mininet / **mininet**

FAQ

Subject22 edited this page on Nov 1 2017 · 333 revisions

Please add helpful **Frequently Asked Questions (FAQs)** and **high-quality answers (HQAs)** below.

Before you send a question to `mininet-discuss` , make sure your question isn't already in the FAQ -- and if you see a question asked repeatedly, feel free to add the answer to this FAQ!

Since github wiki markdown doesn't automatically generate a Table of Contents, please follow the existing format and add 1) an anchor for your question and 2) a link to it at the top of the page. Thanks!

Start Here

- How can I **get started** with Mininet?

VM and Installation Questions

- What is the **login/password** for the Mininet VM?
- Why can't I get **X11 forwarding** to work? I get `cannot open display:` or `$DISPLAY not set` , and `wireshark` doesn't work! `xterm` doesn't work either!
- X11 forwarding is too hard! Can't I just **run a GUI in my VM** console window?
- Can I run a **GUI/X11 application** within a Mininet host?
- How can I do a **install Mininet natively** on my Linux machine?
- How can I **uninstall Mininet**?
- Help! I **can't boot my VM in VirtualBox in Windows**! Do I need a 32-bit VM?
- Help! The **VM console screen is blank**!
- Help! I **can't import the `.ovf`** into VirtualBox/VMware/etc.!
- Help! I **can't connect to the internet** on the Mininet VM to install packages! (or, I can't SSH into the mininet VM from the host box)
- Help! Mininet is not working in an `lxc` container because of AppArmor!

Using Mininet

- How do I **run Linux programs** on my Mininet hosts?
- How do I figure out the **command-line options** for the `mn` command?
- How do I **use Mininet's Python API**?
- Why can't I **ping Google** from my Mininet hosts?/How can I set up **NAT**?
- Why do the **switch data ports** have **random MAC addresses**? How do I assign MAC and IP addresses to the switch data ports?
- Why does my network fail if I use **more than 16 switches**?
- How can I **control Mininet hosts remotely**?
- How can I add a **REST interface** to Mininet?
- How do I run the **Mininet examples**?
- Why doesn't `dpctl` work? How can I **dump a switch's flow table**?
- How do I **generate traffic** on my Mininet network?
- How do I **modify packet headers** using Mininet?
- How do I implement a **custom routing algorithm**?
- How do I **update to a new version of Mininet**?
- In addition to the `mininet-discuss` mailing list, is there a **`#mininet` IRC channel**?
- Can I turn on SSL for Open vSwitch?
- Why doesn't pmonitor display any output for some Python commands?

OpenFlow Questions

- Which **versions of OpenFlow** does Mininet support?
- What **OpenFlow switch implementations** does Mininet support?

▸ **Pages** 67

- Mininet
- Get Started
- Sample Workflow
- Walkthrough
- Overview

- Download
- Documentation
- Videos
- Source Code
- Apps and Tools
- FAQ
- Wiki
- Teaching
- Papers

- Support
- Contribute
- News
- Credits

Clone this wiki locally

`https://github.com/mininet/`

Clone in Desktop

- How can I use **OpenFlow 1.3 only**?
- How can I use **multiple controllers** in my network?
- Can I **upgrade Open vSwitch** to a newer version?

Troubleshooting

- Help! I **updated my Ubuntu kernel** and now **Open vSwitch won't start**!
- Why does my controller, which implements an **Ethernet bridge or learning switch**, not work with my network which has **loops** in it? I can't ping anything!
- Help! Mininet is hanging on startup!

Advice for the Desperate and/or Clueless!

- Help! **I don't understand OpenFlow** or SDN!
- Help! **I have never used Linux or Unix before!**
- Help! **I don't understand networking** (or maybe computers) at all!!
- How do I **use (some Linux command)**?
- Will you **do my (home)work assignment** for me? It's due next Tuesday at 4pm!

How can I get started with Mininet?

Quick answer: Follow the steps on our Documentation page!

The best way to get started with Mininet is to install our ready-to-run virtual machine image as per our Download instructions, then go through the Walkthrough, and then continue with the other steps on our Documentation page. You may find the Introduction to Mininet and the OpenFlow tutorial to be particularly useful in helping you to understand and use Mininet and OpenFlow/Software-Defined Networking, respectively.

What is the login/password for the Mininet VM?

As of Mininet 2.0.0 and newer, it is currently:

```
mininet-vm login: mininet
Password: mininet
```

On some older VMs it was `openflow` / `openflow` .

Why can't I get X11 forwarding to work? I get `cannot open display:` or `$DISPLAY not set` , and `wireshark` doesn't work! `xterm` doesn't work either!

This is not a Mininet problem. It means that **X11 forwarding is not set up correctly**. First, consult the X11 setup instructions in the OpenFlow Tutorial, including:

- Download X11
- Install X11
- Start up X11 server
- Access VM via `ssh`

Make sure you have carefully followed the necessary steps. If things are still not working for you, you will want to make sure that:

1. Your **X11 server** on your client machine (e.g. your laptop) is installed correctly and is actually running
2. You are connecting via `ssh` using **X11 forwarding** (e.g. `ssh -X` on OS X/Linux or enabling X11 forwarding in a Windows `ssh` client like PuTTY or SecureCRT.)
3. You don't have any options in your client `.ssh/config` which interfere with X11 forwarding

When you log in with `ssh` , your `$DISPLAY` environment variable is set X11 terminology is a bit confusing because the X11 server is actually run on the ssh client machine! The ssh client connects to the sshd server, which in turn forwards connections from X11 client applications (such as wireshark) to the local X11 server (usually running on your laptop or whatever machine is sitting in front of you.)

One note: if you have disabled IPv6, you may find that you need to add `AddressFamily inet` to your `/etc/ssh/sshd_config` .

Other unlikely causes of breaking X11 forwarding include it being disabled in `/etc/sshd_config` or disabled by SELinux. Neither of these should be the case in the Mininet VM image we provide. You may wish to invoke debug logging on your `ssh` client to see why X11 forwarding isn't working. On OS X and Linux, you can use

```
ssh -X -v <VM's IP address>
```

to see where the X11 forwarding is failing.

By default `ssh -X` times out after a while - you may prefer `ssh -Y` for that reason, although it is less secure.

There is **a wealth of information on the internet** explaining how to set up X11 forwarding correctly on any platform. This is easily found using Google or the search engine of your choice.

As an alternative to X11, you could also use VNC, but that is probably about as complicated as X11 and is left as an exercise to the reader.

If this seems too complicated, you can simply run X11 in the VM console window as described below!

X11 forwarding is too hard! Can't I just run a GUI in my VM console window?

Yes, you can!

First, log in to the VM in its console window (i.e. type directly into the VM window without using `ssh`) and make sure `apt` is up to date:

```
sudo apt-get update
```

Then, install the desktop environment of your choice.

```
sudo apt-get install xinit <environment>
```

where `<environment>` is your GUI environment of choice. Some options:

- `lxde` : a reasonably compact and and fast desktop environment
- `flwm` : a smaller but more primitive desktop environment
- `ubuntu-desktop` : the full, heavyweight Ubuntu Unity desktop environment

Then, you can start X11 in the VM console window using

```
startx
```

If you are running VirtualBox, you will want to install the VirtualBox Guest Additions using

```
sudo apt-get install virtualbox-guest-dkms
```

Reboot the VM, log in and run `startx` , and you should be able to resize the VM console window and desktop.

Can I run a GUI/X11 application within a Mininet host?

Yes, you can do so from a host `xterm` with the current version of Mininet.

This allows you to run programs like `wireshark` or `firefox`. You might want to `su` to another user (e.g. `mininet`) to avoid running FireFox with `root` privileges, but it probably doesn't make a fundamental difference in a Mininet VM that is configured for password-less `sudo`.

The CLI `xterm` command actually sets up an X11 tunnel which you can continue to use, e.g.

```
mininet> xterm h1
mininet> h1 wireshark &
```

You can also use the `x` command to set up the X11 tunnel (and optionally run an X program):

```
mininet> x h1 xclock &
mininet> x h2
mininet> h2 wireshark &
```

What about Mininet 2.0.0?

(Thanks to Murphy McCauley for providing the following workaround, which craftily uses the switch's CPU port!)

This is a bit of a hack at this moment but it works!

Using a graphical browser requires that you get X11 traffic out of your Mininet host namespace and into the environment where you actually have an X display.

Say the X display is the host environment with IP address of 192.168.56.1 and the Mininet VM has an IP of 192.168.56.101.

In short, run sshd inside Mininet's h1. Then SSH from the host environment (192.168.56.1) to the Mininet VM (192.168.56.101) with X forwarding (ssh -Y mininet@192.168.56.101). Then SSH from the Mininet VM into h1 with X forwarding (again!).

As an example, open three terminals in the host environment (Term1, Term2, Term3)

On Term1:

```
./pox.py forwarding.l2_learning # Run an OpenFlow controller
```

On Term2:

```
ssh -Y mininet@192.168.56.101 # SSH into the Mininet VM with X forwarding
sudo mn --topo=linear,2 --mac --controller=remote,ip=192.168.56.1:6633
h1 /usr/sbin/sshd # From the mininet> prompt, run sshd inside the h1 namespace
```

On Term3:

```
ssh -Y mininet@192.168.56.101 # SSH into the Mininet VM with X forwarding
sudo ifconfig s1 10.12.12.12 # Give the internal adapter for s1 an address #By default,
all hosts live on 10.0.0.0/8 space.
ssh -Y mininet@10.0.0.1 # SSH into the Mininet h1 namespace with X forwarding
xeyes # Run any X app
```

How do I run Linux programs on my Mininet hosts?

If you are asking this question, it means you haven't yet consulted the Documentation. :(

If you are incredibly lazy, please at least look at the Sample Workflow.

How do I figure out the command-line options for the `mn` command?

```
mn --help
```

How can I install Mininet natively on my Linux machine?

Instructions for native installation can be found at http://mininet.github.com/download and in `INSTALL` .

How can I uninstall Mininet?

If you installed Mininet using `apt-get install mininet` , you can uninstall it using:

```
apt-get remove mininet
```

If you installed from source, there isn't currently an automatic way to uninstall it.

(If someone would like to add reliable, verified uninstall target and/or `install.sh` option, we'd welcome a pull request!)

In the mean time, you might try something like:

```
sudo pip uninstall mininet
sudo rm `which mn`
sudo rm `which mnexec`
sudo rm /usr/share/man/man1/mn.1*
sudo rm /usr/share/man/man1/mnexec.1*
```

lather/rinse/repeat if you have multiple Mininet packages installed.

Note that this procedure will simply uninstall Mininet itself - it will not remove Open vSwitch, the Stanford reference switch or controller, or any other related software which may be installed on your system.

Help! I can't boot my VM in VirtualBox in Windows! Do I need a 32-bit VM?

If you are already running Microsoft's Hyper-V, you may not be able to boot the 64-bit Mininet VM in VirtualBox at the same time. I tested this and got the following error:

> VirtualBox - Error VT-X/AMD-V Hardware acceleration is not available on your system. Your 64-bit guest will fail to detect a 64-bit CPU and will not be able to boot.

Usually the problem is not that you don't have a 64-bit CPU (you probably do if you have anything as good as, say an intel Core 2 Duo from 2006.) It is much more likely that:

- You need to enable VT-X/AMD hardware virtualization in the BIOS, or:

- You are trying to run two virtual machine monitors - Hyper-V and VirtualBox - at the same time, and this does not work with 64-bit guest OSes.

  A simple solution to this problem is to turn off Hyper-V (in Windows 8, this is done via "Enable/Disable Windows Features" in the Windows Control Panel.) This has been tested and verified on Windows 8.

  If you need to keep running Hyper-V, you could try the 32-bit Mininet VM image, or you could run the Mininet VM image natively in Hyper-V by:

  i. Converting the `.vmdk` disk image to a `.vhd` using Microsoft's [Virtual Machine Converter (http://technet.microsoft.com/en-us/library/hh967435.aspx).
  ii. Creating a new Hyper-V virtual machine using the new `.vhd` image as its hard drive.
  iii. Creating an "external" virtual switch in Hyper-V manager, sharing the interface with the host OS.

I have a netbook from 2005 and I really want a 32-bit VM (or maybe I need one for testing or for nested virtualization on older hardware)

OK, use the 32-bit VM image. ;-)

Help! The VM console screen is blank!

Make sure that the VM is actually booting without any errors of any kind. If not, then you may have a problem actually booting the Mininet VM in your VM monitor. The 64-bit Mininet VM image should work on any modern CPU. However, some users have reported conflicts between Microsoft's Hyper-V and VirtualBox on Windows, so if you are running VirtualBox on Windows you may need to turn off Hyper-V. Alternately, you can use Hyper-V to run the VM as noted above! Or you can try the 32-bit image, which seems to work fine on most configurations.

If the VM is booting but boots to a blank screen, then you probably just need to either wake up the Linux console or switch to another virtual console, as follows:

First, select the VM console window. Second, press a key like `A` or `return` a few times - see if any text appears. If nothing happens, try switching to a different Linux console using ```control``alt``F1`` through``` control `alt` F7`.

Note: On a Mac laptop using VMware Fusion, you may need to type ```fn``control``option``F1``` , since the `F1` key controls brightness by default.

Help! I can't import the `.ovf` into VirtualBox/VMware/etc.!

Unfortunately, when VirtualBox was updated to 4.3.4, it stopped being able to import some versions of the Mininet `.ovf` file. Some other VMMs may also not be able to import it directly, but there is an easy workaround:

If you extract the `. ip` file, you should see a `.vmdk` disk image file. You should be able to create a new virtual machine in VirtualBox/VMware/etc. - and you should be able to specify that this new VM should use an existing disk image file, and you should select the Mininet `.vmdk` file. Configure, boot, and enjoy!

Help! I can't connect to the internet on the Mininet VM to install packages! (or, I can't SSH into the mininet VM from the host box)

In VirtualBox, you need two different network interfaces set up if you want to both access the internet from your VM and access your VM from the host. One of them should be a **NAT** interface (to get to the internet), and the other should be a **host-only** interface (to get to, well, the host). Set up the interfaces in VirtualBox that way, then add `eth0` and `eth1` lines in the VM's `/etc/network/interfaces` as below:

```
# The host-only interface
auto eth0
iface eth0 inet dhcp

# The internet interface
auto eth1
iface eth1 inet dhcp
```

Help! Mininet isn't working in an `lxc` container because of AppArmor!

ote that Mininet is itself a container orchestration system, so usually you don't want to run it inside another container system unless you are doing something unusual such as setting up a shared de elopment or lab ser er!

AppArmor's configuration for `lxc` seems to forbid recursive private mounts, which Mininet wants.

This may cause Mininet to hang on startup. Additionally `mnexec -n bash` will fail.

[Note: we should detect this failure.]

In Ubuntu 16.04 and later, this can be allowed by adding the following lines to an appropriate AppArmor configuration file (e.g. `/etc/apparmor.d/abstractions/lxc/container-default`):

```
# allow recursive private mounts  mininet wants this
mount options= rw, make-rprivate  -> **,
```

Then reload the appropriate profile, e.g.

```
apparmor_parser -r /etc/apparmor.d/lxc-containers
```

Bugs: This doesn't seem to work in 14.04 unfortunately.

How do I use Mininet's Python API?

Congratulations! You are asking the right question!! The Python API open's up Mininet's full potential.

Check out the Introduction to Mininet for an introduction to Mininet and its Python API.

Several useful examples of using the Python API can also be found in the `mininet/examples` directory.

We also provide Python DocStrings for every Mininet class and method, and you can view them using from within Python

```
>>> import mininet.node
>>> help mininet.node.Node
```

or by calling Python from within the Mininet CLI:

```
mininet> py help h2
```

In each case, pressing   should quit the pager.

The API documentation is also available at http://api.mininet.org.

Also see Mininet API Documentation for information on how to generate Mininet documentation yourself in .html and .pdf format.

Why can't I ping Google from my Mininet hosts?

You can't ping `google.com` because your Mininet network is not connected to the internet. This is usually a good thing! Usually Mininet networks use a non-routable IP address range like `10.0.0.0/8` .

However, you can set up NAT if you like.

How can I set up NAT?

In Mininet 2.2 and newer, you can use the `--nat` option:

```
mn --nat ...
```

In order for DNS to work in the Mininet hosts, you should not be using `dnsmas` for local caching. If you are running `NetworkManager` on Ubuntu, you can disable `dnsmas` by editing `/etc/NetworkManager/NetworkManager.conf` and making sure this line is commented out:

```
#dns=dnsmas
```

Then restart `NetworkManager` using:

```
sudo service NetworkManager restart  # or network-manager on releases prior to 16.0
```

Warning: By default, enabling NAT via `--nat` or the methods described below will reroute local traffic originating at your Mininet server or VM and destined for Mininet's IP subnet (`10.0.0.0/8` by default) to the
Mininet network, which can break connectivity if you are using addresses in the same range in your LAN. You can change this range using the `--ipbase` option, for example `--ipbase 10.2.2.0/2` .

You can also use `Mininet.addNAT` from the Python API:

```
net = Mininet  topo=...
net.addNAT  .configDefault
net.start
...
```

You can also add it into your topology; one possibility is something like:

```
class NatTopo  Topo  :
    def build  self, natIP='10.0.0.25 '  :
        self.hopts =   'defaultRoute': 'via '   natIP
        hosts  =   self.addHost  h   for h in 'h1', 'h2'
        s1 = self.addSwitch  's1'
        for h in hosts:
            self.addLink  s1, h
        nat1 = self.addNode  'nat1', cls=NAT, ip=natIP,
                             inNamespace=False
        self.addLink  nat1, s1
```

Or perhaps:

```
def Natted  topo lass  :
    "Return a customi ed Topo class based on topo lass"
    class NattedTopo  topo lass  :
        " ustomi ed topology with attached NAT"
        def build  self, *args, **kwargs  :
            """Build topo with NAT attachment
               natIP: local IP address of NAT node for routing  10.0.0.25
               connect: switch to connect  s1 """
            self.natIP = kwargs.pop  'natIP', '10.0.0.25 '
            self.connect = kwargs.pop  'connect', 's1'
            self.hopts.update  defaultRoute='via '   self.natIP
            super  NattedTopo, self  .build  *args, **kwargs
            nat1 = self.addNode  'nat1', cls=NAT, ip=self.natIP,
                                 inNamespace=False
            self.addLink  self.connect, nat1
    return NattedTopo

def natted  topo lass, *args, **kwargs  :
    " reate and invoke natted version of topo lass"
    topo lass = Natted  topo lass
    return topo lass  *args, **kwargs

topo = natted  TreeTopo, depth=2, fanout=2
net = Mininet  topo=topo
...
```

Mininet 2.1.0: Look at `examples/nat.py` .

Mininet 2.0 and earlier:

The illustrious Glen Gibb provided a script to do it back in 2011 on mininet-discuss. There was also a follow-up message from Leo Alterman.

Another updated version of the script can be found here: Mininet NAT Script.

Note that:

- The script assumes that `eth0` is the host interface connected to the internet/your LAN. You may need to change it if this is not the case!
- The script adds the following line to `/etc/network/interfaces` :

```
      iface root-eth0 inet manual
```

If this script does not work for you, please make an effort to debug and fix the problem, and then update this FAQ entry.

Which versions of OpenFlow does Mininet support?

The Ubuntu 14.04 VM uses that release's package for Open vSwitch 2.0.2, which supports 1.0 by default; experimental 1.3 support can be enabled using `--switch ovs,protocols=OpenFlow13` from the command line, or passing `protocols='OpenFlow13'` to the OVSSwitch constructor. For example:

```
switch = partial  OVSSwitch, protocols='OpenFlow13'
net = Mininet  topo, switch=switch ...
```

Open vSwitch 2.3 and newer support 1.3 by default. It is easy to install it using:

```
install.sh -V 2.3.1
```

What OpenFlow switch implementations does Mininet support?

Mininet currently includes support for the user space reference implementations, Open vSwitch in kernel and user space modes, and the Indigo Virtual Switch. The reference switch and OVS are included in the VM image, and IVS can easily be installed using `install.sh -i` . Mininet used to support the OpenFlow 0.8.9 kernel reference implementation (`--switch kernel`) but that is now obsolete and has largely been replaced with Open vSwitch.

The command line options are `--switch user` and `--switch ovsk` for the user reference and Open vSwitch kernel switches, respectively.

You can also install the CPqD ofsoftswitch13 switch using `install.sh -3f` ; it will replace the Stanford reference switch, i.e. `--switch user` and `serSwitch` . See below for an example of using it.

How can I use OpenFlow 1.3 only?

Usually the switch and controller will negotiate the highest version of OpenFlow that they both support.

If you wish to use OpenFlow 1.3, you should use a switch that supports it and a controller that supports it.

It's possible to use OVS in OpenFlow 1.3-only mode by specifying `protocols=OpenFlow13` and using a 1.3 compatible controller. For example:

```
sudo mn -v output --switch ovs,protocols=OpenFlow13 --controller ryu,simple_switch_13
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2
h2 -> h1
*** Results: 0  dropped  2/2 received
```

Additionally, the CpQD switch may be installed using `install.sh -3f` - it replaces the Stanford Reference switch.

```
mininet/util/install.sh -3f
...
sudo mn -v output --switch user  --controller ryu,simple_switch_13
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2
h2 -> h1
*** Results: 0  dropped  2/2 received
```

Why does my controller, which implements an Ethernet bridge or learning switch, not work with my network which has loops in it? I can't ping anything!

tl;dr: use `--switch lxbr,stp=1` or `--switch ovsbr,stp=1` and wait for STP to converge.

It doesn't work because your network has loops in it.

Transparent bridging of L2/Ethernet networks doesn't work if the topology has loops in it, for a variety of reasons: ARP does broadcasts, packets are flooded by default, learning switches don't deal well with seeing the same MAC address on multiple ports and could potentially learn a route to themselves, and Ethernet frames don't have a time to live field (TTL) the way IP packets do (otherwise flooding might work, if inefficiently.) As a result, many Ethernet bridges implement variants of a Spanning Tree Protocol (STP), which simply deactivates links in the network to remove loops. Of course, this also throws away network bandwidth that you could otherwise be using, and creates a bottleneck at the root of the tree!

The OpenFlow reference controller (`controller`) implements a bridge/learning switch, as does `ovs-controller` and/or `test-controller` (currently the default controllers for Mininet), as does NOX's `pyswitch` module, and they don't implement a spanning-tree protocol by default. As a result, they **will not work with a network that has loops in it**.

In general, if you want to use a network with loops in it, you need to be absolutely sure that your controller supports such a network. As mentioned above, `ovs-controller` , `controller` and `pyswitch` **do not** by default. POX includes a spanning tree module, and other controllers (Floodlight, ONOS, ODL, etc.) **may** support multipath and/or spanning tree - you will want to consult the documentation for your controller, make sure it is configured correctly to support multipath or spanning tree, and test it to make sure that it actually works. A simple test is to use RemoteController pointed at your controller and use the `torus` topology, e.g.:

```
sudo mn --topo torus,3,3  --controller remote,ip=<controller ip address>,port=<controller
port>
```

Please feel free to fill in this chart with test results from various controllers:

controller	version	topo	result	details
ONOS	1.0	`--torus 8,8`	success	need to create proactive routes or start reactive forwarding
OpenDaylight	Beryllium	`--torus 3,3`	success	support for looping topologies

If you just want to get your network "working", you can run STP. In Mininet 2.2 you can use the Linux bridge:

```
sudo mn --topo torus,3,3 --switch lxbr,stp=1
```

In the current master branch, you can also use OVS in bridging mode:

```
sudo mn --topo torus,3,3 --switch ovs,failMode=standalone,stp=1
```

or the more compact:

```
sudo mn --topo torus,3,3 --switch ovsbr,stp=1
```

You will need to **wait for STP to converge**. You can observe its progress with `sh brctl show s1` (`LinuxBridge`) or `sh ovs-ofctl show s1`) (`OVSBridge` .) You can also call `net.wait onnected` to wait for STP to converge:

```
mininet> py net.wait onnected
```

Note that if you are running a *remote* controller (rather than a local OVS or Linux bridge as suggested here), `wait onnected` will only wait for the switches to *connect* to your controller. If you are using a remote controller, you should check the controller console or logs for any updates.

As noted above, running spanning tree removes any performance improvement from multipath networks, although it can still provide redundancy for reliability (if you deactivate a link, STP can compute a new spanning tree that uses a different link and restores connectivity.) If this sounds terrible, it's because it is - one of the advantages of using a multipath-capable OpenFlow controller is that you can potentially escape the tyranny of Spanning Tree!

If you wish to code your own multipath-capable controller in POX, you may also wish to take a look at RipL-POX, which provides starter code for a multipath-capable controller, as well as some of the multipath experiments on http://reproducingnetworkresearch.wordpress.com . But, you will still probably have to **do some work** and **actually understand what you are doing**.

Help! Mininet is hanging on startup!

Over time, we should implement more error checking.

For now, here are some things to check to make sure that Mininet is working correctly:

1. Make sure `mnexec` is working

   The results of `sudo mnexec -n ifconfig -a` should be something like

   ```
   lo        Link encap:Local Loopback
             LOOPBA    MT :65536  Metric:1
             RX packets:0 errors:0 dropped:0 overruns:0 frame:0
             TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
             collisions:0 tx ueuelen:1
             RX bytes:0  0.0 B   TX bytes:0  0.0 B
   ```

2. Make sure Open vSwitch is running:

   ```
   sudo ovs-vsctl show
   5099b 6d-00 b- bf5-a6b3-60510b6fc88a
       ovs_version: "2.5.0"
   ```

You can further troubleshoot Mininet startup by running `mn -v debug` and carefully looking at the output for error messages.

Why do the switch data ports have random MAC addresses? How do I assign MAC and IP addresses to the switch data ports?

The MAC addresses reported by Linux for the switch data ports are meaningless. The switch is controlled by OpenFlow, so you should use OpenFlow to ensure that any packets destined "for the switch" are properly routed. You "assign" MAC and IP addresses "to the switch" by using OpenFlow rather than the Linux IP stack. If you attempt to use the Linux IP stack instead, it really won't work unless you are using the Linux kernel for routing (which you aren't - you're using an OpenFlow switch!) You should never attempt to use `ifconfig` or `ip addr` or other utilities to assign an IP address to a switch data port that is connected to a host or another switch (`veth` interfaces are not bridges!) Usually you will want your controller to handle packets such as ARP and ICMP which are sent to and from "the switch," and you will want IP packets which are sent to to be handled by appropriate flow table entries. You can pick any "MAC" address you like for the switch.

Why does my network fail if I use more than 16 switches?

For Mininet 2.0.0, the default controller for the `mn` command is `ovs-controller` (which can be installed automatically in Ubuntu.) Unfortunately `ovs-controller` only supports up to 16 switches. If you want to use more than 16 switches, you should use a controller that supports more than 16 switches, for example:

```
sudo mn --controller ref --topo linear,20 --test pingall
```

Make sure you've installed the Mininet version of the OpenFlow reference controller, which is easily done using:

```
mininet/util/install.sh -f
```

You can also create a custom controller class or use `--controller external:IP` and use any custom or off-the-shelf controller that you like. For example, ou can easily install POX by checking it out or using `util/install.sh -p`, and you can install Floodlight on Ubuntu using `apt-get install floodlight`.

If you are using the default controller or any controller which implements an Ethernet bridge (aka learning switch), be sure that your network does not have loops in it or that you have activated spanning tree.

How can I control my Mininet hosts remotely?

It's trivial to control Mininet hosts from the CLI or from within a Python script running locally, but what if you want some other process or even another computer on your LAN to be able to control your Mininet network remotely?

Well, there are lots of ways to do this. One idea is that anything you can do in Python, you can do in Mininet, and it's often very easy to do so. For example, there are all sorts of frameworks available for any kind of messaging you can imagine. (See below for a REST example - it's just a few lines of code.)

Another easy way to control Mininet hosts in the current `master` branch is to use the `util/m` script.

For example if my Mininet server is `ubuntu1`, I can run `ifconfig` on host `h1` using

```
$ ssh ubuntu1 mininet/util/m h1 ifconfig
```

Another way is to actually connect your Mininet network to your LAN and to run `sshd` on your Mininet hosts. This is left as an exercise for the reader, but you may want to look at the `hwintf.py` and `sshd.py` scripts in `examples/` to understand how you might possibly do this.

How can I add a REST interface to Mininet?

Basically anything you can do in Python you can do in Mininet, and it's often very easy to do so. For example, there are all sorts of frameworks available for various kinds of messaging and RPC, REST, JSON, SOAP, XML, etc.. You can hook Python code up to node.js, you can have it speak ZeroMQ, you can use Apache Thrift... really the possibilities are endless!!

Note however that if you are running locally it's much easier to control Mininet directly from within a Python script or using the CLI.

It's trivial to add a REST (or ReST if you prefer) API to Mininet using Python. For example, using the Bottle framework, you could do something like:

```
# /usr/bin/python

from mininet.net import Mininet
from mininet.topo import SingleSwitchTopo

from bottle import route, run, template

net = Mininet  topo=SingleSwitchTopo  2

@route '/cmd/<node>/<cmd>'
def cmd  node='h1', cmd='hostname'  :
    out, err, code = net.get  node  .pexec  cmd
    return out   err

@route '/stop'
def stop  :
    net.stop
net.start
run host='localhost', port=8080
```

This allows you to send simple commands to your Mininet hosts.

ote that `host.pexec()` like `host.cmd()`) runs commands as root, so this isn't really something you want to expose to the whole internet though you could firewall port on your Mininet ser er and then use `ssh` for a secure connection.) ut it's quite con enient, isn't it n a real example you would probably want a method to shut down both the ST ser er and the Mininet network in a graceful manner rather than using control C and `mn -c` .

After running this script in one window:

```
$ sudo ./rest.py
```

You can easily try it out. For example, you could run the `ifconfig h1-eth0` command on host `h1` as follows:

```
$ curl localhost:8080/cmd/h1/ifconfig 20h1-eth0
h1-eth0   Link encap: thernet  H addr 36:6f:c0:28:a3:f9
          inet addr:10.0.0.1  Bcast:10.255.255.255  Mask:255.0.0.0
           P BROAD AST R NNING M LTI AST  MT :1500  Metric:1
          RX packets:  errors:0 dropped:0 overruns:0 frame:0
          TX packets:3 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 tx ueuelen:1000
          RX bytes:328  328.0 B   TX bytes:238  238.0 B
```

Note that you need to escape the space in `ifconfig h1-eth0` as `20` as you would normally in a URL.

How do I run the Mininet examples?

1. Fetch the Mininet source tree if you haven't already

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

2. Run an example

   ```
   sudo mininet/examples/treeping6 .py
   ```

Note some examples (`consoles.py` , `miniedit.py`) require you to have set up X11 forwarding. Others require that you have installed the OpenFlow reference implementation using `mininet/util/install.sh -f` . Consult the comments in the example's `.py` file for details.

Why doesn't `dpctl` work? How can I dump a switch's flow table?

`dpctl` should work fine with the Stanford OpenFlow reference implementation or the CPqD version of same.

But **you probably don't want to use `dpctl` at all - use `ovs-ofctl` instead!** Especially if you're running install - `ovs-ofctl` it's particularly easy to use with OVS and, importantly, will actually dump the complete flow table (unlike `ovs-dpctl` !)

If you're running Open vSwitch, or need to open up a listening port on either OVS or the reference switch so that you can connect to a port, read on...

1. If you are using Open vSwitch, the correct command to use from the shell prompt is

   ```
   $ sudo ovs-ofctl dump-flows s1
   ```

   (If you try to use `ovs-dpctl` with Open vSwitch, you will only see the cached flows in the kernel, rather than the switch's full flow table.)

2. The correct command to use from the Mininet prompt is

   ```
   mininet> dpctl dump-flows
   ```

   which dumps all the flows on all switches, and works for both OVS and the reference switch.

   If you're running OVS, you can pass the switch name to `ovs-ofctl` and it will connect to it via the file system:

   ```
   mininet> sh ovs-ofctl dump-flows s1
   ```

   or, from the shell prompt:

   ```
   $ ovs-ofctl dump-flows s1
   ```

3. If you want to open up a listening port on the switch, you need to specify the base listening port, e.g.

   ```
   net = Mininet  topo=topo, listenPort=663
   ```

   Ports will be allocated sequentially starting with the value you specify.

   Note if you want to dump the flows from the reference switch, you will need to have a listening port opened up; then you can use `dpctl` :

   ```
   $ dpctl dump-flows tcp:localhost:663
   ```

   Note that `ovs-ofctl` doesn't like `localhost` , so you should use 127.0.0.1:

   ```
   $ ovs-ofctl dump-flows tcp:12 .0.0.1:663
   ```

How do I generate traffic on my Mininet network?

Asking this question usually means you haven't read or understood the Documentation or indeed this FAQ.

Mininet runs pretty much any Linux program. So, you can use pretty much any client or server program you can think of (e.g. `ping` , `iperf` , `wget` , `curl` , `netperf` , `netcat` etc..) You can easily capture traffic using programs like `tcpdump` and `wireshark` .

You probably should do a Google search on something like linux generate packets or linux traffic generator.

It's also easy to generate packets in Python using `scapy` .

if you want to generate or decode OpenFlow messages, you should look at various controller frameworks like POX or OpenFlow messaging libraries like OpenFlowJ or LOXI.

How do I modify packet headers using Mininet?

Asking this question usually means you haven't read or understood the documentation and that you don't understand what OpenFlow is.

Use OpenFlow. Please go through the OpenFlow tutorial and consult the OpenFlow specification.

How do I implement a custom routing algorithm?

Asking this question usually means that you haven't read or understood the documentation and that you don't understand what OpenFlow is.

Use OpenFlow. Please go through the OpenFlow tutorial and consult the OpenFlow specification.

How do I update to a new version of Mininet?

What you need to do depends on how you installed Mininet:

1. If you are upgrading from Mininet 1.0.0 and/or an old version of OVS compiled in `/usr/local`, make sure you remove all traces of the old Mininet and OVS

```
sudo rm -rf /usr/local/bin/mn /usr/local/bin/mnexec
    /usr/local/lib/python*/*/*mininet*
    /usr/local/bin/ovs-* /usr/local/sbin/ovs-*
```

2. If you are upgrading from a package install of Mininet, you should remove the old Mininet and OVS packages:

```
sudo apt-get remove mininet openvswitch-switch
```

3. You should now be able to install from source as per the instructions on http://mininet.org/download/

If you wish to install a newer version of OVS than the vendor-supplied version, you may wish to follow the instructions here .

In addition to the `mininet-discuss` mailing list, is there a `#mininet` IRC channel?

According to Nick Bastin, there is a fair amount of expertise on the `#openflow` channel on FreeNode, so you may want to consider joining it.

Also, by request, we have created a `#mininet` IRC channel for additional Mininet-specific discussion.

(Note: this is an experiment, and of course we cannot guarantee that anyone is logged on! But if you want to chat about Mininet on IRC, you may wish to look at `#mininet` and/or `#openflow` .)

Can I turn on SSL for Open vSwitch?

Yes, Open vSwitch and ovs-controller both support SSL. It isn't turned on by default in Mininet. For an example, look [here] (SSL-on-Open-vSwitch-and-ovs-controller).

Why doesn't pmonitor display any output for some Python commands?

If you try to recreate the simple web server and client example from the walkthrough with Mininet's Python API you may find that `util.pmonitor` blocks and returns no output (not even the expected `Serving HTTP on 0.0.0.0 port 80`). This is because, for performance reasons, Python buffers its output to `stdout` (as discussed here). You simply need to pass the `-u` flag, as shown in the example below.

```
from mininet.net import Mininet
from mininet.topo import SingleSwitchTopo
import mininet.util as util
from time import sleep


def main  :
    topo = SingleSwitchTopo hosts=2
    net = Mininet topo=topo
    net.start

    http_client = net.hosts 0
    http_server = net.hosts 1

    popens =
    popens http_server  = http_server.popen "python -u -m SimpleHTTPServer 80"
    sleep 1   #  ait for the server to start up.
```

```
        popens http_client  = http_client.popen "wget -O -   ".format http_server.IP

        try:
            for host, line in util.pmonitor popens :
                if host:
                    print host.name, line
        finally:
            # Don't leave things running if this script crashes.
            for process in popens.values  :
                if not process.poll  :
                    process.kill
            net.stop

    if __name__ == '__main__':
        main
```

How can I use multiple controllers in my network?

One way is to make a custom switch class, as shown in `examples/controllers.py`

Another way is to use the mid-level API, as shown in [`examples/controllers2.py`] (https://github.com/mininet/mininet/tree/master/examples/controllers2.py)

Can I upgrade Open vSwitch to a newer version?

Mininet usually uses the latest version of Open vSwitch that is included in that distribution's release. To find out what version you're running, you can use

```
ovs-vsctl --version
```

`install.sh` includes an option to easily upgrade OVS to a new or different version:

```
install.sh -V 2.3.1
```

Help! I updated my Ubuntu kernel and now Open vSwitch won't start!

If you are using Ubuntu's `openvswitch-datapath-dkms` and `openvswitch-switch` packages, they should update automatically when you reboot.

Until `openvswitch-datapath-dkms` is reconfigured/rebuilt, Open vSwitch will refuse to start saying that its kernel module is missing. For example, you may see a message like:

```
FATAL: Module openvswitch not found.
```

If rebooting doesn't fix the problem, or if you don't want to reboot, you can reconfigure the kernel module manually and restart OVS:

```
sudo dpkg-reconfigure openvswitch-datapath-dkms
sudo service openvswitch-switch restart
```

Help! I don't understand OpenFlow or SDN!

There is a wealth of useful information to be found at http://opennetworking.org.

Definitely read the OpenFlow White Paper and go through the OpenFlow Tutorial.

You may also wish to search the ACM and IEEE digital libraries for recent papers that reference Software-Defined Networking and OpenFlow.

I [BL] also highly recommend Nick Feamster's SDN course on Coursera.

Help! I don't understand networking (or maybe computers) at all!!

There is a wealth of very useful information available for free on the internet - try some Google (or search engine of your choice) searches for things like 'networking tutorial', 'IP networking basics', etc..

You may also find useful content on sites like Wikipedia, About.com and YouTube, as well as free online courses. Universities like Stanford, Berkeley, MIT (and many others) all offer free on-line courses, and free online courses are also available through iTunes U, Coursera, etc.. You can learn Python on sites like Codecademy or Khan Academy, Nick Parlante's course on developer.google.com etc. - really there has been no better time for free on-line education.

My [BL's] personal recommendation is to take an introductory CS course and an introductory networking course at your local college or university, but the free on-line options are pretty cool as well.

You may also wish to consult an introductory networking textbook such as *Computer etworks Top own pproach* by Kurose and Ross, or *Computer etworks Systems pproach* by Peterson and Davie. To master Ethernet switches you might want to check out *The ll ew Switch ook* by Seifert, and for a vintage but still classic and somewhat relevant view of socket programming you might find a copy of the *nix etwork rogramming* tomes by the late Richard Stevens.

Help! I have never used Linux or Unix before!

How do I use (some Linux command)?

There is a wide variety of very useful information available for free on the internet. Try searching for "Linux tutorial" or "Ubuntu tutorial" in Google or your search engine of choice.

Additionally, nearly every Unix system since the beginning of time includes online documentation which can be accessed using the `man` command. For example, to find out about the `ls` command, you can type

```
man ls
```

Each section (traditionally 1-8) of the manual has an `intro` page, and you can actually read *all* of the intro sections by typing:

```
man -a intro
```

Some GNU software hides its documentation in the (powerful but less friendly) `info` documentation system. Usually there is a `man` page which will direct you to it.

Additionally, `bash` has a `help` command which can be used to find out how to use shell commands.

There are also many useful books on Linux available on Amazon and in your local bookstore. Mark Sobell's books are classics, as are Nemeth's books on Unix/Linux system administration.

Will you do my (home)work assignment (or paper/thesis/project/etc.) for me? It's due next Tuesday at 4pm.

No, because:

- We don't want to and don't have time
- You don't have enough money to pay our consulting rates anyway
- It would be a violation of the honor code
- You will learn much more doing the assignment yourself
- You really want to learn how to use Google scholar, citeseer, various digital libraries from the ACM, IEEE, and USENIX, and also learn how to do a literature search
- It's the job of your advisor/TAs/school/company/self to teach/train you, so make them do their job!

Mininet Project