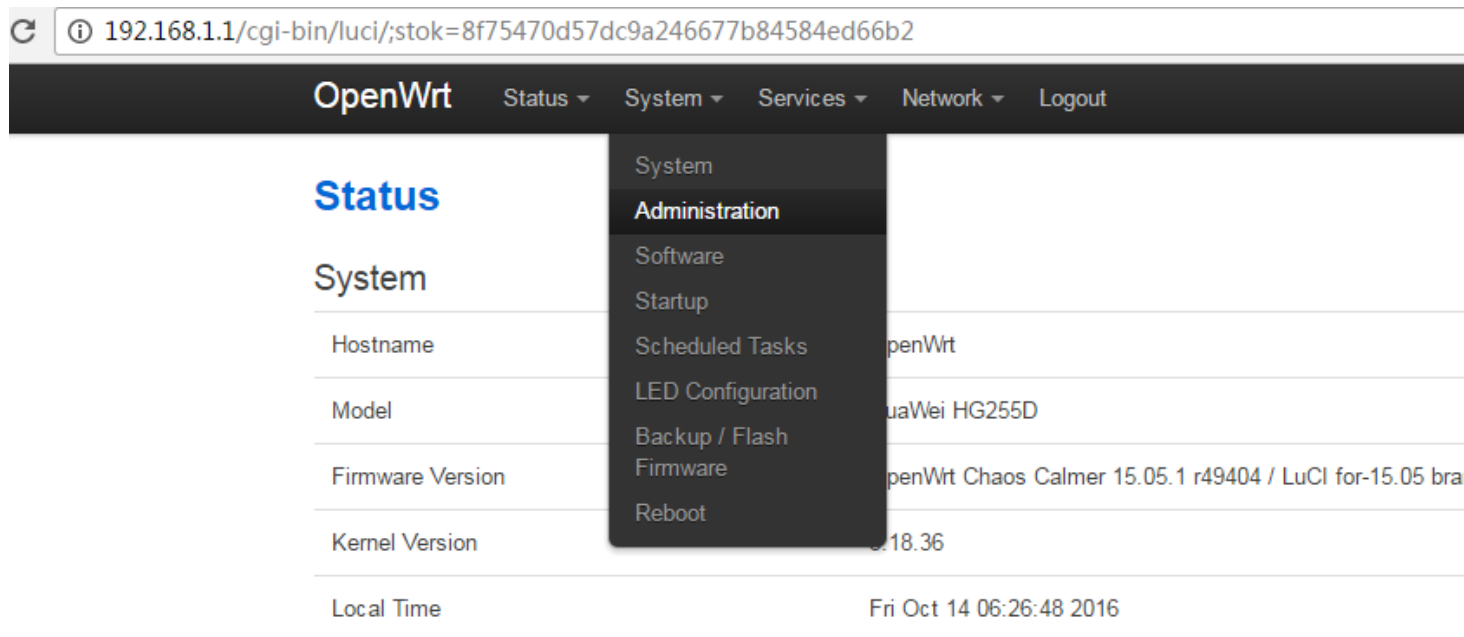
Powered by [LuCI for-15.05 branch \(git-16.206.66468-6f5c466\)](#) / OpenWrt Chaos Calmer 15.05.1 r49404

10月15日

设置ssh密码

在192.168.1.1登陆，用户名root，密码admin。

选择system->administration。




设置ssh密码

在password和confirmation里设置ssh密码，之后就可以用ssh打开openwrt里的linux终端了。


OpenWrt Status ▾ System ▾ Services ▾ Network ▾ Logout

Router Password
Changes the administrator password for accessing the device

Password

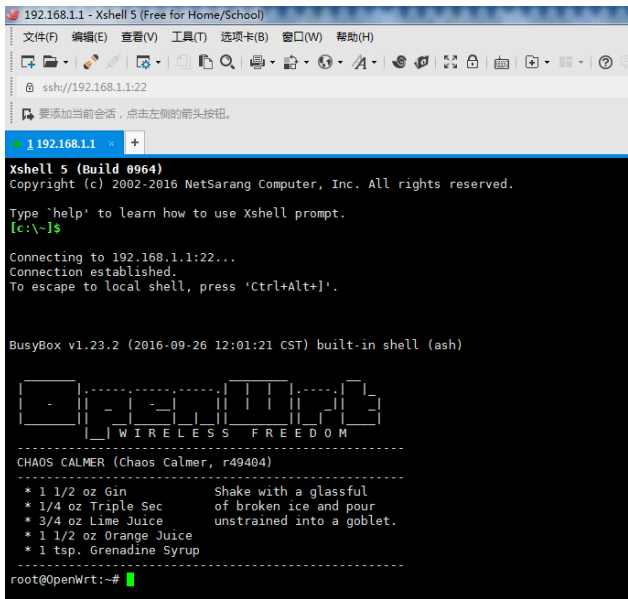


Confirmation



SSH Access

用ssh打开openwrt的终端



```
root@OpenWrt:~# ls
root@OpenWrt:~# cd ..
root@OpenWrt:/# ls
bin      dev      etc      init     lib      mnt      overlay  proc     rom      root     sbin     sys      tmp      usr      var      www
```

复制/lib里的functions.sh到/etc

```
root@OpenWrt:/# cp /lib/functions.sh /etc  
root@OpenWrt:/# █
```

修改/etc/config/openflow

在ofports处写上需要使用openflow的端口

在ofctl处写上控制器的相关信息，ip地址、端口。

```
root@OpenWrt:/# vim /etc/config/openflow
root@OpenWrt:/#
```

```
config 'ofswitch'
    option 'dp' 'dp0'
    option 'dpid' '00000000000001'
    option 'ofports' 'eth0.1 eth0.2'
    option 'ofctl' 'tcp:192.168.1.136:6633'
    option 'mode' 'outofband'
```

```
~
```

开启openflow

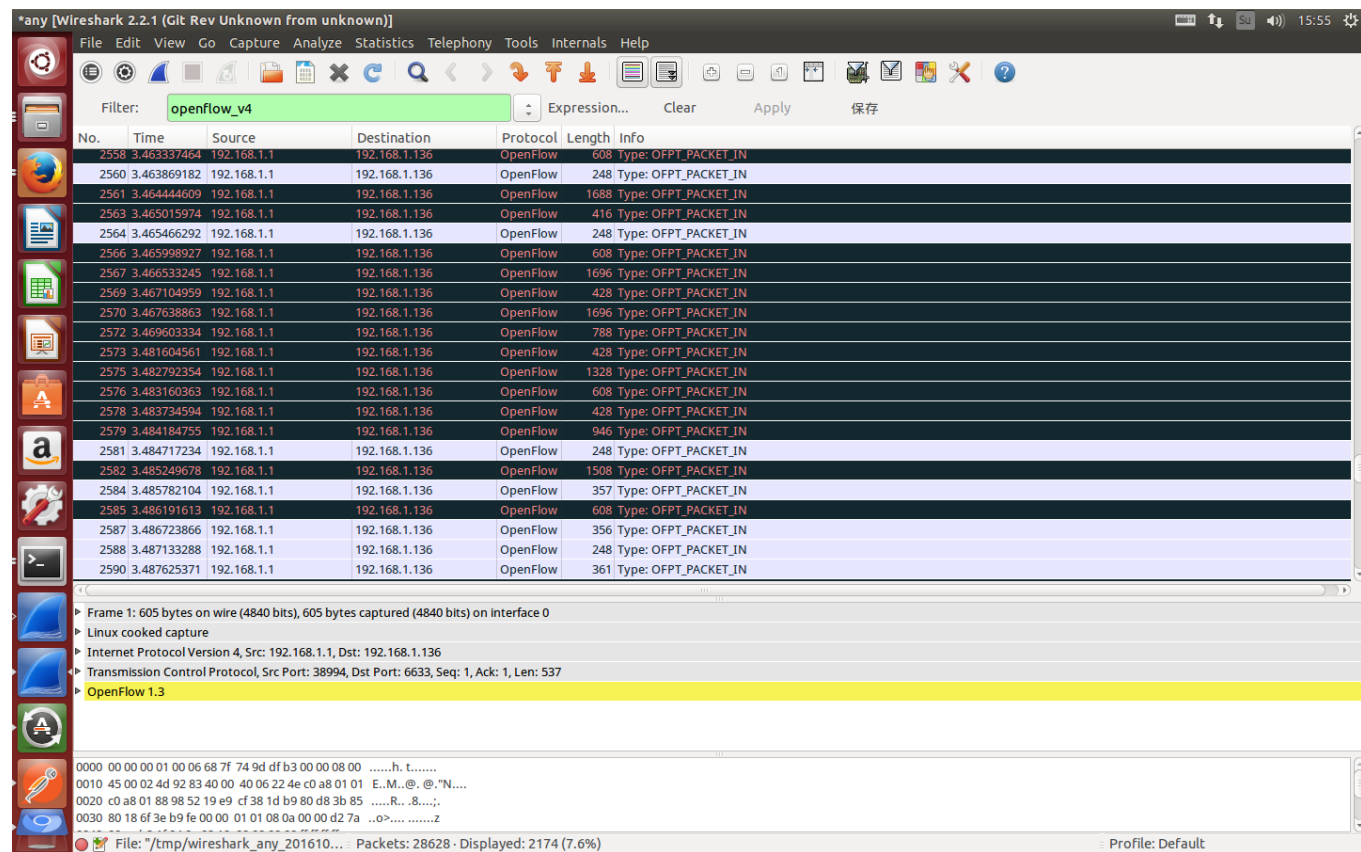
```
root@OpenWrt:~# /etc/init.d/openflow start
eth0.1,eth0.2
Configuring OpenFlow switch for out-of-band control
Oct 14 07:45:25|00001|dp_ports|ERR|eth0.2 device has assigned IP address 211.80.196.152
Oct 14 07:45:25|00002|dp_ports|ERR|eth0.2 device has assigned IPv6 address fe80::6c92:caff:fe74:8141
No need for further configuration for out-of-band control
Oct 14 07:45:28|00001|vlog|INFO|opened log file /tmp/log/ofdatapath.log
Oct 14 07:45:28|00002|secchan|INFO|OpenFlow reference implementation version 1.3.0
Oct 14 07:45:28|00003|secchan|INFO|OpenFlow protocol version 0x04
Oct 14 07:45:28|00004|secchan|WARN|new management connection will receive asynchronous messages
Oct 14 07:45:28|00005|rconn|INFO|tcp:127.0.0.1:6634: connecting...
root@OpenWrt:~# Oct 14 07:45:28|00006|rconn|INFO|tcp:192.168.1.136:6633: connecting...
Oct 14 07:45:28|00007|rconn|INFO|tcp:127.0.0.1:6634: connection failed (Connection refused)
Oct 14 07:45:28|00008|rconn|WARN|tcp:127.0.0.1:6634: connection dropped (Connection refused)
Oct 14 07:45:28|00009|vconn|WARN|tcp:192.168.1.136:6633: extra-long hello:
00000000 04 00 00 10 00 00 00 15-00 01 00 08 00 00 00 12 |.....|

Oct 14 07:45:28|00010|rconn|INFO|tcp:192.168.1.136:6633: connected
Oct 14 07:45:29|00011|rconn|INFO|tcp:127.0.0.1:6634: connecting...
Oct 14 07:45:29|00012|rconn|INFO|tcp:127.0.0.1:6634: connection failed (Connection refused)
Oct 14 07:45:29|00013|rconn|WARN|tcp:127.0.0.1:6634: connection dropped (Connection refused)
Oct 14 07:45:29|00014|rconn|INFO|tcp:127.0.0.1:6634: waiting 2 seconds before reconnect
Oct 14 07:45:32|00015|rconn|INFO|tcp:127.0.0.1:6634: connecting...
Oct 14 07:45:32|00016|rconn|INFO|tcp:127.0.0.1:6634: connection failed (Connection refused)
Oct 14 07:45:32|00017|rconn|WARN|tcp:127.0.0.1:6634: connection dropped (Connection refused)
Oct 14 07:45:32|00018|rconn|INFO|tcp:127.0.0.1:6634: waiting 4 seconds before reconnect
```

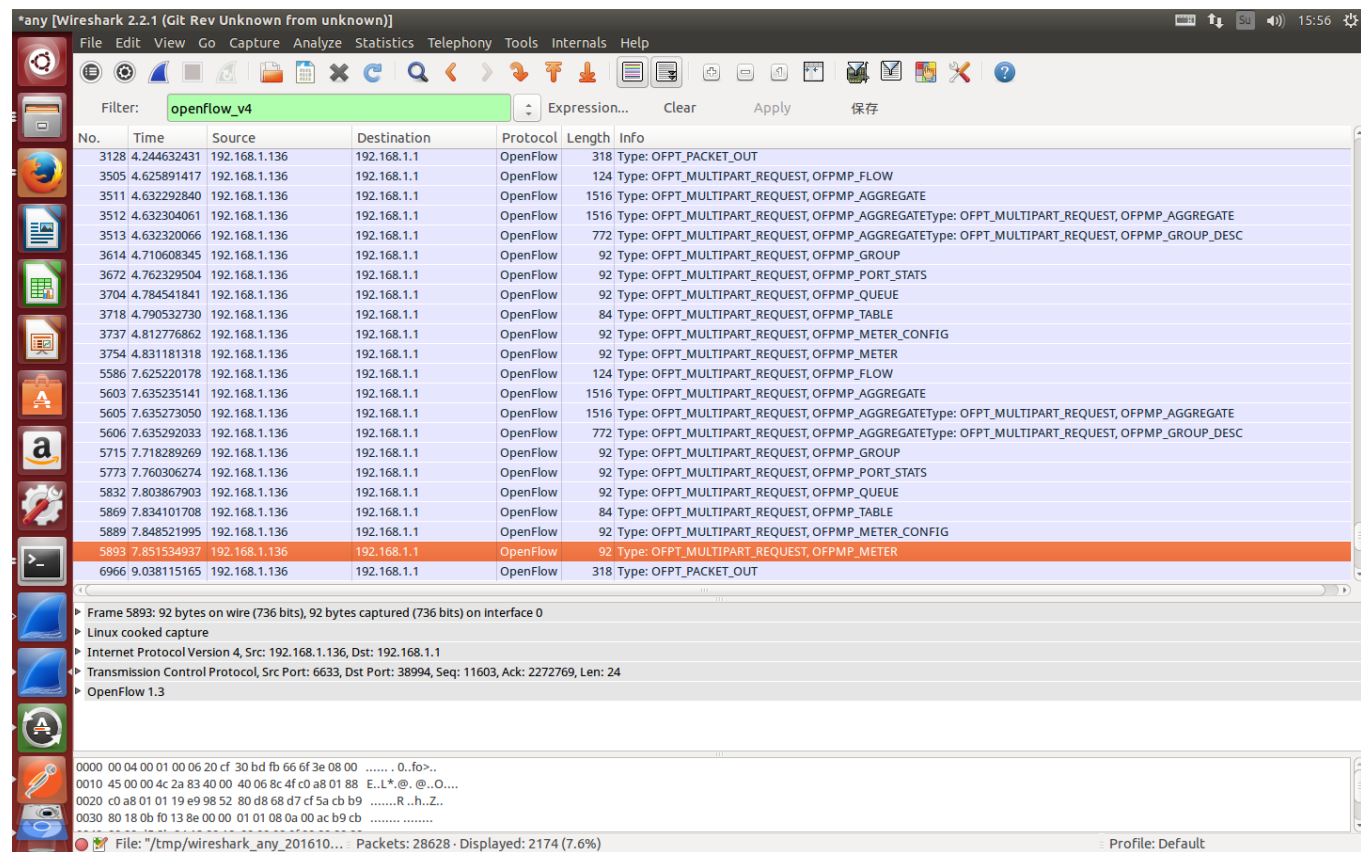
未成功开启

连接tcp:127.0.0.1:6634总是失败。

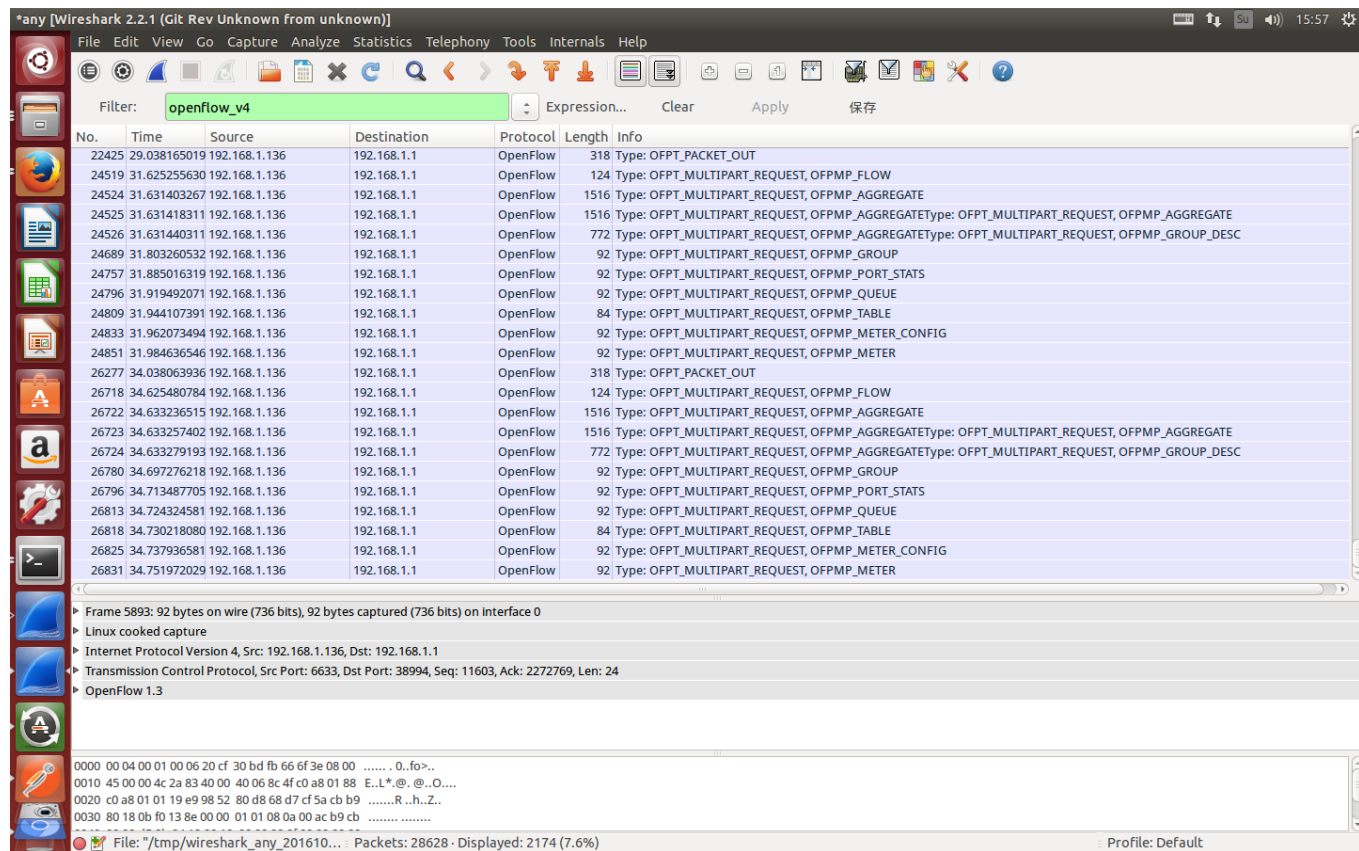
在控制器上用wireshark2.2.1抓包



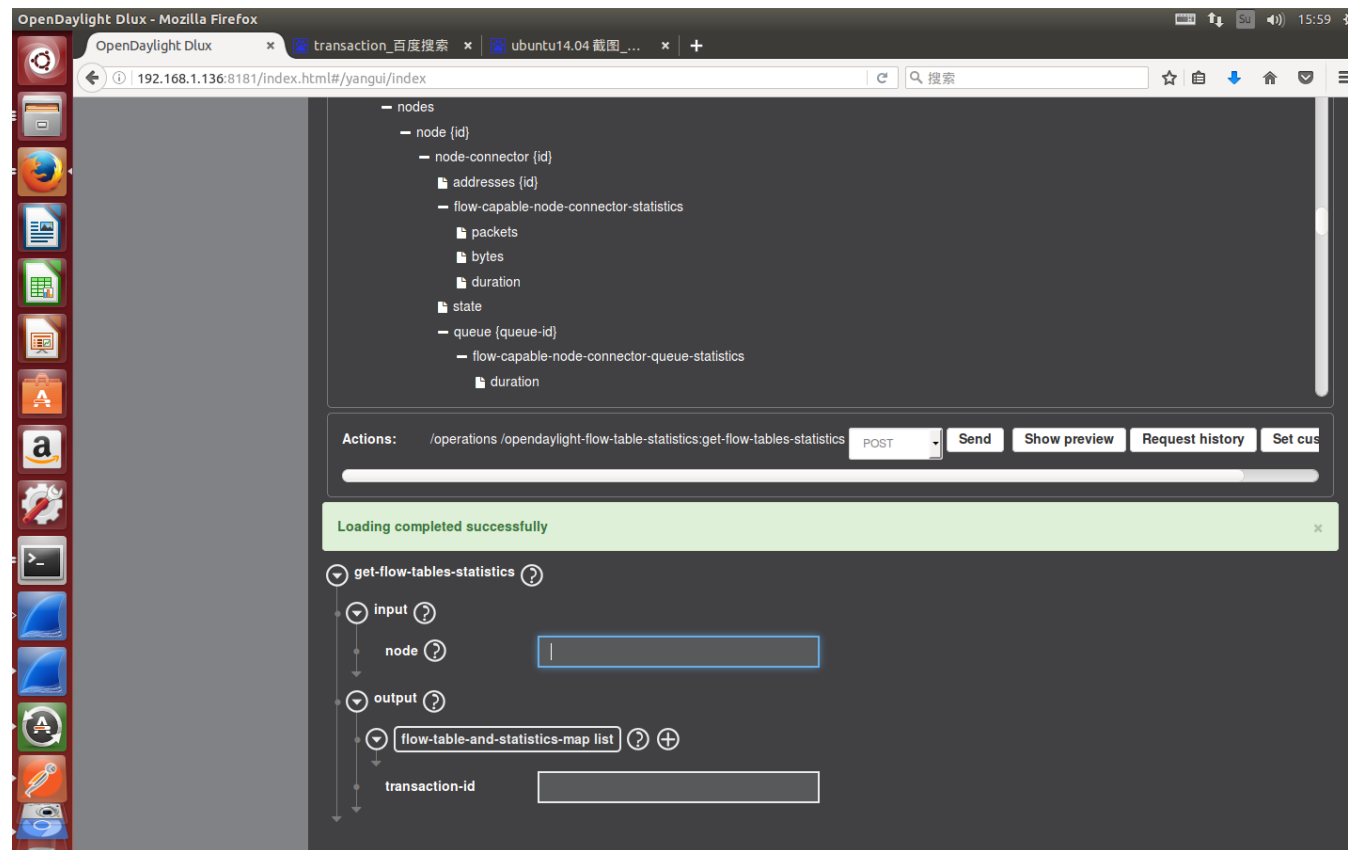
在控制器上用wireshark2.2.1抓包



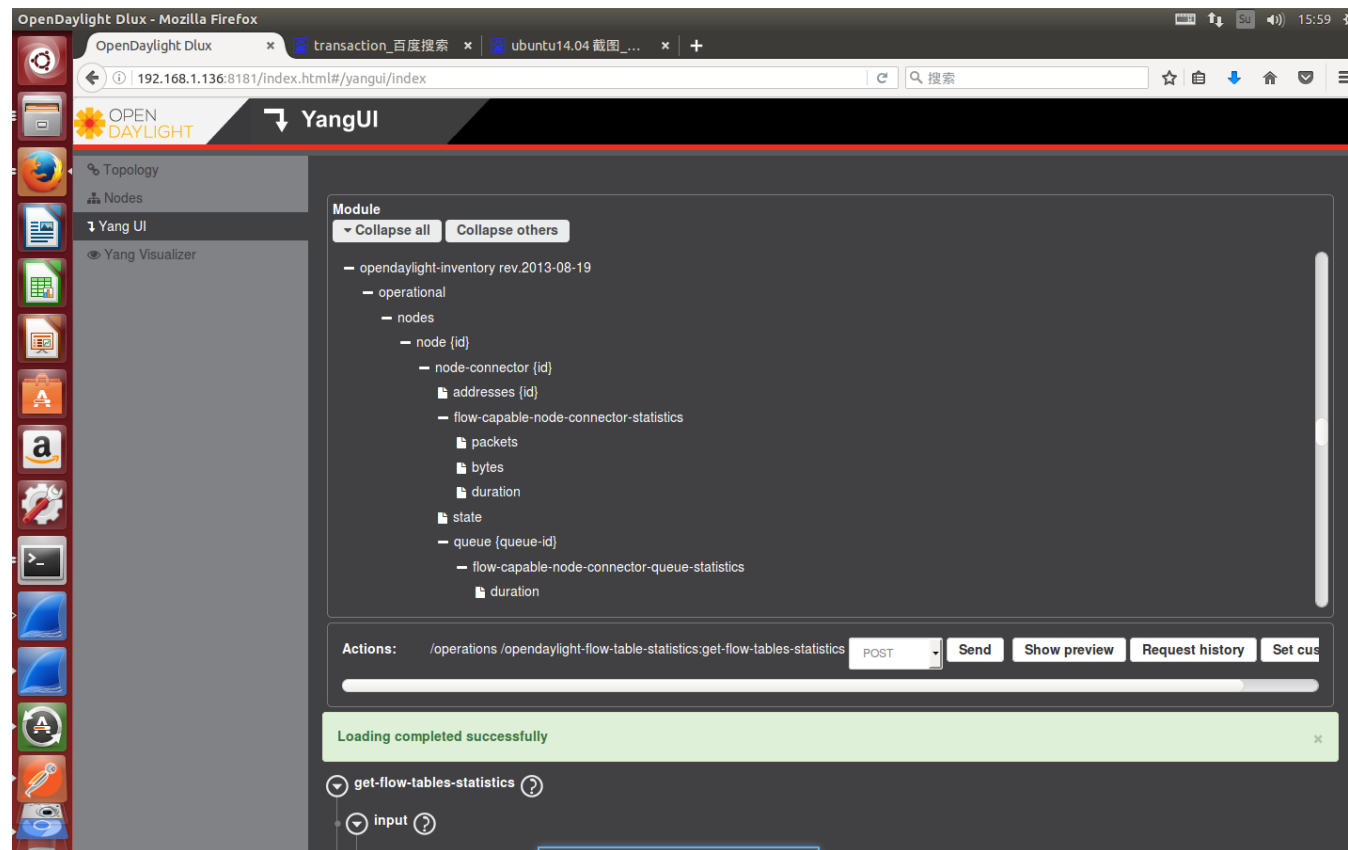
在控制器上用wireshark2.2.1抓包



用OpenDayLight操作流表



用OpenDayLight操作流表




OpenDaylight Dlux

transaction_百度搜索

+

192.168.1.136:8181/index.html#/topology

搜索

 OPEN DAYLIGHT

Topology

Topology


Nodes

Yang UI

Yang Visualizer

Controls

Reload


openflow:1

基于Cisco M20之Openflow交换机制作

目标板：Cisco Valet M20

Chips: BCM4716

RAM: 32M

Flash: 4M

支持OpenWRT路由器信息: <https://wiki.openwrt.org/toh/start#supported.hardware.-.router.type>

安装编译环境

```
#apt-get update
```

```
#apt-get install build-essential binutils flex bison autoconf gettext texinfo sharutils  
subversion libncurses5-dev ncurses-term zlib1g-dev gawk git-core unzip
```

基于DD-WRT制作Openflow交换机

官方固件下载地址: <http://www.dd-wrt.com/site/support/router-database>

ProfessionalSupportCommunityContact

Router DatabaseDocumentationFAQOther Downloads

Router Database

Cisco

(Click into the search field to return to the list)

Cisco Valet M20 v1.0

Router details

ChipsetBCM4716
RAM32 MB
FLASH4 MB

Additional information

- DD-WRT Wiki: Cisco Valet Plus M20
- DD-WRT Wiki: Build variations
- Other downloads

Supported byv24 PreSP2 [Beta] Build: 21061

Description	Filename	Date	Size
K26 mini	dd-wrt.v24-21061_NEWD-2_K2.6_mini.bin	2013-04-19	3,38 MB
K26 mini_hotspot	dd-wrt.v24-21061_NEWD-2_K2.6_mini_hotspot.bin	2013-04-19	3,87 MB
K26 mini_usb	dd-wrt.v24-21061_NEWD-2_K2.6_mini_usb.bin	2013-04-19	3,50 MB
K26 openvpn-small	dd-wrt.v24-21061_NEWD-2_K2.6_openvpn_small.bin	2013-04-22	3,39 MB
K26 std-nokaid-small	dd-wrt.v24-21061_NEWD-2_K2.6_std_nokaid_small.bin	2013-04-22	3,54 MB
K26 voip-small	dd-wrt.v24-21061_NEWD-2_K2.6_voip_small.bin	2013-04-22	3,64 MB

Latest DD-WRT Releases

Remarks

For more information regarding installation and configuration please refer to the following sources:

- DD-WRT Wiki
- DD-WRT User Forum

从源码编译并制作固件

(1) 下载编译工具和源码:

官方工具链和源码下载地址:

<http://www.dd-wrt.com/site/support/other-downloads>


The screenshot shows the 'Other Downloads' section of the DD-WRT website. The navigation bar includes 'Professional', 'Support', and 'Community'. Under 'Support', there are links for 'Router Database', 'Documentation', 'FAQ', and 'Other Downloads' (which is highlighted). The main content area has a sub-header 'Other Downloads' and a paragraph explaining that the router database is the preferred source for firmwares and related files. Below this, the 'Path' is set to 'Downloads > toolchains'. A table lists the contents of the 'toolchains' directory:

Name	Type	Size	Date
..	<DIR>		1970-01-01
toolchains.tar.xz		3.12 GB	2016-08-09

The screenshot shows the 'Other Downloads' section of the DD-WRT website. The navigation bar is the same as the previous screenshot. The 'Path' is set to 'Downloads > others > sourcecode'. A table lists the contents of the 'sourcecode' directory:

Name	Type	Size	Date
..	<DIR>		1970-01-01
v21	<DIR>	2 items	2007-03-18
v22	<DIR>	5 items	2007-03-18
v23	<DIR>	12 items	2007-03-18

V24源码SVN下载地址: <http://svn.dd-wrt.com/browser#src>

 **trac**
Integrated SCM & Project Management

[Login](#) | [Help/Guide](#) | [About Trac](#) | [Preferences](#) | [Register](#)

[Wiki](#) | [Timeline](#) | **[Browse Source](#)** | [Roadmap](#) | [View Tickets](#) | [Search](#)

[Last Change](#) | [Revision Log](#) | [Repository URL](#)


— source:

View revision: View diff against:

Name ▲	Size	Rev	Age	Author	Last Change
▸ ar5315_microredboot		29118 ⚙	8 months	BrainSlayer	some updates
▸ image		1 ⚙	11 years	brainslayer	initial checkin
▸ opt		30620 ⚙	5 weeks	brainslayer	fix
▸ redboot		6159 ⚙	10 years	BrainSlayer	redboot for avila gateworks boards including dd-wrt changes
▸ src		30757 ⚙	5 hours	brainslayer	auto commit of driver changelogs
▸ tools		30172 ⚙	3 months	BrainSlayer	include svg
.cvsignore	6 bytes	1 ⚙	11 years	brainslayer	initial checkin
.project	199 bytes	1 ⚙	11 years	brainslayer	initial checkin
Makefile	717 bytes	1 ⚙	11 years	brainslayer	initial checkin

DD-WRT repository


V24源码github下载地址: <https://github.com/mirror/dd-wrt>

 [mirror / dd-wrt](#) 36 169 131

[Code](#) | [Issues](#) 2 | [Pull requests](#) 0 | [Projects](#) 0 | [Wiki](#) | [Pulse](#) | [Graphs](#)

DD-WRT Subversion mirror <https://www.dd-wrt.com/>

Branch: [master](#) |

 [brainslayer auto commit of driver changelogs](#) ... Latest commit 4827681 5 hours ago

ar5315_microredboot/microredboot	some updates	8 months ago
opt	fix	a month ago
redboot	redboot for avila gateworks boards including dd-wrt changes	10 years ago
src	auto commit of driver changelogs	5 hours ago
tools	include svg	3 months ago
.cvsignore	initial checkin	11 years ago
.project	initial checkin	11 years ago
Makefile	initial checkin	11 years ago

(2) 编译DD-WRT源码

参考网址: http://blog.sina.com.cn/s/blog_626619730100y75q.html

执行三个脚本

调整源码, 编译本地工具和确定编译对象

`#!/ready_ddwrt.sh`

建立编译环境, 提供toolchain和目标文件放置出的链接

`#!/ready_ddwrt_root.sh`

编译v24 std模式

`#!/install_std.v24_2.sh`

编译结束后在/GruppenLW/生成bin目标文件

基于OpenWRT制作openflow交换机

支持OpenWRT路由器信息网址:

<https://wiki.openwrt.org/toh/start#supported.hardware.-.router.type>

#	↓ Brand	Model	Versions	Current Release	Device Page	Device Techdata
213	Cisco	Valet Plus M20	1.0			View/Edit data

```
Device Type: WiFi Router
Brand: Cisco
Model: Valet Plus M20
Version: 1.0
Availability: Available
Unsupported: Possible to support
Platform: Broadcom BCM4716
Target: brcm47xx
Instruction Set: MIPS32
Sub Instruction Set: MIPS32 74K series
CPU MHz: 300
Flash MB: 4
RAM MB: 32
Ethernet 100M ports: -
Ethernet GBit ports: 5
Modem: No
VLAN Hardware: Broadcom (integrated)
VLAN 2.4GHz: b/g/n
VLAN 5.0GHz: -
Detachable Antennas: -
USB ports: -
SATA ports: -
Serial: L
WikiDevi URL: https://wikidevi.com/wiki/Cisco\_Valet\_Plus\_M20
```

尝试刷入OpenWRT固件

OpenWRT官方网址: <http://downloads.openwrt.org/>

Cisco M20相关资源下载网址:

http://downloads.openwrt.org/chaos_calmer/15.05/brcm47xx/mips74k/

Index of /chaos_calmer/15.05/brcm47xx/mips74k/		
File Name ↓	File Size ↓	Date ↓
Parent directory/	-	-
packages/	-	10-Sep-2015 18:39
OpenWrt-ImageBuilder-15.05-brcm47xx-mips74k.Linux-x86_64.tar.bz2	106179913	14-Sep-2015 07:30
OpenWrt-SDK-15.05-brcm47xx-mips74k_gcc-4.8-linaro_uClibc-0.9.33.2.Linux-x86_64.tar.bz2	78385283	14-Sep-2015 07:30
openwrt-15.05-brcm47xx-mips74k-squashfs-gz.trx	4001792	14-Sep-2015 07:31
openwrt-15.05-brcm47xx-mips74k-squashfs-noloader-nodictionary.trx	3674112	14-Sep-2015 07:31
openwrt-15.05-brcm47xx-mips74k-squashfs.trx	3608576	14-Sep-2015 07:31

192.168.1.1/cgi-bin/luci/stok=5c8851317ee2e6fc0106037b5923e019

语音处理 github 学校 数模 openMV 首页 - 我的园子 - ... 公共检索——北创... 基于MATLAB的声...

OpenWrt

Status ▾

System ▾

Network ▾

Logout

Status

System

Hostname	OpenWrt
Model	Cisco M20
Firmware Version	OpenWrt Chaos Calmer 15.05 / LuCI (git-15.248.30277-3836b45)
Kernel Version	3.18.20
Local Time	Fri Oct 14 07:24:49 2016
Uptime	1d 17h 55m 3s
Load Average	0.00, 0.04, 0.15

OpenWRT ImageBuilder生成固件

参考网址: <http://www.cnblogs.com/tanhangbo/p/4559168.html>

(1) 将openflow_1.0.0-1_brcm47xx.ipk拷贝到/packages/base/目录下

查看当前已有的预定义的型号信息 (profile)

#make info

查看命令格式

#make help

(2) 将openflow.ipk编译进固件

#make image PROFILE="TLWR740" PACKAGES="luci-app-firewall luci-base luci-lib-ip luci-lib-nixio luci-mod-admin-full luci-proto-ppp luci-theme-bootstrap luci openflow"

(3) 在bin/brcm47XX/下 可以找到openwrt-15.05-brcm47xx-mips74k-squashfs.trx固件

从源码编译openflow_1.0.0-1_brcm47xx.ipk

参考网址: http://jphome.github.io/blog/2014/03/29/openwrt_sdk.html

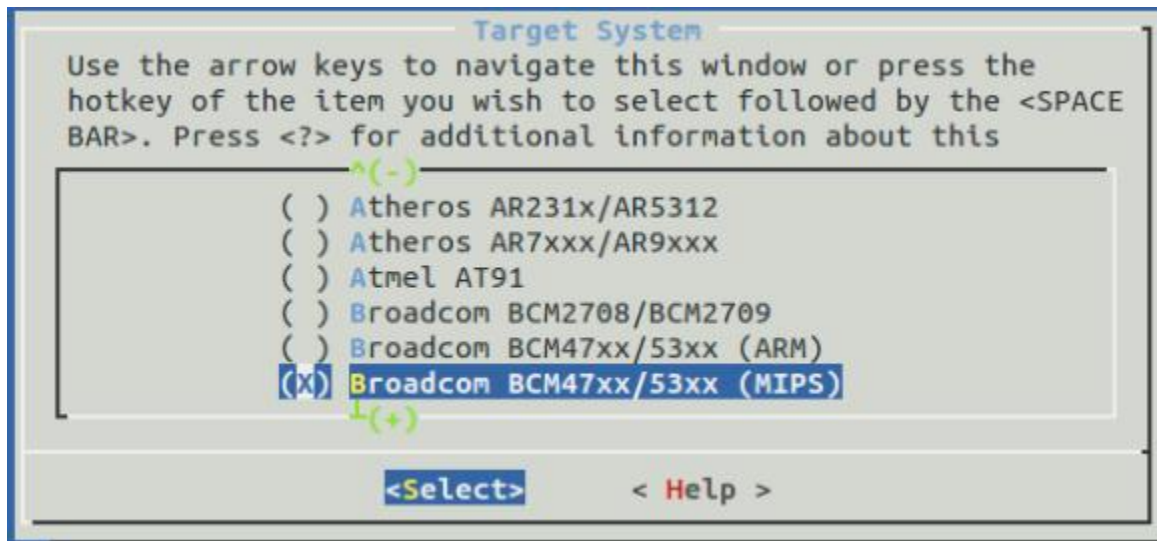
(1) 更新feeds

```
#mv feeds.conf.default feeds.conf  
#./scripts/feeds update -a  
#./scripts/feeds install -a  
#./scripts/feeds install -a luci
```

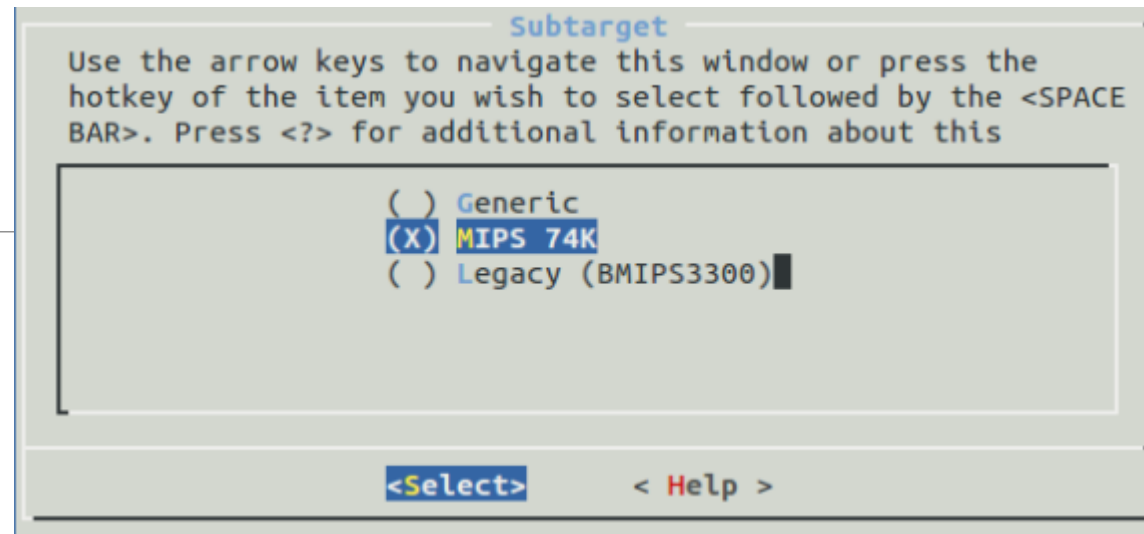
(2) 添加Openflow

```
#git clone https://github.com/CPqD/openflow-openwrt.git  
#cd openwrt/package/  
#ln -s openwrt/openflow-openwrt/openflow-1.3/  
#cd openwrt/  
#ln -s openwrt/openflow-openwrt/openflow-1.3/files
```

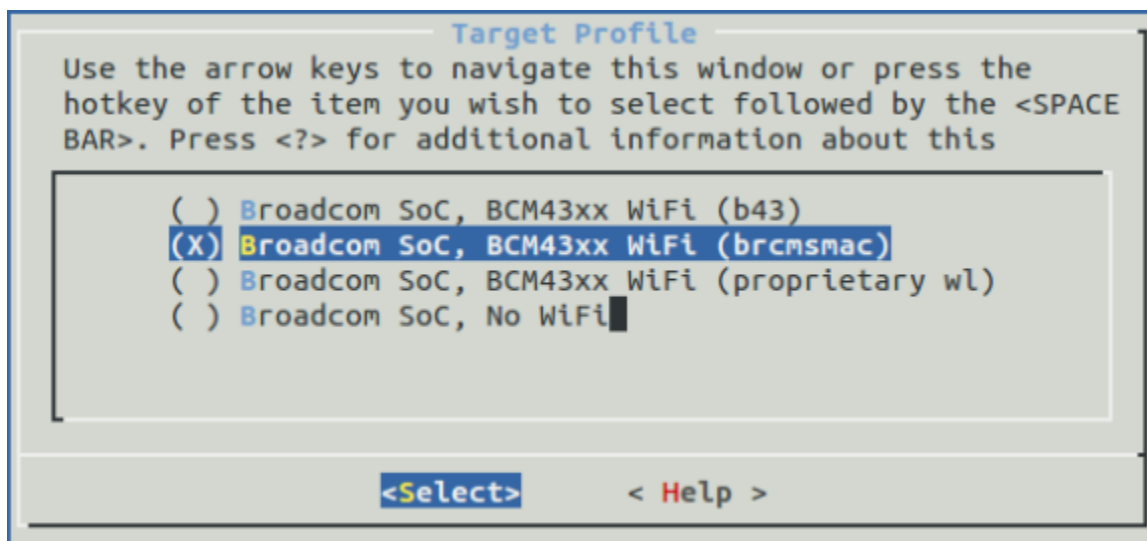
(3) 配置基本信息 #make menuconfig



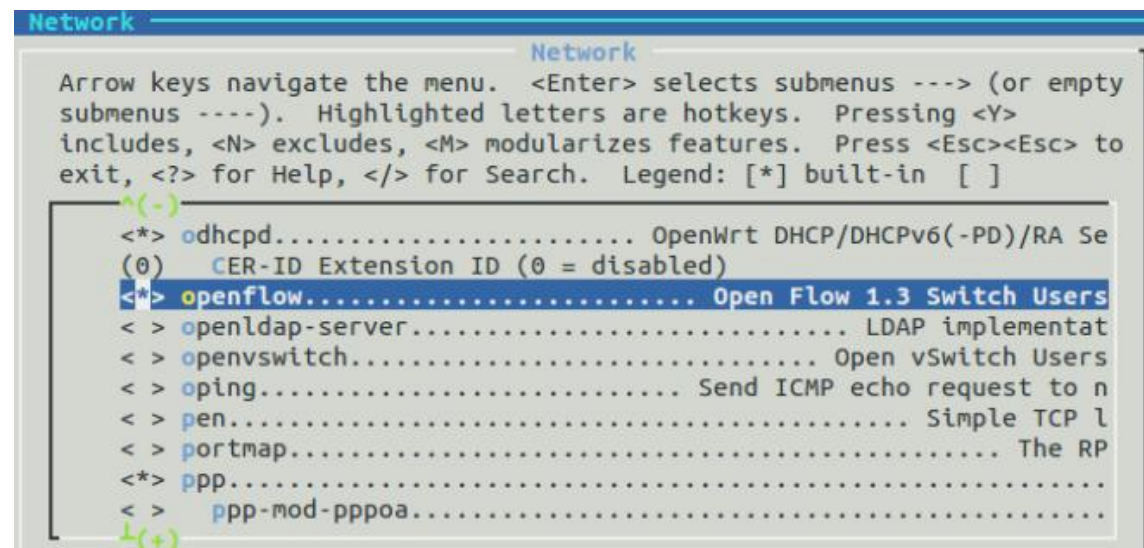
Target System—Broadcom BCM47xx/53xx(MIPS)



Subtarget—MIPS 74K



Target Profile—Broadcom Soc,BCM43xx WiFi(b43)



添加openflow

(4) 编译

#make V=99

执行完后会在bin/brcm47xx/packages/base 生成
openflow_1.0.0-1_brcm47xx

PS:

此时也会在bin/brcm47xx/生成openwrt-15.05-brcm47xx-
mips74k-squashfs.trx固件

使用OpenWRT SDK开发ipk

参考网址: http://jphome.github.io/blog/2014/03/29/openwrt_sdk.html

(1) 在package路径下建立helloworld工程

```
package
├── helloworld
│   ├── Makefile
│   └── src
│       ├── helloworld.c
│       └── Makefile
└── Makefile
```

(2) 建立staging_dir链接(交叉编译工具)

```
cd sdk
ln -s ../trunk/staging_dir ./staging_dir
```

(3) 编译

```
make distclean
make menuconfig
make V=99
```

```
bin
├── brcm63xx
│   └── packages
│       ├── helloworld_1.0-1_brcm63xx.ipk
│       ├── Packages
│       └── Packages.gz
└── packages
```

刷机经验

192.168.1.1/cgi-bin/luci/stok=5c8851317ee2e6fc0106037b5923e019/admin/system/flashops

语音处理github学校数模openMV首页 - 我的园子 - ...公共检索——北创...基于MATLAB的声...matlab 语音 延时...实时音频变调算法..

OpenWrtStatusSystemNetworkLogout

Flash operations

ActionsConfiguration

Backup / Restore

Click "Generate archive" to download a tar archive of the current configuration files. To reset the firmware to its initial state, click "Perform reset" (only possible with squashfs images).

Download backup:Generate archive

Reset to defaults:Perform reset

To restore configuration files, you can upload a previously generated backup archive here.

Restore backup:选择文件未选择任何文件Upload archive...

Flash new firmware image

Upload a sysupgrade-compatible image here to replace the running firmware. Check "Keep settings" to retain the current configuration (requires an OpenWrt compatible firmware image).

Keep settings:☒

Image:选择文件未选择任何文件Flash image...

无法登录web页面（luci）如何刷入固件

参考网址：<http://blog.csdn.net/zyphio/article/details/8547365>

一、window方法

（1）用HFS建立路由器固件文件的本地服务器

- 1.打开HFS，进入主菜单“Menu”的“IP address”子菜单；
- 2.勾选内网IP，一般内网IP是“192.168.X.X”的形式，即路由分配给当前网卡的IP；
- 3.在“Virtual File System”（虚拟文件系统）树状目录框中点鼠标右键弹出快捷菜单；
- 4.选择“添加文件”，打开本地存储器路由器固件“rf.bin”，看到“rf.bin”出现在“Virtual File System”（虚拟文件系统）树状目录框中即表示已添加；
- 5.可以在“Virtual File System”（虚拟文件系统）树状目录框中的“rf.bin”上点鼠标右键弹出快捷菜单，选择“浏览”，如果网络浏览器弹出窗口显示下载“rf.bin”文件，即表示本地固件文件服务器成功运行。

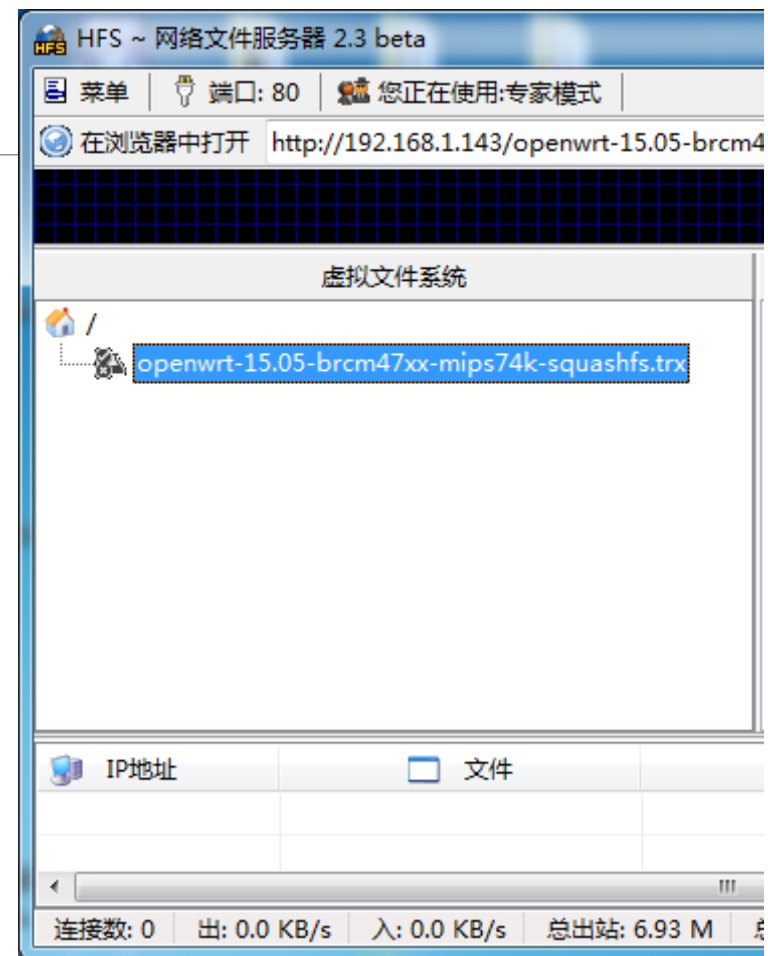
（2）telnet 或 SSH 登录（由于没有密码SSH无法登录）

```
#telnet 192.168.1.1
```

```
#cd /tmp
```

```
#wget http://192.168.1.143/xxx.bin(192.168.1.143 地址为路由器给电脑分配的ip)
```

```
#mtd -r write xxx.bin firmware (firmware或linux、OS1 这里是firmware)
```



二、linux方法

```
#scp xxx.bin root@192.168.1.1:/tmp(刷固件)
```

```
#scp xxx.ipk root@192.168.1.1:/tmp (刷入ipk)
```

```
#ssh root@192.168.1.1
```

刷固件

```
#wget http://192.168.1.143/xxx.bin
```

```
#mtd -r write xxx.bin firmware
```

刷ipk

```
#opkg install xxx.ipk (由于flash太小 没有将openflow装成功)
```

ps:刷新固件OpenSSH会核对公钥，导致无法登陆ssh，执行一下命令重新认证。

```
#rm -rf ~/.ssh/known_hosts
```

配置Openflow

相关脚本

```
#vim /etc/config/network
```

```
#vim /etc/config/wireless
```

```
#vim /etc/config/openflow
```

```
#vim /lib/openflow/ofswitch.sh
```

刚更新OpenWRT固件默认只有eth0.1和eth0.2,设置控制器IP

```
config 'ofswitch'
```

```
    option 'dp' 'dp0'
```

```
    option 'dpid' '0000000000001'
```

```
    option 'ofports' 'eth0.1 eth0.2'
```

```
    option 'ofctl' 'tcp:192.168.1.136:6633'
```

```
    option 'mode' 'outofband'
```

#/etc/init.d/openflow start

```
root@openwrt:/lib/openflow# /etc/init.d/openflow restart
eth0.1,eth0.2
Configuring OpenFlow switch for out-of-band control
Oct 14 03:01:58|00001|dp_ports|ERR|eth0.2 device has assigned IP address 211.80.196.51
Oct 14 03:01:58|00002|dp_ports|ERR|eth0.2 device has assigned IPv6 address 2001:250:6803:2304:6a7f:74ff:fe9d:dfb3
No need for further configuration for out-of-band control
root@openwrt:/lib/openflow# Oct 14 03:02:01|00001|vlog|INFO|opened log file /tmp/log/ofdatapath.log
Oct 14 03:02:01|00002|secchan|INFO|openFlow reference implementation version 1.3.0
Oct 14 03:02:01|00003|secchan|INFO|openFlow protocol version 0x04
Oct 14 03:02:01|00004|secchan|WARN|new management connection will receive asynchronous messages
Oct 14 03:02:01|00005|rconn|INFO|tcp:127.0.0.1:6634: connecting...
Oct 14 03:02:01|00006|rconn|INFO|tcp:192.168.1.136:6633: connecting...
Oct 14 03:02:01|00007|rconn|INFO|tcp:127.0.0.1:6634: connected
Oct 14 03:02:01|00008|vconn|WARN|tcp:192.168.1.136:6633: extra-long hello:
00000000 04 00 00 10 00 00 00 15-00 01 00 08 00 00 00 12 |.....|

Oct 14 03:02:01|00009|rconn|INFO|tcp:192.168.1.136:6633: connected
Oct 14 03:02:01|00010|port_watcher|INFO|Datapath id is 0000000000001
8 800

Oct 14 03:02:09|00003|dp_acts|WARN|can't directly forward to input port.
```

#/etc/init.d/openflow stop

<https://sourceforge.net/p/ryu/mailman/message/31853841/>

<https://forum.openwrt.org/viewtopic.php?id=43742>

I had the same issue with openflow 1.0
I fix it by editing the file : /lib/openflow/ofswitch.sh on my openwrt router.
And add the --no-slicing parameter.

```
echo "Configuring OpenFlow switch for out-of-band control"
```

```
[ -n "$dpid" ] && {
```

```
ofdatapath punix:/var/run/dp0.sock -i "$dpports" --no-slicing --no-local-port "--pidfile=$pidfile" -d "$dpid" &  
} || {
```

```
ofdatapath punix:/var/run/dp0.sock -i "$dpports" --no-slicing --no-local-port "--pidfile=$pidfile" &  
}
```

11月2日

关于wifi

情况：OpenWRT系统里的各种配置文件以及web上的配置看似正常，但是路由器并未发出wifi信号。但是之前用别人做好的OpenWRT系统wifi就正常。

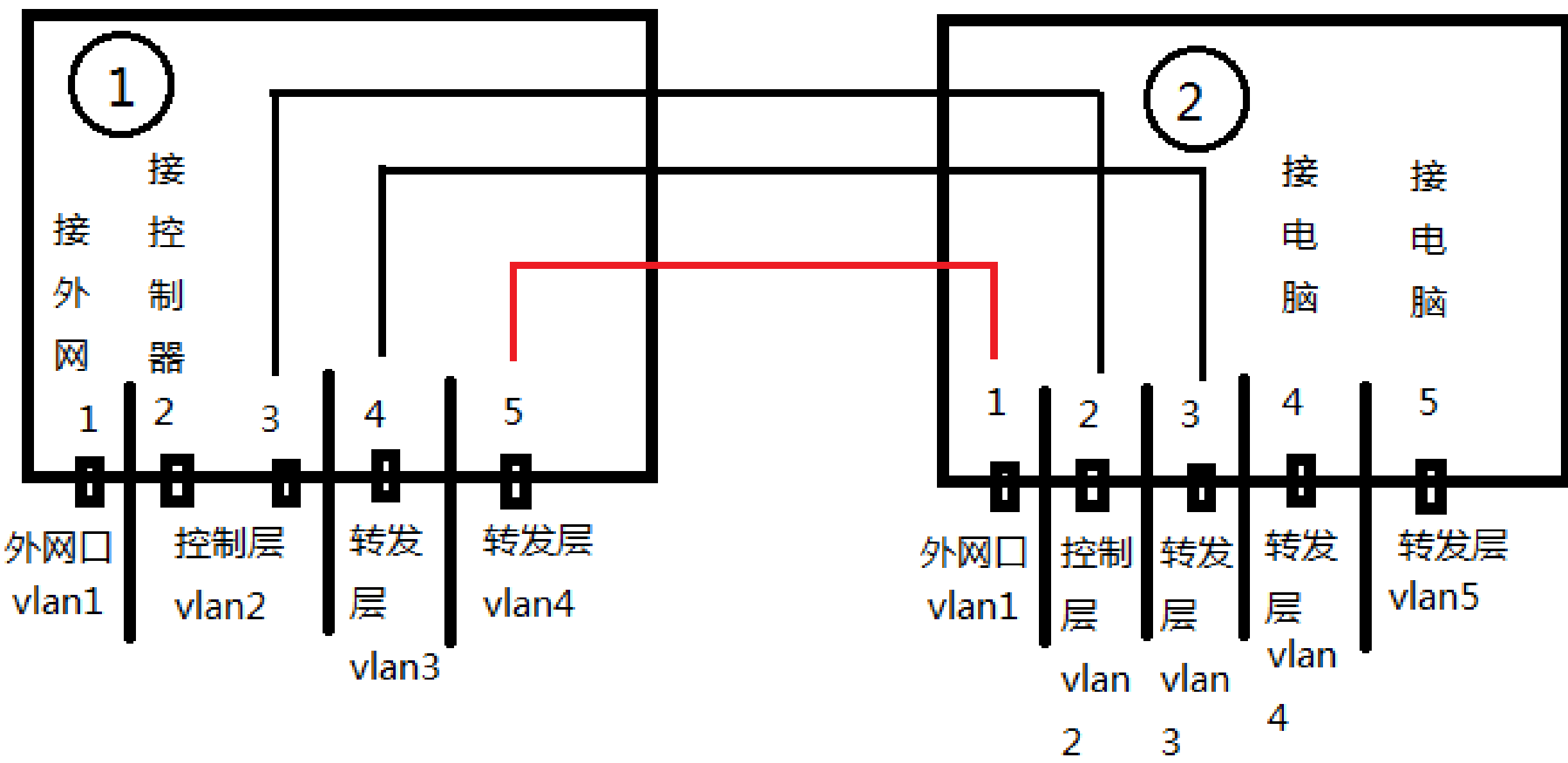
可能的原因：编译系统的时候某些必要的wifi相关的模块没有编译进去。

解决方案：

1. 重新编译，尝试用各种方法多加入一些wifi模块。
2. 在别人做好的wifi正常的OpenWRT系统中加入openflow功能。

多台openflow交换机

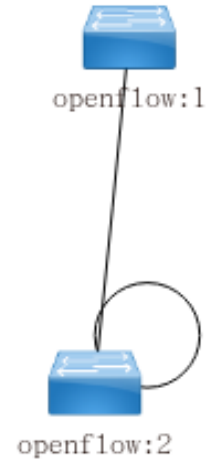
已经成功搭建两台交换机的openflow网络，与之前的两台交换机不同的是，为了可以通过流表管理，将转发层的每个端口分为一个vlan。



将每个口分为一个vlan后控制器上的拓扑中会出现自环，并且连接在交换机2上的电脑不能上网。

Controls

Reload



猜想：网络中有两个交换机，且为每个端口分一个vlan，这样使得网络太复杂，交换机不知道包如何处理。

应对：手动下发流表，使交换机1上的所有包发往1、4、5口，交换机2上的所有包发往1、3、4、5口。

结果：交换机2上的电脑不再是完全不能上网了，变为了网络极其不稳定，偶尔可以连通一下网。并且交换机终端的openflow中显示丢包严重。

```
ubuntu@ubuntu-System-Product-Name: ~  
Nov 01 07:52:55|00192|dp_acts|WARN|Dropped 2030 messages in last 1 seconds due t  
o excessive rate  
Nov 01 07:52:56|00193|dp_acts|WARN|can't directly forward to input port.  
Nov 01 07:52:56|00194|dp_acts|WARN|Dropped 2207 messages in last 1 seconds due t  
o excessive rate  
Nov 01 07:52:57|00195|dp_acts|WARN|can't directly forward to input port.  
Nov 01 07:52:57|00196|dp_acts|WARN|Dropped 2202 messages in last 1 seconds due t  
o excessive rate  
Nov 01 07:52:58|00197|dp_acts|WARN|can't directly forward to input port.  
Nov 01 07:52:58|00198|dp_acts|WARN|Dropped 2078 messages in last 1 seconds due t  
o excessive rate  
Nov 01 07:52:59|00199|dp_acts|WARN|can't directly forward to input port.  
Nov 01 07:52:59|00200|dp_acts|WARN|Dropped 2239 messages in last 1 seconds due t  
o excessive rate  
Nov 01 07:53:00|00201|dp_acts|WARN|can't directly forward to input port.  
Nov 01 07:53:00|00202|dp_acts|WARN|Dropped 2218 messages in last 1 seconds due t  
o excessive rate  
Nov 01 07:53:01|00203|dp_acts|WARN|can't directly forward to input port.  
Nov 01 07:53:01|00204|dp_acts|WARN|Dropped 2119 messages in last 1 seconds due t  
o excessive rate  
Nov 01 07:53:02|00205|dp_acts|WARN|can't directly forward to input port.  
Nov 01 07:53:02|00206|dp_acts|WARN|Dropped 2287 messages in last 1 seconds due t  
o excessive rate  
Nov 01 07:52:56|00193|dp_acts|WARN|can't directly forward to input port.  
Nov 01 07:52:56|00194|dp_acts|WARN|Dropped 2079 messages in last 1 seconds due t  
o excessive rate  
Nov 01 07:52:57|00195|dp_acts|WARN|can't directly forward to input port.  
Nov 01 07:52:57|00196|dp_acts|WARN|Dropped 1827 messages in last 1 seconds due t  
o excessive rate  
Nov 01 07:52:58|00197|dp_acts|WARN|can't directly forward to input port.  
Nov 01 07:52:58|00198|dp_acts|WARN|Dropped 1583 messages in last 1 seconds due t  
o excessive rate  
Nov 01 07:52:59|00199|dp_acts|WARN|can't directly forward to input port.  
Nov 01 07:52:59|00200|dp_acts|WARN|Dropped 1820 messages in last 1 seconds due t  
o excessive rate  
Nov 01 07:53:00|00201|dp_acts|WARN|can't directly forward to input port.  
Nov 01 07:53:00|00202|dp_acts|WARN|Dropped 1775 messages in last 1 seconds due t  
o excessive rate  
Nov 01 07:53:01|00203|dp_acts|WARN|can't directly forward to input port.  
Nov 01 07:53:01|00204|dp_acts|WARN|Dropped 1466 messages in last 1 seconds due t  
o excessive rate  
Nov 01 07:53:02|00205|dp_acts|WARN|can't directly forward to input port.  
Nov 01 07:53:02|00206|dp_acts|WARN|Dropped 1683 messages in last 1 seconds due t  
o excessive rate
```

```
ubuntu@ubuntu-System-Product-Name: ~  
Nov 01 07:48:25|01113|dp_acts|WARN|Dropped 1978 messages in last 1 seconds due t  
o excessive rate  
Nov 01 07:48:26|01114|dp_acts|WARN|can't directly forward to input port.  
Nov 01 07:48:26|01115|dp_acts|WARN|Dropped 1984 messages in last 1 seconds due t  
o excessive rate  
Nov 01 07:48:27|01116|dp_acts|WARN|can't directly forward to input port.  
Nov 01 07:48:27|01117|dp_acts|WARN|Dropped 1878 messages in last 1 seconds due t  
o excessive rate  
Nov 01 07:48:28|01118|dp_acts|WARN|can't directly forward to input port.  
Nov 01 07:48:28|01119|dp_acts|WARN|Dropped 1996 messages in last 1 seconds due t  
o excessive rate  
Nov 01 07:48:29|01120|dp_acts|WARN|can't directly forward to input port.  
Nov 01 07:48:29|01121|dp_acts|WARN|Dropped 1999 messages in last 1 seconds due t  
o excessive rate  
Nov 01 07:48:30|01122|dp_acts|WARN|can't directly forward to input port.  
Nov 01 07:48:30|01123|dp_acts|WARN|Dropped 1915 messages in last 1 seconds due t  
o excessive rate  
Nov 01 07:48:31|01124|dp_acts|WARN|can't directly forward to input port.  
Nov 01 07:48:31|01125|dp_acts|WARN|Dropped 1906 messages in last 1 seconds due t  
o excessive rate  
Nov 01 07:48:32|01126|dp_acts|WARN|can't directly forward to input port.  
Nov 01 07:48:32|01127|dp_acts|WARN|Dropped 1715 messages in last 1 seconds due t  
o excessive rate
```

猜想：下发的流表要求的工作量太大，导致交换机工作中出现力不从心的现象。

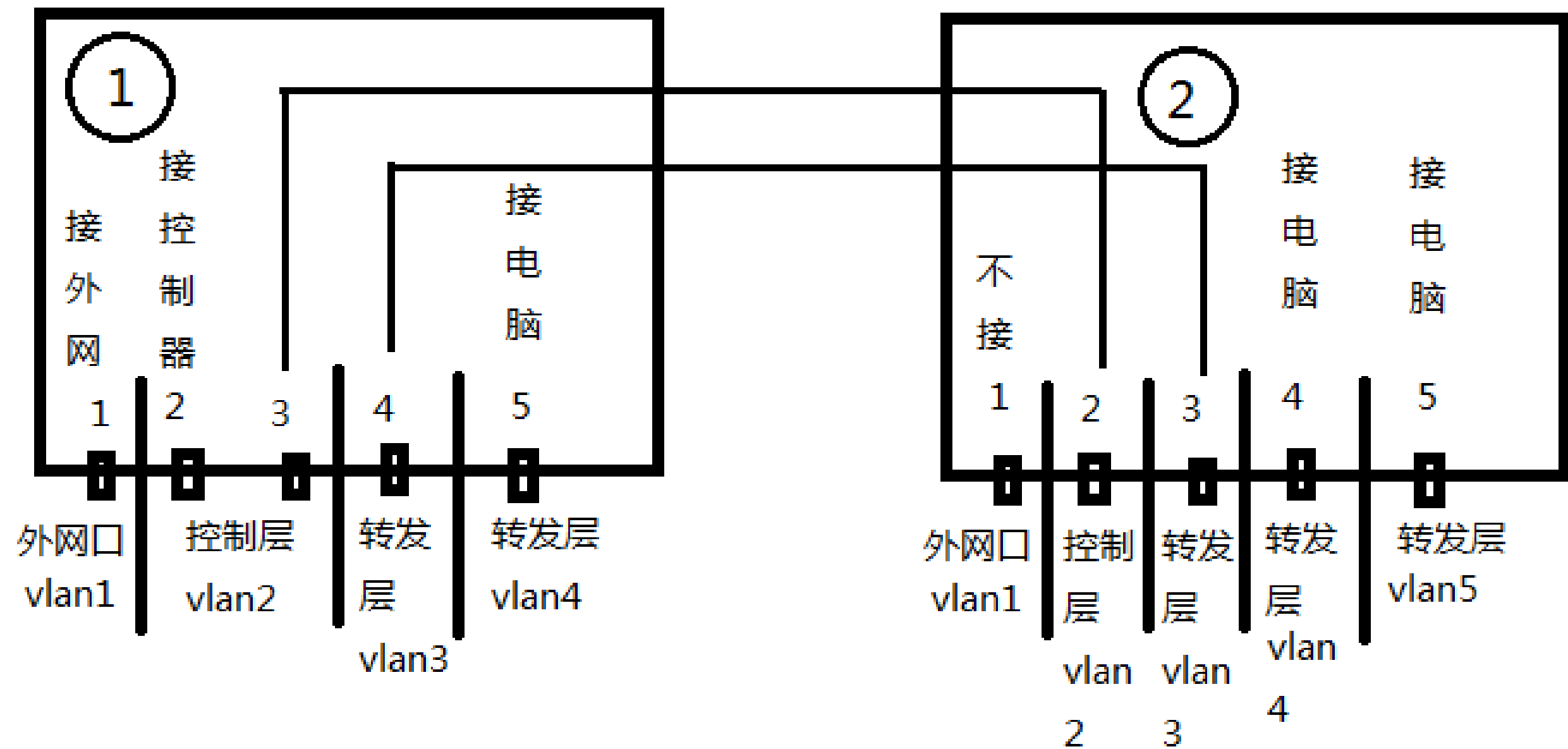
应对：由于下发流表之前只是交换机2上的电脑不能上网，所以只在交换机2上下发流表，并且理论上讲交换机2不需要1口收发数据。先删除之前下发的流表，然后在交换机2上下发流表，让所有的包发往3、4、5口。

结果：交换机2上的电脑可以正常上网上，但是自环还在。

猜想：之前交换机2的外网口需要连接一根网线来获取ip地址，否则交换机1和交换机2上的电脑都不能上网，可能也是因为网络太过复杂，交换机不知道如何处理包导致的。

尝试：在之前下发流表的基础上，不在交换机2的外网口连接网线。

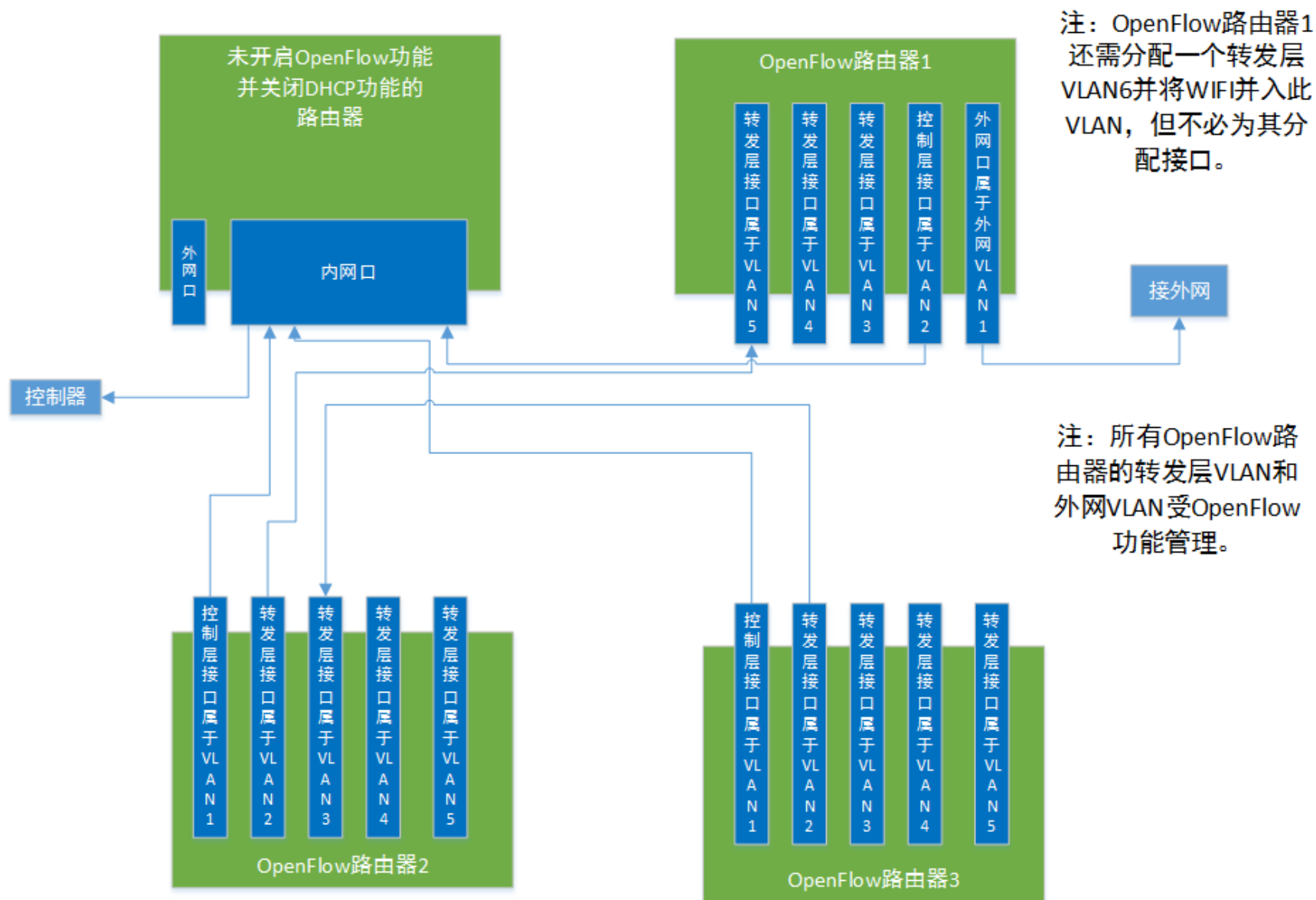
结果：交换机1和交换机2上的电脑都可以正常上网。且拓扑图中的自环消失。



11月10日

概述

1. 与外网相连的路由器用思科的就不容易掉线了。
2. WIFI功能思科的路由器没有问题。
3. 具体网络接法如下页图，OpenFlow路由器2与OpenFlow路由器3需要用OpenDayLight下发流表，使所有包都发往每一个由OpenFlow功能管理的端口。



注：OpenFlow路由器1
还需分配一个转发层
VLAN6并将WIFI并入此
VLAN，但不必为其分
配接口。

注：所有OpenFlow路
由器的转发层VLAN和
外网VLAN受OpenFlow
功能管理。

注：OpenFlow路由器2
还需分配一个VLAN6
作为外网VLAN但不必
为其分配接口。

注：OpenFlow路由器3
还需分配一个VLAN6
作为外网VLAN但不必
为其分配接口。

网速

1. 有线网浏览网页SDN略快。
2. 无线网浏览网页SDN略快。
3. 用迅雷下载同一部电影，SDN网速200KBPS，传统网络网速2.7MBPS，并且下载时传统网络浏览网页速度基本不变，SDN基本无法浏览网页。
4. 玩游戏SDN性能稍好。

猜测：下发流表使每一个包都要向所有的转发层VLAN发一次，导致无谓的占用带宽。

11月23日

实时流量监控

OpenDayLight Lithium版本带有实时流量监控功能。

Node Connector Statistics for Node Id - openflow:1												
Node Connector Id	Rx Pkts	Tx Pkts	Rx Bytes	Tx Bytes	Rx Drops	Tx Drops	Rx Errs	Tx Errs	Rx Frame Errs	Rx OverRun Errs	Rx CRC Errs	Collisions
openflow:1:2	0	1253979	0	890094778	0	0	0	0	0	0	0	0
openflow:1:1	1238198	16815	888447758	2026086	0	0	0	0	0	0	0	0
openflow:1:4	0	1253979	0	890094778	0	0	0	0	0	0	0	0
openflow:1:3	1497	1252481	280987	889813706	0	0	0	0	0	0	0	0
openflow:1:6	14344	1239557	1662309	888392704	0	76	0	0	0	0	0	0
openflow:1:5	0	1253979	0	890094778	0	0	0	0	0	0	0	0

端口查看

只有需要看openflow:1:2, openflow:1:4, openflow:1:6。因为1是外网，3是连接别的openflow路由器的，5是为了能用openflow管理wifi而额外添加的。2和4是连电脑的，6是wifi。

Node Connector Id	Name	Port Number	Mac Address
openflow:1:2	eth0.3	2	68:7F:74:9D:DF:B3
openflow:1:1	eth0.2	1	68:7F:74:9D:DF:B3
openflow:1:4	eth0.5	4	68:7F:74:9D:DF:B3
openflow:1:3	eth0.4	3	68:7F:74:9D:DF:B3
openflow:1:6	wlan0	6	68:7F:74:9D:DF:B5
openflow:1:5	eth0.6	5	68:7F:74:9D:DF:B3

刷新一下

Node Connector Statistics for Node Id - openflow:1

Node Connector Id	Rx Pkts	Tx Pkts	Rx Bytes	Tx Bytes	Rx Drops	Tx Drops	Rx Errs	Tx Errs	Rx Frame Errs	Rx OverRun Errs	Rx CRC Errs	Collisions
openflow:1:2	0	1257412	0	891238846	0	0	0	0	0	0	0	0
openflow:1:1	1241625	16821	889591303	2026609	0	0	0	0	0	0	0	0
openflow:1:4	0	1257412	0	891238846	0	0	0	0	0	0	0	0
openflow:1:3	1497	1255914	280987	890957774	0	0	0	0	0	0	0	0
openflow:1:6	14345	1242989	1662407	889536674	0	76	0	0	0	0	0	0
openflow:1:5	0	1257412	0	891238846	0	0	0	0	0	0	0	0

打开视频的测试

打开视频前：

Node Connector Statistics for Node Id - openflow:1												
Node Connector Id	Rx Pkts	Tx Pkts	Rx Bytes	Tx Bytes	Rx Drops	Tx Drops	Rx Errs	Tx Errs	Rx Frame Errs	Rx OverRun Errs	Rx CRC Errs	Collisions
openflow:1:2	0	5636	0	2404760	0	0	0	0	0	0	0	0
openflow:1:1	5107	1226	2501898	122386	0	0	0	0	0	0	0	0
openflow:1:4	0	5636	0	2404760	0	0	0	0	0	0	0	0
openflow:1:3	0	5636	0	2404760	0	0	0	0	0	0	0	0
openflow:1:6	1221	4293	121961	2208894	0	120	0	0	0	0	0	0
openflow:1:5	0	5636	0	2404760	0	0	0	0	0	0	0	0

打开视频后

Node Connector Statistics for Node Id - openflow:1

Node Connector Id	Rx Pkts	Tx Pkts	Rx Bytes	Tx Bytes	Rx Drops	Tx Drops	Rx Errs	Tx Errs	Rx Frame Errs	Rx OverRun Errs	Rx CRC Errs	Collisions
openflow:1:2	0	28352	0	15195997	0	0	0	0	0	0	0	0
openflow:1:1	23215	5834	14969737	445784	0	0	0	0	0	0	0	0
openflow:1:4	0	28352	0	15195997	0	0	0	0	0	0	0	0
openflow:1:3	0	28352	0	15195997	0	0	0	0	0	0	0	0
openflow:1:6	5809	22421	443659	14678433	0	120	0	0	0	0	0	0
openflow:1:5	0	28352	0	15195997	0	0	0	0	0	0	0	0

通过不断刷新，可以发现打开视频的时候Rx Types这一项变化很快，关掉视频后Rx Types这一项变化很慢。所以可以通过Rx Types这一项的变化速度来判断这台设备是否打开了视频。

下一页为该页面的代码。

```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="utf-8" />
5     <title>OpenDaylight Dlux</title>
6
7     <meta name="description" content="overview & stats" />
8     <meta name="viewport" content="width=device-width, initial-scale=1.0" />
9     <script type="text/javascript">
10 var module = ['angularAMD', 'ocLazyLoad', 'angular-ui-router', 'angular-translate', 'angular-translate-loader-static-files', 'common/config/env.module', 'angular-css-
injector', 'app/core/core.module', 'common/login/login.module', 'common/authentication/auth.module', 'common/navigation/navigation.module', 'common/topbar/topbar.module', 'common/general/common.general.module', 'commo
n/layout/layout.module', 'app/topology/topology.module', 'app/node/nodes.module', 'app/yangui/yangui.module', 'common/yangutils/yangutils.module', 'app/yangvisualizer/yangvisualizer.module', 'common/sigmatopology/sigma
topology.module'];
11 var e = ['oc.lazyLoad', 'ui.router', 'pascalprecht.translate',
'angular.css.injector', 'app.core', 'app.common.login', 'app.common.auth', 'app.common.nav', 'app.common.topbar', 'app.common.general', 'app.common.layout', 'app.topology', 'app.nodes', 'app.yangui', 'app.common.yangUtils',
'app.yangvisualizer', 'app.common.sigmatopology'];
12 // global variables
13
14 </script>
15
16 <!-- HTML5 shim and Respond.js IE8 support of HTML5 elements and media queries -->
17
18 <!--[if lt IE 9]>
19 <script src="assets/js/html5shiv.js"></script>
20 <script src="assets/js/respond.min.js"></script>
21 <![endif]>
22
23 <!-- compiled CSS -->
24 <link rel="stylesheet" type="text/css" href="vendor/ng-grid/ng-grid.min.css" />
25 <link rel="stylesheet" type="text/css" href="vendor/select2-bootstrap-css/select2-bootstrap.css" />
26 <link rel="stylesheet" type="text/css" href="vendor/footable/css/footable.core.min.css" />
27 <link rel="stylesheet" type="text/css" href="vendor/footable/css/footable.standalone.min.css" />
28 <link rel="stylesheet" type="text/css" href="vendor/vis/dist/vis.min.css" />
29 <link rel="stylesheet" type="text/css" href="vendor/ng-slider/dist/css/ng-slider.min.css" />
30 <link rel="stylesheet" type="text/css" href="assets/opendaylight-dlux-0.1.0.css" />
31 <link rel="stylesheet" href="assets/css/sb-admin.css" />
32
33 <script type="text/javascript" data-main="src/main.js" src="vendor/requirejs/require.js"></script>
34
35 <link rel="stylesheet" href="assets/css/font-awesome.min.css" />
36 <!-- the font-awesome is different from the 'official' one -->
37
38 <!-- application CSS -->
39 </head>
40
41 <body class="skin-3">
42 <div id="wrapper" ui-view="mainContent"></div>
43 </body>
44 </html>
45
```

让某台设备不能上qq

需要在流表中对某个包进行4层匹配，我用OpenDayLight Lithium版本下发4层匹配的流表不能成功，于是准备试试floodlight控制器，用static flow pusher机制试一试。