**Mininet Overview**

Mininet is a *network emulator* which creates a network of virtual hosts, switches, controllers, and links. Mininet hosts run standard Linux network software, and its switches support OpenFlow for highly flexible custom routing and Software-Defined Networking.

Mininet supports research, development, learning, prototyping, testing, debugging, and any other tasks that could benefit from having a complete experimental network on a laptop or other PC.

Mininet:

* Provides a simple and inexpensive **network testbed** for developing OpenFlow applications
* Enables **multiple concurrent developers** to work independently on the same topology
* Supports **system-level regression tests**, which are repeatable and easily packaged
* Enables **complex topology testing**, without the need to wire up a physical network
* Includes a **CLI** that is topology-aware and OpenFlow-aware, for debugging or running network-wide tests
* Supports **arbitrary custom topologies**, and includes a basic set of **parametrized topologies**
* is **usable out of the box** without programming, but
* also Provides a straightforward and extensible **Python API** for network creation and experimentation

Mininet provides an easy way to get correct system *behavior* (and, to the extent supported by your hardware, performance) and to experiment with topologies.

Mininet networks run *real code* including standard Unix/Linux network applications as well as the real Linux kernel and network stack (including any kernel extensions which you may have available, as long as they are compatible with network namespaces.)

Because of this, the code you develop and test on Mininet, for an OpenFlow controller, modified switch, or host, *can move to a real system with minimal changes*, for real-world testing, performance evaluation, and deployment. Importantly this means that a design that works in Mininet can usually move directly to hardware switches for line-rate packet forwarding.

**How it Works**

Nearly every operating system virtualizes computing resources using a *process* abstraction. Mininet uses *process-based virtualization* to run many (we’ve successfully booted up to 4096) hosts and switches on a single OS kernel. Since version 2.2.26, Linux has supported *network namespaces*, a lightweight virtualization feature that provides individual processes with separate network interfaces, routing tables, and ARP tables. The full *Linux container architecture* adds **chroot()** jails, process and user namespaces, and CPU and memory limits to provide full OS-level virtualization, but Mininet does not require these additional features. Mininet can create kernel or user-space OpenFlow switches, controllers to control the switches, and hosts to communicate over the simulated network. Mininet connects switches and hosts using virtual ethernet (**veth**) pairs. While Mininet currently depends on the Linux kernel, in the future it may support other operating systems with process-based virtualization, such Solaris containers or !FreeBSD jails.

Mininet’s code is almost entirely Python, except for a short C utility.

**Why it’s better**

Mininet combines many of the best features of emulators, hardware testbeds, and simulators.

Compared to full system virtualization based approaches, Mininet:

* **Boots faster**: seconds instead of minutes
* **Scales larger**: hundreds of hosts and switches vs. single digits
* **Provides more bandwidth**: typically 2Gbps total bandwidth on modest hardware
* **Installs easily**: a prepackaged VM is available that runs on VMware or VirtualBox for Mac/Win/Linux with OpenFlow v1.0 tools already installed.

Compared to hardware testbeds, Mininet

* is **inexpensive** and **always available** (even before conference deadlines)
* is **quickly reconfigurable and restartable**

Compared to simulators, Mininet

* runs **real, unmodified code** including application code, OS kernel code, and control plane code (both OpenFlow controller code and Open vSwitch code)
* easily **connects to real networks**
* offers **interactive performance** - you can type at it

**Limitations**

Mininet-based networks cannot (currently) exceed the CPU or bandwidth available on a single server.

Mininet cannot (currently) run non-Linux-compatible OpenFlow switches or applications; this has not been a major issue in practice.