

Universität Stuttgart

Institute of Parallel and
Distributed Systems (IPVS)

Universitätsstraße 38
D-70569 Stuttgart

Tutorial: Software-defined Networking

Part 3: Mininet, ovs-ofctl, cURL

Frank Dürr

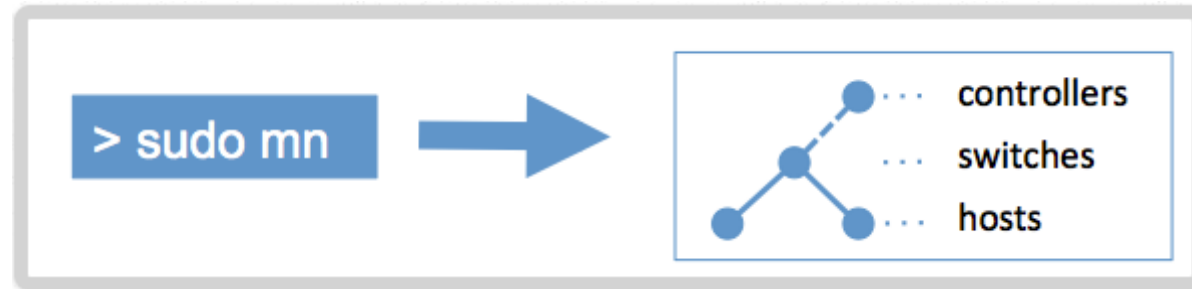
Acknowledgements / other contributors:
Sukanya Bhowmik, Ben Carabelli, Thomas Kohler

Overview

- **Mininet**
- Mininet Python API
- ovs-ofctl
- cURL



What is Mininet?



- Mininet is a network emulator
- Mininet creates realistic virtual networks on a single machine
 - Virtual OpenFlow switches
 - Virtual hosts
 - Virtual network interfaces



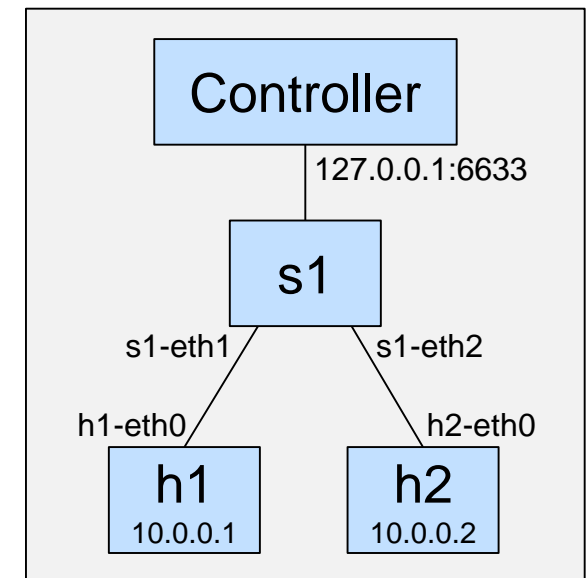
Starting Mininet

- To start a virtual network, run **mn** with super user permissions:

```
~$ sudo mn
```

- Starts a simple forwarding controller
- Starts minimal network topology
- Brings up the Mininet CLI

```
mininet> _
```



Default Mininet topology



Mininet Command Line Arguments

- h, --help** display (brief) Mininet usage help
- mac** use easy-to-read MAC addresses
(e.g. 00:00:00:00:00:01 for 10.0.0.1, etc.)
- arp** populate host ARP tables with static entries for all other hosts
- switch ...** choose switch implementation
ovsk for the Open vSwitch kernel module
- controller ...** choose controller
remote for controller on 127.0.0.1:6633
remote, ip=[ip], port=[port] for controller on [ip]:[port]
- topo ...** set network topology (e.g. **single**, **linear**, **tree**, ...)
- custom ...** load custom topology script
- c, --clean** shut down any previous Mininet instances



The Mininet CLI (1)

- `mininet> help`

List available commands

- `mininet> <host> <command> {<args>}`

Execute a command on a particular host, e.g.

- `mininet> h1 ip address`

- If first <arg> is node name (h1, h2, ...), its IP address is substituted:

```
mininet> h2 ping h3
```

- `mininet> xterm h1 [h2 [h3 ...]]`

Launch an XTerm terminal for each of the given hosts
(to execute commands on specific host more conveniently)



The Mininet CLI (2)

- `mininet> nodes`

List all nodes (hosts, switches, controller)

- `mininet> net`

List network connections (topology)

- `mininet> pingall`

Ping all pairs of hosts (reachability test)

- `mininet> exit`

guess what ;-)

Overview

- Mininet
- **Mininet Python API**
- ovs-ofctl
- cURL



Mininet Python API (1)

- **mn** is just a Python script...
- Mininet's Python API can be used
 - to define custom topologies
 - to run customized tests
- Important classes:
 - **Mininet** (wrapper for a network instance)
 - **Node** (a node in the network)
 - **Host** (a node with IP and MAC addresses, can execute commands)
 - **Switch** (a node with a DPID)
 - **CLI** (a command line interface for mininet)



Mininet Python API (2)

The **Mininet** class

- To create a new Mininet:

```
net = Mininet( <options> )
```

where <options> is a list of **key=value** pairs, e.g.

```
switch=OVSSwitch, controller=RemoteController, ...
```

- To add a Host to the network and store it in a variable **h**:

```
h = net.addHost( <name> [, <options> ] )
```

where <name> is just a string, e.g. **'h1'**,

and <options> can be IP and MAC addresses

- Similarly, to add a switch or controller:

```
s = net.addSwitch( <name> [, <options> ] )
```

```
c = net.addController( <name> [, <options> ] )
```



Mininet Python API (3)

- To add a link between two nodes:
`net.addLink(<node1>, <node2>)`
where <node1> and <node2> are either Host or Switch objects
- To initialize the network:
`net.build()`
this sets up all nodes and links of the network
- To start the network:
`net.start()`
- To bring the network and all associated processes down cleanly:
`net.stop()`



Mininet Python API (4)

The **Host** class has the following methods:

- `h.setIP(<string>)` sets the IP address
- `h.setMAC(<string>)` sets the MAC address
- `ip = h.IP()` gets the IP address (as string)
- `mac = h.MAC()` gets the MAC address (as string)
- `h.setARP(<IP>, <MAC>)`
adds an entry to the ARP cache of node `h`
- `h.cmd(<command>)`
executes `<command>` in the shell associated with node `h`
e.g. `h1.cmd('ping ' + h2.IP())`



Mininet Python API (5)

- The **CLI** class brings up a command line interface for the net
- To start the CLI, simply instantiate it:

- `CLI(net)`
- Returns once the user quits the CLI by typing

```
mininet> exit
```

- **More Information:**

- <https://github.com/mininet/mininet/wiki/Introduction-to-Mininet>
- <http://mininet.org/api/annotated.html>

- **Attention:** In Python, indentation matters! (If you're not defining functions/classes or doing flow control, don't indent lines.)



Overview

- Mininet
- Mininet Python API
- **ovs-ofctl**
- cURL



ovs-ofctl

- Command line tool
- For monitoring and administering OpenFlow switches
 - Comes with Open vSwitch (ovs-...)
 - Can be used also with other OpenFlow switches
- Often used for administering local switches on same machine
 - Open vSwitch(es) instances on a Linux machine
 - CLI of whitebox hardware switches
 - Switches of emulated Mininet network



Show Switch Information

```
$ sudo ovs-ofctl -OOpenFlow13 show s1
```

- OpenFlow version (here 1.3): -OOpenFlowXX
 - Version should match switch capabilities



Querying Flows of Switch

```
$ sudo ovs-ofctl -OOpenFlow13 dump-flows s1
```

Shows:

- Flow table entries
 - With all details: also information that you might have specified only implicitly when creating the flow (e.g., when you leave out priorities)
- How often did each flow table entry match incoming packets
 - Might be hand for testing whether your flow table entries are effective

Pushing New Flows to Switch

Add one specific flow to a switch (s1):

```
$ sudo ovs-ofctl -OOpenFlow13 add-flow s1 \  
    <matches>,actions=<list of actions>
```

- With priorities ([0,65535]): `priority=value`

Add multiple flows as defined in a file (one flow per line) to a switch:

```
$ sudo ovs-ofctl -OOpenFlow13 \  
    add-flow switch - < file
```

Note “\
” is a line break. You can also write this in one line without “\
”.



Deleting Flows from a Switch

Deleting all flows from a switch (s1):

```
$ sudo ovs-ofctl -OOpenFlow13 del-flow s1
```

Deleting a specific flow from a switch (s1):

```
$ sudo ovs-ofctl -OOpenFlow13 del-flow s1 \  
    <matches>,actions=<list of actions>
```



Overview

- Mininet
- Mininet Python API
- ovs-ofctl
- **cURL**



cURL

- Command line tool for posting HTTP requests

```
~$ curl -u [user]:[pw] -H [hdr] -X [cmd] -d [data] [URL]
```

- Used here to access Floodlight's RESTful web API:

- HTTP GET requests (e.g., query information), e.g.

```
~$ curl http://localhost:8080/wm/device/  
~$ curl http://localhost:8080/wm/core/controller/  
switches/json
```

- HTTP POST requests (e.g., install flows)

```
~$ curl -X POST -d '{<JSON attributes>}'  
http://localhost:8080/wm/staticentrypusher/json
```



cURL

- HTTP DELETE requests (e.g., remove flows), e.g.

```
~$ curl -X DELETE -d '{"name":"..."}'  
http://localhost:8080/wm/staticcentrypusher/json
```

- HTTP PUT requests (e.g., enable statistics collection)

```
~$ curl -X POST -d ''  
http://localhost:8080/wm/statistics/config/  
enable/json
```



JSON Pretty Printing

- REST interface usually returns JSON without whitespace
- JSON can be formatted for better readability by piping cURL output through `python -m json.tool`

```
~$ curl http://localhost:8080/wm/core/controller/  
switches/json | python -m json.tool
```



Questions?

