

# 简述软件定义网络 Mininet 仿真的使用方法

王鹏宇

四川大学锦城学院 成都 610000

【摘要】随着云计算，物联网和 5G 网络的快速发展，SDN 新型网络架构被提出。这样的新型网络架构解决了通过预设好的网络对应需求，将传统的对应转变成了通过需求来定义网络。首先介绍了 Mininet 是作为 SDN 网络搭建前的测试工具，其次从搭建方式出发，分别简述了从 Mininet 可视化工具 Miniedit，Mininet 交互式界面和编写 Python 脚本三种搭建网络拓扑的方式作为 SDN 网络搭建前模拟的使用方法。最后对未来的发展提出了个人的看法。

【关键词】软件定义网络；Mininet 仿真；使用方法

由于云计算，物联网和 5G 网络的飞快发展需要多类型、多种别的网络表层应用和业务以及较为稳定的网络架构设计和系统的运行维护。因为不同的网络需求需要不同的网络设备。传统的网络是基于需求预先设定好的它不会因为你需要不同的网络就马上配置好你所需要的，传统网络从开发出来就已经决定了它的用途。经过了现在的网络快速发展就需要一种一种新型的网络架构，它应该具有即时性、自定义性能够通过实际的需求来客制化。

软件定义网络 SDN 全称 Software Defined Network，这样的新型网络架构解决了通过预设好的网络对应需求，将传统的对应转变成了通过需求来定义网络。SDN 可以通过软件来定义所需要的网络，这样用户能通过软件编程来控制网络同时快速的满足客户的需求。将从设备供应商的设备可编程转变为用户的网络可编程。SDN 主要实现的是控制和转发的分离，通过编程软件来沟通控制器（controller），SDN 可对整个网络编程，无论从任何一个节点接入，转发层面完全不感知，这也是控转分离的好处。

现阶段 SDN 的主要架构有两种。一种是由 ONF（开放网络基金会）定义的架构，这个 SDN 架构分为三层：拥有各种的应用和业务的应用层、负责数据处理资源调配的控制层和负责转运、处理、收集的基础设施层。控制层与应用层之间的接口称为北向接口，控制层与基础设施层之间的接口称为南向接口。在南向接口之间主要应用的是 Open Flow 协议。它将传统的网络设备推翻来构建 SDN。另一种是由 IETF 定义的 SDN 架构。IETF 定义的 SDN 架构分为应用层、网络层和基础设施层。IETF 的 SDN 架构主张在现有的网络层协议上添设插件，并在网络与应用层之间增加 SDN 适配器进行能力封装。IETF 统一接口让所有厂商的 SDN 插件都能在现有的各种网络设备上连接使得现有的网络设备不需要马上淘汰，在 SDN 的发展上被革新掉。IETF 定义的 SDN 架构的特点是主要就是保留当前的网络设备通过加入插件来达到软件定义网络的目的，它充分利用现有的设备。

## 1 Mininet 相关介绍

Mininet 是由虚拟的交换机、路由器终端节点（end-hosts）连接而成的一个网络仿真器。Mininet 是一款开放源代码的软件，是一个基于 LXC 工具虚拟化技术的轻量级网络模拟器。因为它采用轻量级的虚拟化技术，所以使得系统能够和现实的网络情况有非常高的一致性。

### 1.1 Mininet 与 SDN

SDN 即软件定义网络，是现在网络中一种新型网络创新架构，软件定义网络的核心技术是通过 Open Flow 协议使网络设备控制面与数据转发面分离开来，从而达到巧妙控制网络之间的流量。而 Mininet 作为一个轻量化软件定义网络的测试验证工具；它利用轻量级的虚拟化技术在系统中搭建一个完整的网络拓扑结构并且运行。它支持 OpenFlow、Openv Swith 等各类型的协议，Mininet 同时能够虚拟搭建一个完整的网络主机、控制器和交换机在同一台计算机上，这样就有助于 SDN 开发、测试和演示，再将在 Mininet 中测试成功的虚拟化的平台下的主机、交换机控制器的搭建和代码转移到现实的网络环境中进行实际运用。

### 1.2 Mininet 的作用

Mininet 能够很轻松地简历一个支持 SDN 的网络拓扑：主机（host）就像现实的主机一样真正的工作，也能够通过 ssh 远程登录，打开应用程序，程序能够对以太网端口发送包，包会被交换机、路由器接收并处理。拥有这个网络，就能够十分灵便地为网络添加新的功能并进行相关测试，然后轻便布置到现实的硬件情况中。它能够为 Open Flow 供给应用测试平台，用于网络调试和运行测试命令行界面，支持拓扑搭建和 Open Flow 的有关命令，同时提供 Python API 接口，用于编程拓展功能。

### 1.3 Mininet 目录介绍

在 mininet/bin 目录下面有且只有一个 mn.1 的文件，这个文件是 mininet 的主运转文件。在 mininet/util 目录下的所有文件当中一个名为 install.sh 的文件是在安装过程中会使用

到的, 该文件是 Linux 下运行的脚本文件用于 mininet 的安装。在 mininet/custom 目录下有两个文件, 一个是 README 文件另一个是以 “.py” 结尾的 Python 脚本文件这个 Python 脚本文件定义了一个网络拓扑。一般而言自定义的网络拓扑都会保存为 Python 脚本文件在 mininet/custom 目录下。在 mininet/examples 目录下有很多 Python 脚本文件, 这些文件就是实现 mininet 功能的脚本文件。

## 2 Mininet 仿真的使用方法

Mininet 仿真搭建计算机网络拓扑结构的使用方法大致有三种。一是运用 Mininet 可视化工具 Miniedit 搭建计算机网络拓扑结构, 二是运用 Mininet UI 的搭建计算机网络拓扑结构, 三是通过先写 Python 脚本再启用该脚本来构建计算机网络的拓扑结构。

### 2.1 可视化工具 MiniEdit

首先进入装好 mininet 的 Ubuntu Linux 操作系统。

#### 2.1.1 Terminal Emulator 里操作命令

打开 Terminal Emulator, 输入 -ls 命令可以查看当前目录下的所有文件。我们需要进入到 mininet/ 目录下所以输入命令 cd mininet, 而后一样能够输入 -ls 查看当前目录 (mininet/) 下的文件, 这个时候我们需要进入到 examples/ 目录下所以输入命令 cd examples。同样在打开 Terminal Emulator 后也能够输入 cd mininet/examples 直接进 mininet/examples 目录下。在此目录下输入 -ls 能够看到有很多 Python 脚本文件, 这些文件都是实现 mininet 功能的脚本文件, 此中的 miniedit.py 文件就是 mininet 的可视化工具 miniedit。此时输入 sudo python miniedit.py 去运行这个 Python 脚本, 这样就可以进入 miniedit 可视化工具。

#### 2.1.2 MiniEdit 界面

在 MiniEdit 界面中都是图像化的。在左边栏里分别有主机 (Host)、交换机 (Switch)、传统交换机 (LegacySwitch)、传统路由器 (LegacyRouter)、网络连接线 (NetLink) 和控制器 (Controller)。上方栏 File 选项可以选择新建 (New)、打开 (Open)、保存为扩展名是 mn 脚本文件 (Save) 和保存为 Python 脚本 (export level 2 scrip), Edit 选项中有剪切 (Cut) 和参数选择 (preferences) 参数选择里可以设置 IP、打开 CLI、选择 Open vSwitch 的 OpenFlow 版本。Run 选项运行搭建的拓扑结构。Help 选项查看关于 miniedit。在 miniedit 里搭建拓扑的时候, 在 Terminal Emulator 里也在执行对应的命令。在搭建网络拓扑的过程中如果有发现有添加的设备多余了, 可以左键选中多余的设备在搭建好拓扑后可以右键设备进行进一步的设备配置。

### 2.2 Mininet 交互式界面

打开 Terminal Emulator, 输入 sudo mn 能够建立一个简单的一个控制器、一个 OpenFlow 交换机、两个主机的网络拓扑结构。

#### 2.2.1 网络构建参数 --topo

(1) 创建单一拓扑 (一台交换机, n 台主机) 打开 Terminal Emulator, 输入 sudo mn --topo=single,n。

在这条命令中 n 是主机的数目

(2) 创建线性拓扑 (n 台交换机呈线性形式分布并且每台交换机下有一台主机) 打开 Terminal Emulator, 输入 sudo mn --topo=linear,n。

在这条命令中 n 代表的是线性分布中交换机的数目。

创建树形拓扑 (交换机呈树形连接, 每台交换机多台主机)

打开 Terminal Emulator, 输入 sudo mn --topo=tree,depth=n,fanout=m。

在这条命令中 depth 代表的是交换机的深度, 所以是 n 的数值决定了交换机的层数。

fanout 代表的是扇出也就是说最顶端的交换机下有几台交换机和下属交换机中又有几台交换机或者是主机, m 就决定了扇出的个数。

创建自定义拓扑

打开 Terminal Emulator, 输入 sudo mn --custom file.py --topo mytopo。

这条命令执行的是 mininet/custom 目录下的 Python 脚本文件, file 是用户所保存的 Python 脚本的文件名。

#### 2.2.2 网络构建参数 -- switch

定义 mininet 需要使用的交换机, 默认为 openvswitch 交换机。

- (1) ovsk (默认选项);
- (2) lxb (内核态);
- (3) user (用户态);
- (4) ivs (ovsk)。

#### 2.2.3 网络构建参数 --controller

定义 mininet 需要利用的控制器, 如果没有定义就会使用 mininet 本身的控制器。

指定远程控制器命令: sudo mn --controller=remote,--ip=[控制器的 ip],--port=[port]。

#### 2.2.4 网络构建参数 -- mac

自动设置 MAC 地址, 能让 MAC 能够从小到大的顺序设置。

#### 2.2.5 交互界面执行 Python 表达式构建网络

(1) 在原有拓扑的中增加主机: 输入 py net.addHost ( '所需要增添主机的名字' )

(2) 在原有拓扑的中增加交换机: 输入 py net.addSwitch ( '所需要增添交换机的名字' )

(3) 在原有拓扑的中连接交换机和主机: 输入 py net.addLink ( 交换机名 ,net.get ( '主机名' ) )

(4) 在原有拓扑的中连接交换机的以太网端口: 输入 py s1.attach ( 's1-eth3' ),s1 指的是任意交换机的名字, eth3 是

所需连接的以太端口号。

#### 2.2.6 Python 脚本

在 Terminal 中，输入 `sudo vi file.py` (file 可以是任意所需创建的文件的文件名) 进入 vi 编辑器并创建新文件。在 vi 编辑器中可以通过编写 Python 脚本来实现网络拓扑的构建。编写完后点击 ESC 键输入 `wq!` 强制保存退出。在需要使用拓扑的时候在 Python 脚本保存的目录下输入 `sudo python file.py` (file 指的是所需要打开的 Python 脚本的文件名)。这样拓扑就会搭建好了。

#### 2.2.7 内部交互命令

不论是在 Mininet 可视化工具 Miniedit 创建计算机网络拓扑结构，还是运用 Mininet UI 创建计算机网络拓扑结构，或者是通过编写 Python 脚本来构建计算机网络的拓扑结构。这些拓扑的搭建都是可以在 Terminal Emulator 中进行检测、查看的。

#### 2.2.8 检测

经过 `iperf` 可以对两个节点 (主机) 之间的进行带宽检测 (如: `iperf h3 h4`)，经过 `iperfudp` 检测两节点之间的 udp 带宽 (如: `iperfudp bw h3 h4`)，通过 `link` 禁用或者打开节点之间的链路 (如: `link s3 s4 up`)，通过 `dpctl` 可以在所有交换机上增添删除修改流表 (如: `dpctl dump-flows`)，通过 `xterm` 可以进入终端可视化操作 (如: `xterm h3`)。查看通过 `pingall` 可以对网络中的所有主机主机之间执行 ping 测试 (指定主机 ping 测试可以使用如: `h3 ping h4`)。通过 `ifconfig` 可

以查看 IP 等信息。

#### 2.2.9 查看

经过 `help` 命令能够查询帮助列表。经过 `nodes` 命令能够查看 mininet 中各个节点的状态。通过 `net` 命令可以显示网络拓扑。通过 `dump` 命令可以体现出每个节点的接口设置和每个节点的进程的 PID。通过输入 `exit` 退出 mininet 登录。

### 3 未来发展

SDN 在将来会与 5G 网络，云计算，大数据，物联网和人工智能等当下热门的技术发展有着不可分割的关系。SDN 是关键技术，解耦后实现未来可拓展，使网络架构具有灵活性和扩展性，降低业务对设备的依赖实现自主可控。而以 Mininet 作为模拟工具将会大大减少在真实网络环境中所会遇到的问题，使得搭建真实的网络更加顺利。

### 4 结束语

随着云计算，大数据，物联网和人工智能等新兴技术的飞速发展，传统的网络模式已经稍显得不能满足当下的客户业务需求了，而 SDN 的提出便给出了解决的办法。本文从 Mininet 可视化工具 Miniedit，Mininet UI，编写 Python 脚本三种搭建网络拓扑的方式来简述了 Mininet 作为 SDN 网络搭建前模拟的使用方法，最后对 SDN 未来的发展提出了个人的看法。

### 【参考文献】

- [1] 朱金奇, 孙华志, 黄永鑫, 等. 软件定义网络中延迟满足的路由选择与实时调度更新 [J]. 软件学报, 2019, 30 (11): 3440-3456.
- [2] 罗定福, 李艳. 三层交换机与路由器连接的技术研究 [J]. 佛山科学技术学院学报 (自然科学版), 2013, 31 (2): 52-54.