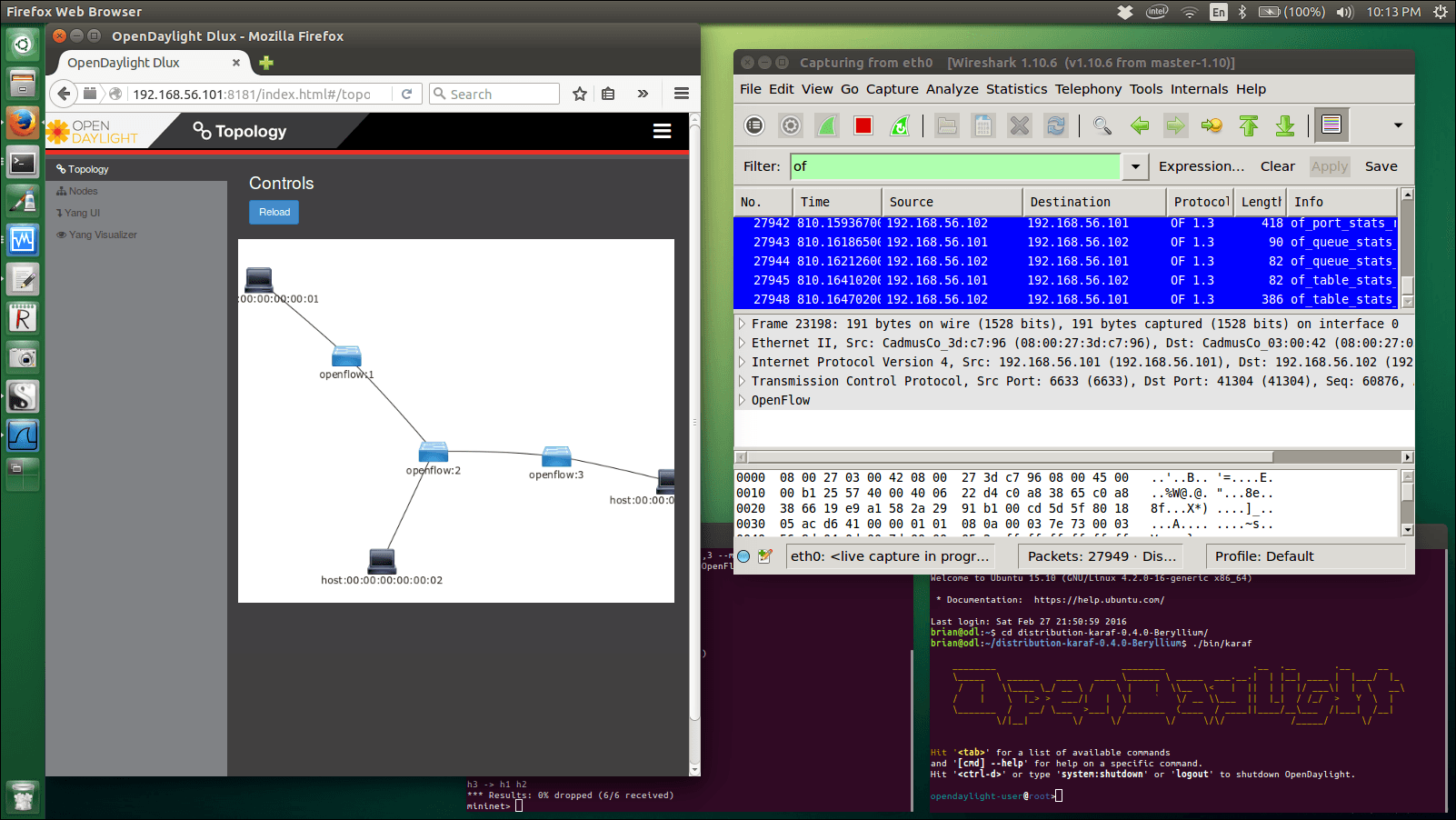
**将OpenDaylight SDN控制器与Mininet网络仿真器一起使用**

2016年2月28日

[OpenDaylight](https://www.opendaylight.org/platform-overview-beryllium)（ODL）是一种流行的开源SDN控制器框架。要了解有关OpenDaylight的更多信息，可以使用它来管理虚拟交换机和虚拟主机的模拟网络。大多数人使用Mininet网络仿真器创建虚拟SDN网络，供OpenDaylight控制。

[](https://i2.wp.com/www.brianlinkletter.com/wp-content/uploads/2016/02/odl-0100-b.png)

在本文中，我将展示如何设置OpenDaylight来使用OpenFlow 1.3来控制仿真的Mininet网络。因为我使用的是虚拟机，所以我使用的过程将在所有常用的主机系统（Linux，Windows和Mac OS X）中相同。

**使用虚拟机**

在这个实验示例中，我将使用两个虚拟机。一个将运行Mininet仿真网络，另一个将运行OpenDaylight控制器。我将两个虚拟机都连接到仅主机的网络，以便它们可以彼此通信，并可以与主机上运行的程序（例如*ssh*和*X11客户端）通信*。

我将[使用VirtualBox运行](http://www.brianlinkletter.com/set-up-mininet/)从[mininet项目网站](https://github.com/mininet/mininet/wiki/Mininet-VM-Images)下载[的Mininet VM](http://www.brianlinkletter.com/set-up-mininet/)，这是尝试Mininet的最简单方法。Mininet项目团队提供了一个Ubuntu 14.04 LTS VM映像，其中已经安装了Mininet 2.2.1，Wireshark和OpenFlow解剖器工具，可以使用。

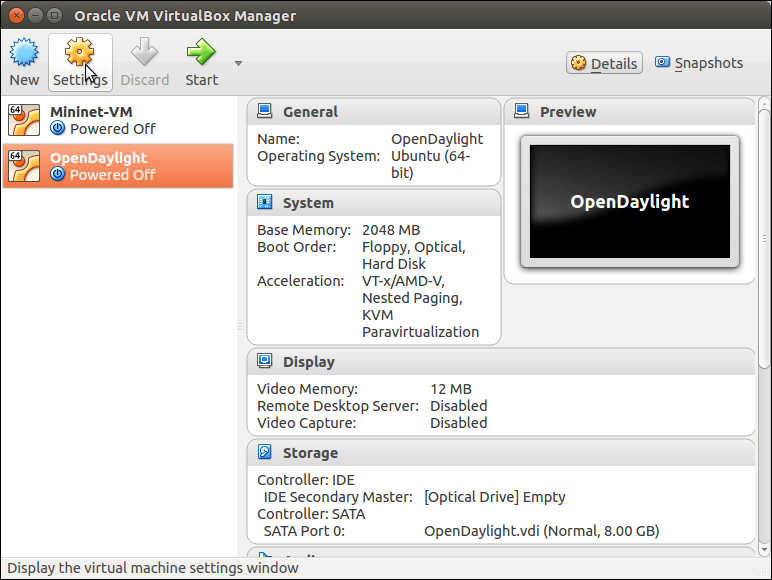
我将在VirtualBox中创建的新VM上安装并运行OpenDaylight SDN控制器。

**Cd b设置OpenDaylight虚拟机**

为了构建OpenDaylight虚拟机，我从ubuntu.com网站[下载了Ubuntu Server ISO映像](http://www.ubuntu.com/download/server)。然后，我将其安装在VirtualBox的新VM中。如果您需要有关如何在VirtualBox虚拟机中安装ISO磁盘映像的说明，请参阅[有关在VirtualBox VM中安装Debian的文章](http://www.brianlinkletter.com/installing-debian-linux-in-a-virtualbox-virtual-machine/)。

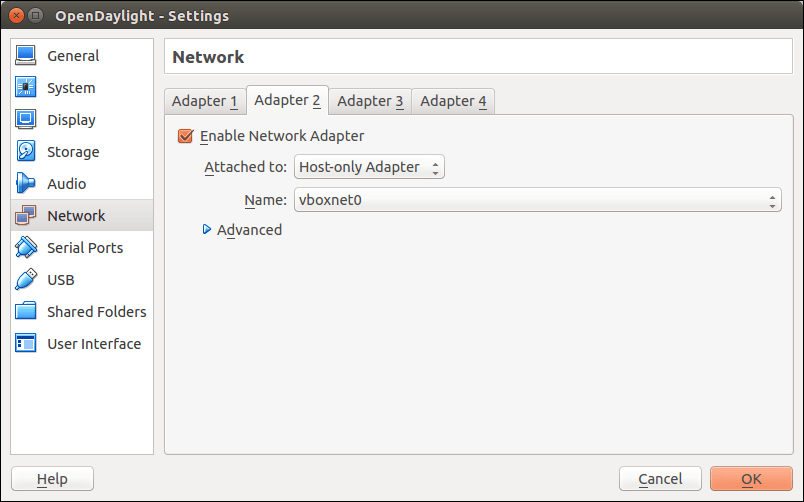
为虚拟机指定一个描述性名称。我将虚拟机命名为*OpenDaylight*。对其进行配置，使其使用两个CPU和2 GB或RAM。这是支持OpenDaylight的最低配置。然后将仅主机的网络适配器添加到VM。

关闭虚拟机电源后，单击“ *设置”*按钮：

[](https://i1.wp.com/www.brianlinkletter.com/wp-content/uploads/2016/02/odl-0008.png)

OpenDaylight虚拟机

在VM的VirtualBox网络设置中，启用两个网络接口。将第一个网络适配器连接到NAT接口（这是默认设置），将第二个网络适配器连接到仅主机的网络*vboxnet0*。

[](https://i2.wp.com/www.brianlinkletter.com/wp-content/uploads/2016/02/odl-0038.png)

将网络适配器2连接到仅主机的网络

**配置OpenDaylight VM接口**

默认情况下，虚拟机的第一个网络适配器连接到VirtualBox NAT接口，并且在虚拟机启动时已经配置。我们需要配置第二个网络适配器，该适配器连接到VirtualBox仅限主机的接口*vboxnet0*。

使用*ip*命令列出所有设备：

brian@odl:~$ ip addr show

1: lo: <LOOPBACK,UP,LOWER\_UP> mtu 65536 qdisc noqueue state UNKNOWN group default

link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00

inet 127.0.0.1/8 scope host lo

valid\_lft forever preferred\_lft forever

inet6 ::1/128 scope host

valid\_lft forever preferred\_lft forever

2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER\_UP> mtu 1500 qdisc pfifo\_fast state UP group default qlen 1000

link/ether 08:00:27:ec:a9:f1 brd ff:ff:ff:ff:ff:ff

inet 10.0.2.15/24 brd 10.0.2.255 scope global enp0s3

valid\_lft forever preferred\_lft forever

inet6 fe80::a00:27ff:feec:a9f1/64 scope link

valid\_lft forever preferred\_lft forever

3: enp0s8: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default qlen 1000

link/ether 08:00:27:b0:f6:70 brd ff:ff:ff:ff:ff:ff

brian@odl:~$

**注意：**从15.10开始，Ubuntu使用[可预测的网络接口](https://www.freedesktop.org/wiki/Software/systemd/PredictableNetworkInterfaceNames/)名称（例如*enp0s3*和*enp0s8）*，而不是经典的接口名称（例如*eth0*和*eth1）*。

我们看到接口*enp0s8*没有IP地址。这是连接到*vboxnet0*的第二个网络适配器。如果DCHP客户端请求，VirtualBox可以使用DHCP在此接口上分配IP地址。因此，运行以下命令来设置接口*enp0s8*：

brian@odl:~$ sudo dhclient enp0s8

现在检查分配给*enp0s8*的IP地址：

brian@odl:~$ ip addr show enp0s8

3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER\_UP> mtu 1500 qdisc pfifo\_fast state UP group default qlen 1000

link/ether 08:00:27:b0:f6:70 brd ff:ff:ff:ff:ff:ff

inet 192.168.56.101/24 brd 192.168.56.255 scope global enp0s8

valid\_lft forever preferred\_lft forever

inet6 fe80::a00:27ff:feb0:f670/64 scope link

valid\_lft forever preferred\_lft forever

brian@odl:~$

现在我们看到连接到仅主机网络的VirtualBox DHCP服务器为此接口分配了IP地址*192.168.56.101*。这是连接到VM上运行的任何应用程序时应使用的IP地址。

在您的系统上，分配的IP地址可能不同。您可能已将VirtualBox首选项设置为对仅主机网络使用不同的网络前缀，或者可能已将DHCP服务器配置为提供不同的地址范围。此外，如果在此VM之前启动了任何其他VM并将其连接到仅主机网络，则分配的IP地址将不同。如果IP地址不同，则可以。只需使用分配的地址即可。

现在，配置接口*enp0s8，*以便在重新启动后仍保持配置。编辑*/ etc / network / interfaces*文件：

brian@odl:~$ sudo nano /etc/network/interfaces

将以下行添加到文件*/ etc / network / interfaces*的末尾：

# the host-only network interface

auto enp0s8

iface enp0s8 inet dhcp

**使用SSH连接到OpenDaylight VM**

在虚拟机上工作时，我喜欢使用终端应用程序。VirtualBox控制台窗口有太多烦人的限制。例如，我不能将文本从主机系统剪切并粘贴到连接到虚拟机的VirtualBox控制台上，反之亦然。

在主机上打开终端并使用SSH登录：

brian@T420:~$ ssh -X brian@192.168.56.101

现在，您已连接到OpenDaylight虚拟机，并且可以看到提示中的主机名已更改为*odl*，该名称是我在VM上安装Ubuntu时配置的。

brian@odl:~$

启动SSH时，我还启用了X转发，因此我可以在OpenDaylight VM上运行X程序，尽管在本教程中我们不会这样做。

**安装Java**

OpenDaylight SDN控制器是一个Java程序，因此请使用以下命令安装Java运行时环境：

$ sudo apt-get update

$ sudo apt-get install default-jre-headless

设置JAVA\_HOME环境变量。编辑*bashrc*文件

brian@odl:~$ nano ~/.bashrc

*将以*下行添加到*bashrc*文件中：

export JAVA\_HOME=/usr/lib/jvm/default-java

然后运行文件：

brian@odl:~$ source ~/.bashrc

**安装OpenDaylight**

从[OpenDaylight网站](https://www.opendaylight.org/)下载OpenDaylight软件。在Linux或Mac OS主机上，我们可以使用*wget*命令下载tar文件。

brian@odl:~$ wget https://nexus.opendaylight.org/content/groups/public/org/opendaylight/integration/distribution-karaf/0.4.0-Beryllium/distribution-karaf-0.4.0-Beryllium.tar.gz

通过解压缩tar文件来安装OpenDaylight：

brian@odl:~$ tar -xvf distribution-karaf-0.4.0-Beryllium.tar.gz

这将创建一个名为*distribution-karaf-0.4.0-Beryllium*的文件夹，其中包含OpenDaylight软件和插件。

OpenDaylight包装在[*karaf*容器中](http://karaf.apache.org/)。Karaf是一种容器技术，允许开发人员将所有必需的软件放在一个[分发文件夹中](https://wiki.opendaylight.org/view/Karaf_Distribution_Folder_and_File_Guide)。由于所有内容都在一个文件夹中，因此可以在需要时轻松安装或重新安装OpenDaylight。稍后我们将看到，karaf还允许程序与可选模块捆绑在一起，这些可选模块可以在需要时安装。

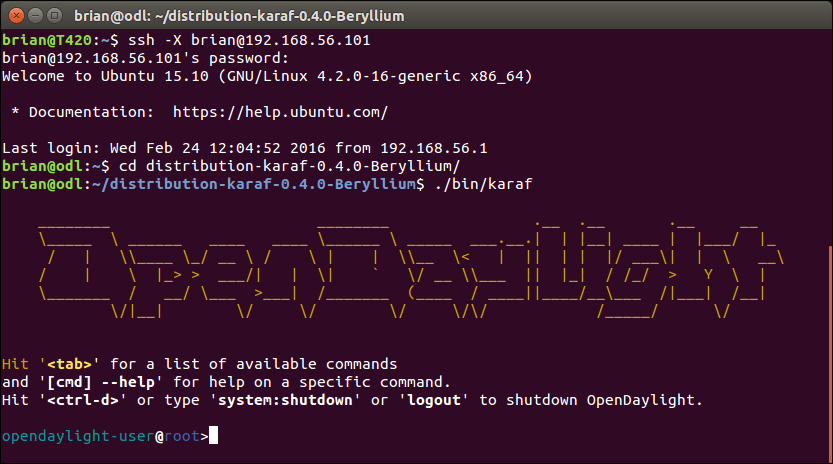
**启动OpenDaylight**

要运行OpenDaylight，请在程序包分发文件夹中运行*karaf*命令。

brian@odl:~$ cd distribution-karaf-0.4.0-Beryllium

brian@odl:~$ ./bin/karaf

现在，OpenDaylight控制器正在运行。

[](https://i2.wp.com/www.brianlinkletter.com/wp-content/uploads/2016/02/odl-0060.png)

在虚拟机中运行的OpenDaylight

**安装OpenDaylight功能**

接下来，安装测试OpenDaylight和OpenDaylight GUI所需的最少功能集：

opendaylight-user@root> feature:install odl-restconf odl-l2switch-switch odl-mdsal-apidocs odl-dlux-all

上面是在karaf容器中安装可选模块的示例。您只需安装一次可选功能。安装后，这些功能将永久添加到控制器中，并在每次启动时运行。

我们安装了以下功能。单击每个功能以了解更多信息：

* [odl-restconf](https://wiki.opendaylight.org/view/OpenDaylight_Controller:MD-SAL:Restconf)：允许访问RESTCONF API
* [odl-l2switch-switch](https://wiki.opendaylight.org/view/OpenDaylight_Controller:MD-SAL:L2_Switch)：提供类似于以太网交换机的网络功能
* [odl-mdsal-apidocs](https://wiki.opendaylight.org/view/OpenDaylight_Controller:MD-SAL:Restconf_API_Explorer)：允许访问Yang API
* [odl-dlux-all](https://wiki.opendaylight.org/view/OpenDaylight_DLUX:DLUX_Karaf_Feature)：OpenDaylight图形用户界面
* 现版本 odl-dlux-all 已经换成 odl-dlux-core

要列出所有可用的可选功能，请运行以下命令：

feature:list | grep odl-dlux

opendaylight-user@root>feature:install odl-l2switch-switch-ui   
opendaylight-user@root>feature:install odl-mdsal-apidocs   
opendaylight-user@root>feature:list | grep odl-dlux

opendaylight-user@root>feature:install odl-dluxapps-yangui   
opendaylight-user@root>feature:install odl-dluxapps-topology

要列出所有已安装的功能，请运行以下命令：

opendaylight-user@root> feature:list --installed

有关OpenDaylight可选功能的信息，请参见OpenDaylight Wiki。

**停止OpenDaylight**

当您要停止控制器时，请输入<ctrl-d>组合键或键入，system:shutdown或logout在*opendaylight-user*提示符下输入。

**设置Mininet虚拟机**

我不会在本文中介绍设置Mininet VM所需的所有步骤，因为我已经在另一篇文章“ [设置Mininet VM”中](http://www.brianlinkletter.com/set-up-mininet/)介绍了该主题。

在VirtualBox管理器中启动Mininet VM。现在，我们应该有两个正在运行的VM：OpenDaylight VM和Mininet VM。如果我们首先启动OpenDaylight VM，它将具有IP地址192.168.56.101，并且mininet VM将在仅主机网络上获得第二个可用IP地址192,168.56.102。我们可以通过在Mininet VM控制台上运行*ip*命令来验证这一点：

mininet@mininet-vm:~$ ip addr show

1: lo: <LOOPBACK,UP,LOWER\_UP> mtu 65536 qdisc noqueue state UNKNOWN group default

link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00

inet 127.0.0.1/8 scope host lo

valid\_lft forever preferred\_lft forever

2: eth0: <BROADCAST,MULTICAST,UP,LOWER\_UP> mtu 1500 qdisc pfifo\_fast state UP group default qlen 1000

link/ether 08:00:27:e2:98:cc brd ff:ff:ff:ff:ff:ff

inet 192.168.56.102/24 brd 192.168.56.255 scope global eth0

valid\_lft forever preferred\_lft forever

3: eth1: <BROADCAST,MULTICAST,UP,LOWER\_UP> mtu 1500 qdisc pfifo\_fast state UP group default qlen 1000

link/ether 08:00:27:1b:c1:07 brd ff:ff:ff:ff:ff:ff

inet 10.0.2.16/24 brd 10.0.2.255 scope global enp0s3

valid\_lft forever preferred\_lft forever

mininet@mininet-vm:~$

**注意：** Mininet VM基于Ubuntu Server 14.04，它尚未使用[可预测的网络接口](https://www.freedesktop.org/wiki/Software/systemd/PredictableNetworkInterfaceNames/)名称，如*enp0s3*和*enp0s8*，因此我们看到的接口名称如*eth0*和*eth1*。

我们看到*eth0*连接到仅主机接口，因为它的IP地址为192.168.56.102，该地址在VirtualBox仅热网络DHCP服务器分配的地址范围内。因此，我们知道我们需要使用IP地址192.168.56.102来访问在此虚拟机上运行的应用程序。

**使用SSH连接到Mininet VM**

现在，在主机上打开一个终端窗口，然后通过SSH进入Mininet VM。开启X转发。（如果使用的是Windows，则将[Xming](http://www.straightrunning.com/XmingNotes/)用于X Window系统服务器，并将[Putty](http://www.chiark.greenend.org.uk/~sgtatham/putty/)作为SSH客户端）

brian@T420:~$ ssh -X 198.168.56.102

**启动Mininet**

在Mininet VM上，启动简单的网络拓扑。在这种情况下，我们将执行以下操作：

* 在线性拓扑中设置三个交换机
* 每台交换机将连接到一台主机
* 每个主机上的MAC地址将设置为一个简单数字
* 遥控器OpenDaylight的IP地址为192.168.56.101:6633
* 我们将使用OpenFlow版本1.3

用于启动此操作的Mininet命令为：

mininet@mininet-vm:~$ sudo mn --topo linear,3 --mac --controller=remote,ip=192.168.56.101,port=6633 --switch ovs,protocols=OpenFlow13

**测试网络**

通过ping所有节点来测试OpenDaylight控制器是否正常工作。每个主机都应该能够访问其他每个主机：

mininet> pingall

\*\*\* Ping: testing ping reachability

h1 -> h2 h3

h2 -> h1 h3

h3 -> h1 h2

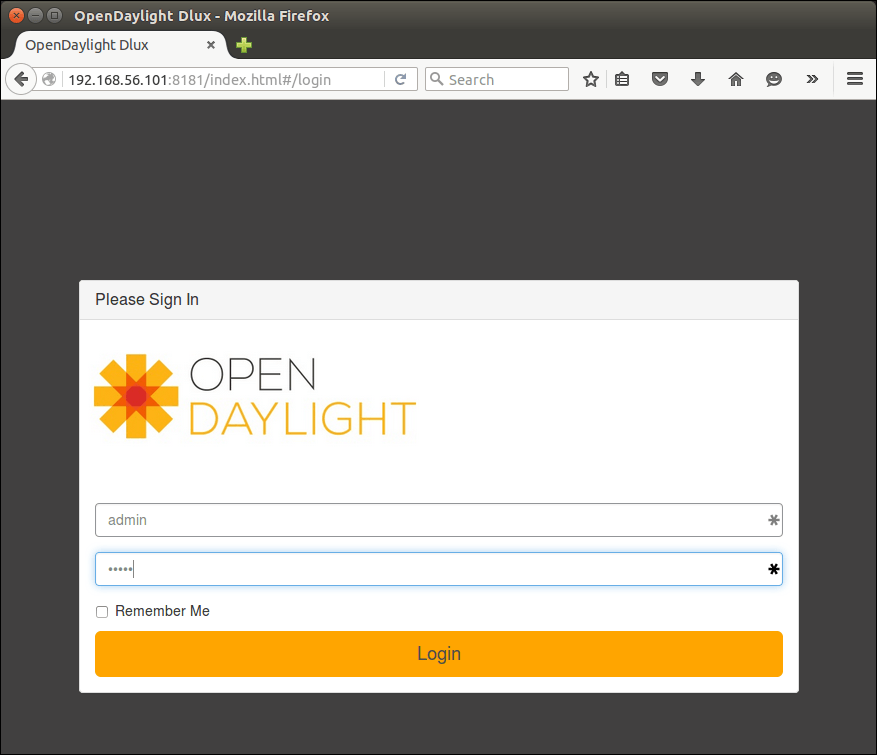
\*\*\* Results: 0% dropped (6/6 received)

**OpenDaylight图形用户界面**

在主机系统上打开浏览器，然后输入OpenDaylight用户界面（DLUX UI）的URL。它在OpenDaylight VM上运行，因此IP地址为192.168.56.102，应用程序定义的端口为8181：

因此，网址为：http://192.168.56.101:8181/index.html。

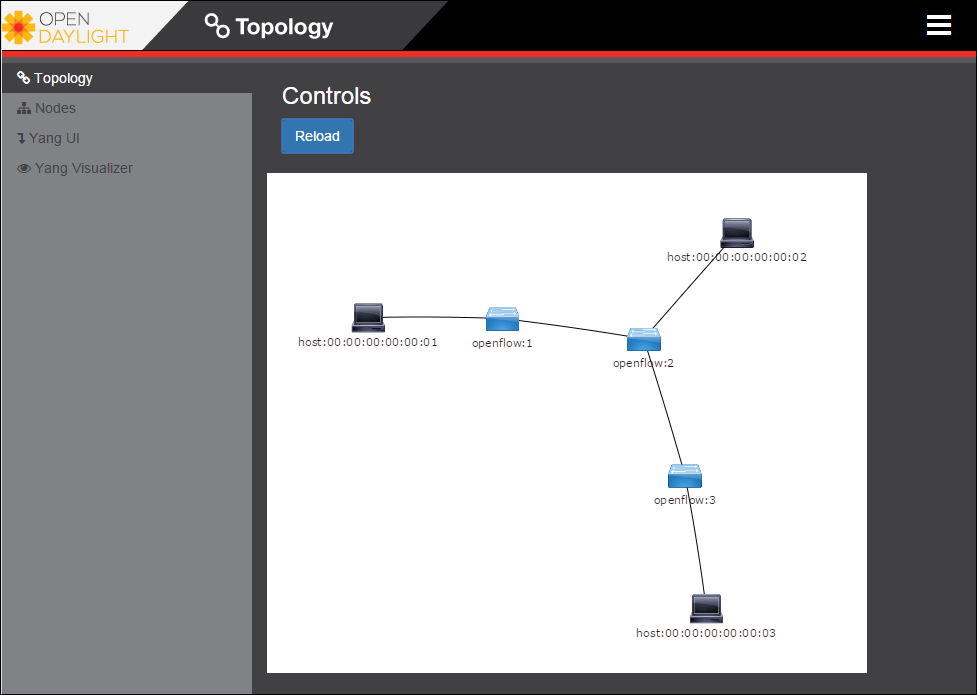
默认的用户名和密码均为*admin*。

[](https://i0.wp.com/www.brianlinkletter.com/wp-content/uploads/2016/02/odl-0061.png)

登录到OpenDaylight控制器

**拓扑结构**

现在，我们在OpenDaylight控制器的“ *拓扑”*选项卡中看到了网络拓扑。

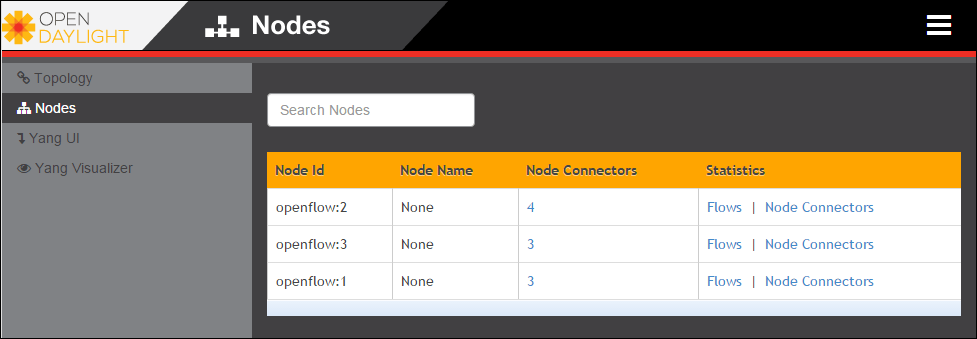
[](https://i0.wp.com/www.brianlinkletter.com/wp-content/uploads/2016/02/b-odl-0062.png)

Mininet网络的拓扑

您可以看到由Mininet网络仿真器仿真的网络。您可以通过在Mininet中使用不同的属性构建不同的网络拓扑，以及使用OpenDaylight在仿真网络上运行实验来测试OpenDaylight功能。例如，您可以断开Mininet中交换机之间的链接，以测试网络如何响应故障。

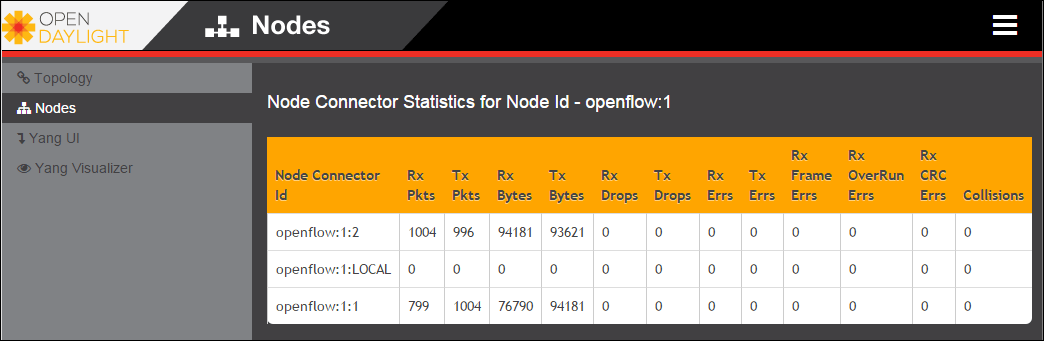
**节点数**

单击“ *节点”*选项卡以查看有关网络中每个交换机的信息：

[](https://i1.wp.com/www.brianlinkletter.com/wp-content/uploads/2016/02/b-odl-0063.png)

节点清单

单击每一行中的“ *节点连接器”*链接，以查看有关交换机上每个端口的信息：

[](https://i2.wp.com/www.brianlinkletter.com/wp-content/uploads/2016/02/b-odl-0064.png)

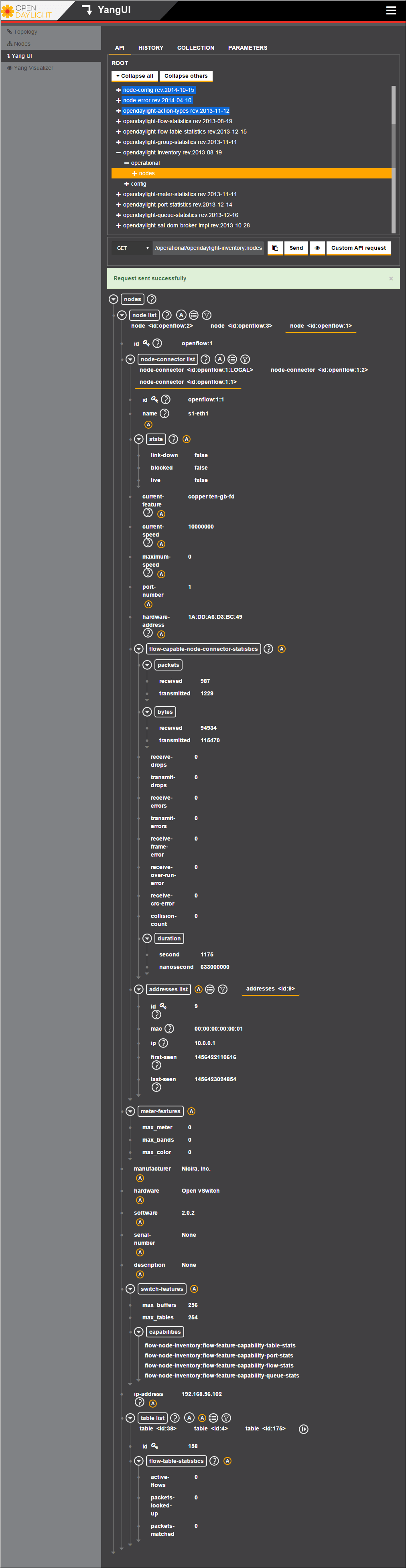
介面

**杨UI**

[Yang](http://www.yang-central.org/twiki/bin/view/Main/WebHome)是一种数据建模结构。使用硬件路由器和交换机的工程师将熟悉基于SNMP，SMI和MIB的另一种数据建模结构。Yang在SDN交换机中提供的功能类似于非SDN交换机的SMI。

OpenDaylight Yang UI是一个图形[REST](http://rest.elkstein.org/)客户端，用于构建REST请求并将其发送到OpenDaylight数据存储。我们可以使用Yang UI从数据存储中获取信息，或构建REST命令来修改数据存储中的信息-更改网络配置。

单击“ *Yang UI”*选项卡。然后单击*全部展开*按钮以查看所有可用的API。并非所有功能都可以使用，因为我们没有安装所有功能。一种有效的API是*清单* API。单击它，然后向下导航到*nodes*属性，然后单击*Send*按钮将*GET* API方法发送到控制器。

[](https://i2.wp.com/www.brianlinkletter.com/wp-content/uploads/2016/02/b-odl-0065.png)

网络杨数据模型

向下滚动以查看有关网络的所有清单信息：节点，端口，统计信息等。单击交换机和接口以查看每个的详细信息。

了解Yang数据模型并学习如何读写数据存储是了解OpenDaylight控制器的软件定义网络的关键。

**捕获OpenFlow消息**

为了更深入地研究SDN控制器和交换机的工作方式，您可能希望查看网络中控制器和交换机之间交换的OpenFlow消息。

Mininet VM 已安装[Wireshark](https://www.wireshark.org/)，并已设置[OpenFlow解剖器](https://wiki.wireshark.org/OpenFlow)的自定义版本。

因此，查看OpenFlow消息的最简单方法是在Mininet VM上启动Wireshark，并在连接到仅主机网络的接口上捕获数据，本例中为*eth0*。

打开一个新的终端窗口，并使用启用了X转发的SSH使用SSH连接到Mininet VM（如果使用Windows，则使用Putty和Xming）：

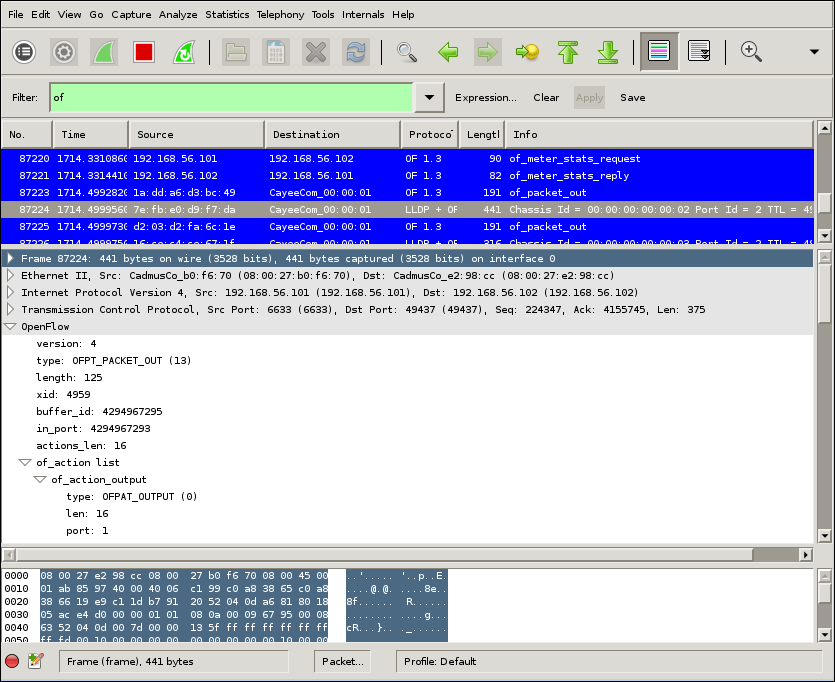
brian@T420:~$ ssh -X 192.168.56.102

在Mininet VM上启动Wireshark：

mininet@mininet-vm:~$ sudo wireshark &

您将看到一个警告对话框，但可以忽略它。使用root特权启动Wireshark会带来安全风险，但是对于我们的简单测试，我们可以忽略这一点，或者您可以按照警告消息中的说明以更安全的方式设置Wireshark。

为OpenFlow消息创建显示过滤器。of在“过滤器”窗口中输入文本，然后单击“ *应用”*。现在，您将在Wireshark显示屏中仅看到OpenFlow消息，如下所示。

[](https://i0.wp.com/www.brianlinkletter.com/wp-content/uploads/2016/02/b-odl-0066.png)

在Wireshark中查看捕获的OpenFlow消息

**关闭项目**

当需要结束项目时，请使用以下命令关闭Mininet和OpenDaylight：

在Mininet VM上，停止Mininet并清理节点，然后关闭VM：

mininet> exit

mininet@mininet:~$ sudo mn -c

mininet@mininet:~$ sudo shutdown -h now

在OpenDaylight VM上，停止OpenDaylight并关闭VM：

opendaylight-user@root> system:shutdown

brian@odl:~$ sudo shutdown -h now

现在，两个VM都应显示它们已在VirtualBox Manager应用程序中停止。

**结论**

我们展示了如何在虚拟机中安装OpenDaylight并将其连接到在另一台虚拟机上运行的Mininet网络仿真器。我们演示了OpenDaylight的一些功能，并展示了如何捕获控制器和仿真交换机之间交换的OpenFlow消息。