OpenFlow Tutorial

Future Internet Communication Technologies

Prof. Dr. Panagiotis Papadimitriou
David Dietrich

http://www.openflow.org

Experimental Setup

- Virtual machine running in VirtualBox
  - SDNHub ODL tutorial VM*
  - Login (user/pw): *ubuntu / ubuntu*

- Network topology emulated by **Mininet**
- OpenFlow switch: **OpenvSwitch** (OVS)
- OpenFlow controller: **RYU**

* *http://sdnhub.org/tutorials/sdn-tutorial-vm-64bit*

Network Topology, Mininet

- Topology creation with Mininet:

```
$ sudo mn --topo single,3 --mac --switch ovsk --controller remote
```

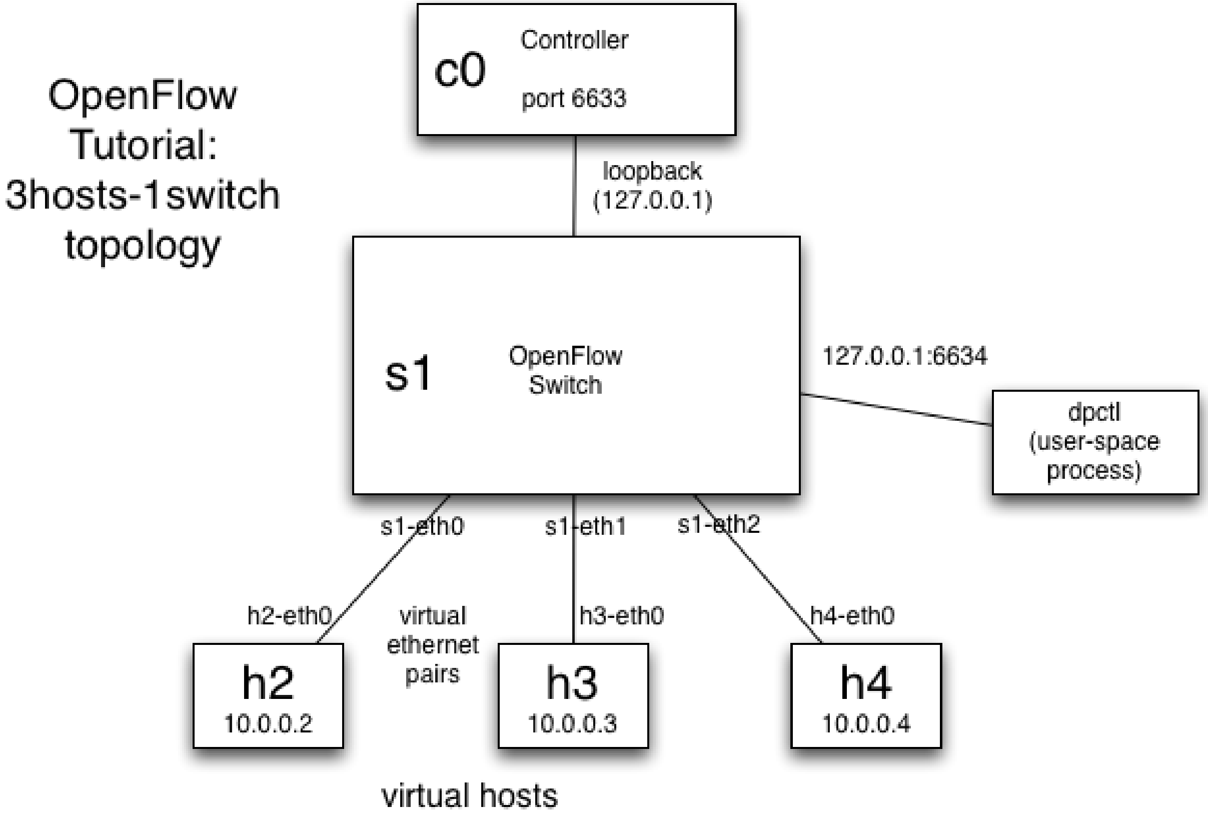


Figure source: http://www.openflow.org/wk/index.php/OpenFlow_Tutorial

Mininet Console

- Available commands in the Mininet console
  - Node and link list

```
mininet> dump
<Host h1: h1-eth0:10.0.0.1 pid=2680>
<Host h2: h2-eth0:10.0.0.2 pid=2681>
<Host h3: h3-eth0:10.0.0.3 pid=2682>
<OVSSwitch s1: lo:127.0.0.1,s1-eth1:None,s1-eth2:None,s1-eth3:None pid=2685>
<RemoteController c0: 127.0.0.1:6633 pid=2673>

mininet> net
h1 h1-eth0:s1-eth1
h2 h2-eth0:s1-eth2
h3 h3-eth0:s1-eth3
S1 lo:  s1-eth1:h1-eth0 s1-eth2:h2-eth0 s1-eth3:h3-eth0
c0
```

Mininet Console

- Available commands in the Mininet console
  - *ifconfig* : network interface configuration
  - *ping* : send echo requests to network hosts
  - *iperf* : perform network throughput tests
  - *dpctl* : admininister OpenFlow datapaths
  - Complete list or command description:

```
mininet> help [command]
```

- Exiting the Mininet console:

```
mininet> exit
```

- Cleaning up for restart:

```
# mn -c
```

Mininet Console

- Some commands can be run from the different hosts that are emulated by Mininet

```
mininet> h2 ifconfig
h2-eth0        Link encap:Ethernet  HWaddr 00:00:00:00:00:02
               inet addr:10.0.0.2  Bcast:10.255.255.255  Mask:255.0.0.0        …
lo             Link encap:Local Loopback
               inet addr:127.0.0.1  Mask:255.0.0.0      …

mininet> h2 ping -c3 h3
PING 10.0.0.3 (10.0.0.3) 56(84) bytes of data.
64 bytes from 10.0.0.3: icmp_req=2 ttl=64 time=0.402 ms
64 bytes from 10.0.0.3: icmp_req=3 ttl=64 time=0.077 ms
--- 10.0.0.3 ping statistics ---3 packets transmitted, 2 received, 33% packet loss, time 2006ms
    rtt min/avg/max/mdev = 0.077/0.239/0.402/0.163 ms
```

OpenFlow Datapath

- Use *dpctl* to administer OpenFlow datapaths

- Example: simple forwarding between h2 and h3
  - Identify OF switch port numbers for
    - s1-eth2 (link to h2)
    - s1-eth3 (link to h3)
  - Add flow entries
    - Incoming packets at s1-eth2 -> forward to s1-eth3
    - Incoming packets at s1-eth3 -> forward to s1-eth2
  - Check whether h3 is reachable from h2

OpenFlow Datapath

```
mininet> s1 dpctl show tcp:127.0.0.1:6634

OFPT_FEATURES_REPLY (xid=0x2): dpid:0000000000000001
…
 1(s1-eth1): addr:4e:1a:36:f9:2c:81
…
 2(s1-eth2): addr:3a:0f:c2:b0:51:68
     config:     0
     state:      0
     current:    10GB-FD COPPER
     speed: 10000 Mbps now, 0 Mbps max
 3(s1-eth3): addr:b2:6e:a4:b4:b0:52
     config:     0
     state:      0
     current:    10GB-FD COPPER
     speed: 10000 Mbps now, 0 Mbps max
…
```

OpenFlow Datapath

```
mininet> s1 dpctl add-flow tcp:127.0.0.1:6634 in_port=2,actions=output:3
mininet> s1 dpctl add-flow tcp:127.0.0.1:6634 in_port=3,actions=output:2
mininet> h2 ping -c3 h3
PING 10.0.0.3 (10.0.0.3) 56(84) bytes of data.
64 bytes from 10.0.0.3: icmp_req=1 ttl=64 time=1.41 ms
64 bytes from 10.0.0.3: icmp_req=2 ttl=64 time=0.069 ms
…
--- 10.0.0.3 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2004ms
…
mininet> s1 dpctl dump-flows tcp:127.0.0.1:6634
*** s1 ------------------------------------------------------------------------
NXST_FLOW reply (xid=0x4):
 cookie=0x0, duration=28.007s, table=0, n_packets=5, n_bytes=378, idle_age=7, in_port=3 actions=output:2
 cookie=0x0, duration=33.471s, table=0, n_packets=5, n_bytes=378, idle_age=7, in_port=2 actions=output:3
```

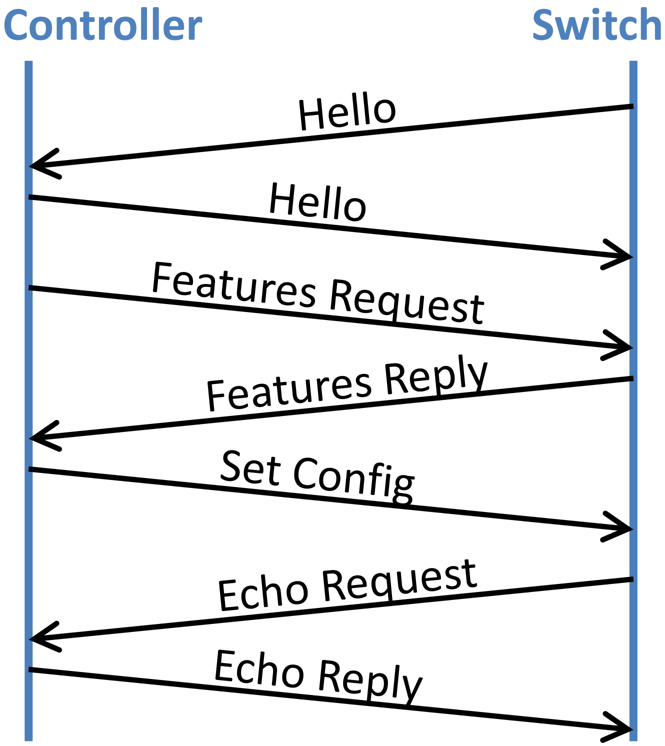
OpenFlow Messages

- Start Wireshark: # ***wireshark &***
  - Start capturing on interfaces *lo1* and s1-*
  - OF controller is on port 6633
  - Hint: filter expressions:
    - *tcp.port==6633*
    - *ip.addr==10.0.0.x*
  - Add columns for source/destination port
    - Choose “Column preferences…“ in the context menu (press right mouse button) of the existing column labels
    - Add column, then choose field type “Source port“
    - Repeat with “Destination port“

OF Controller / Startup Messages

- Start OF controller:

```
$ cd ~/ryu
$ ./bin/ryu-manager  --verbose ryu/app/simple_switch_13.py
```



<u>Connection Setup:</u>

Following the TCP handshake, the switch sends its version number to the controller.

The controller replies with its supported version number.

The controller asks to see which ports are available.

The switch replies with a list of ports, port speeds, and supported tables and actions.

The controller asks the switch to set configuration flags.

The switch checks every 5 seconds whether the controller is still alive.

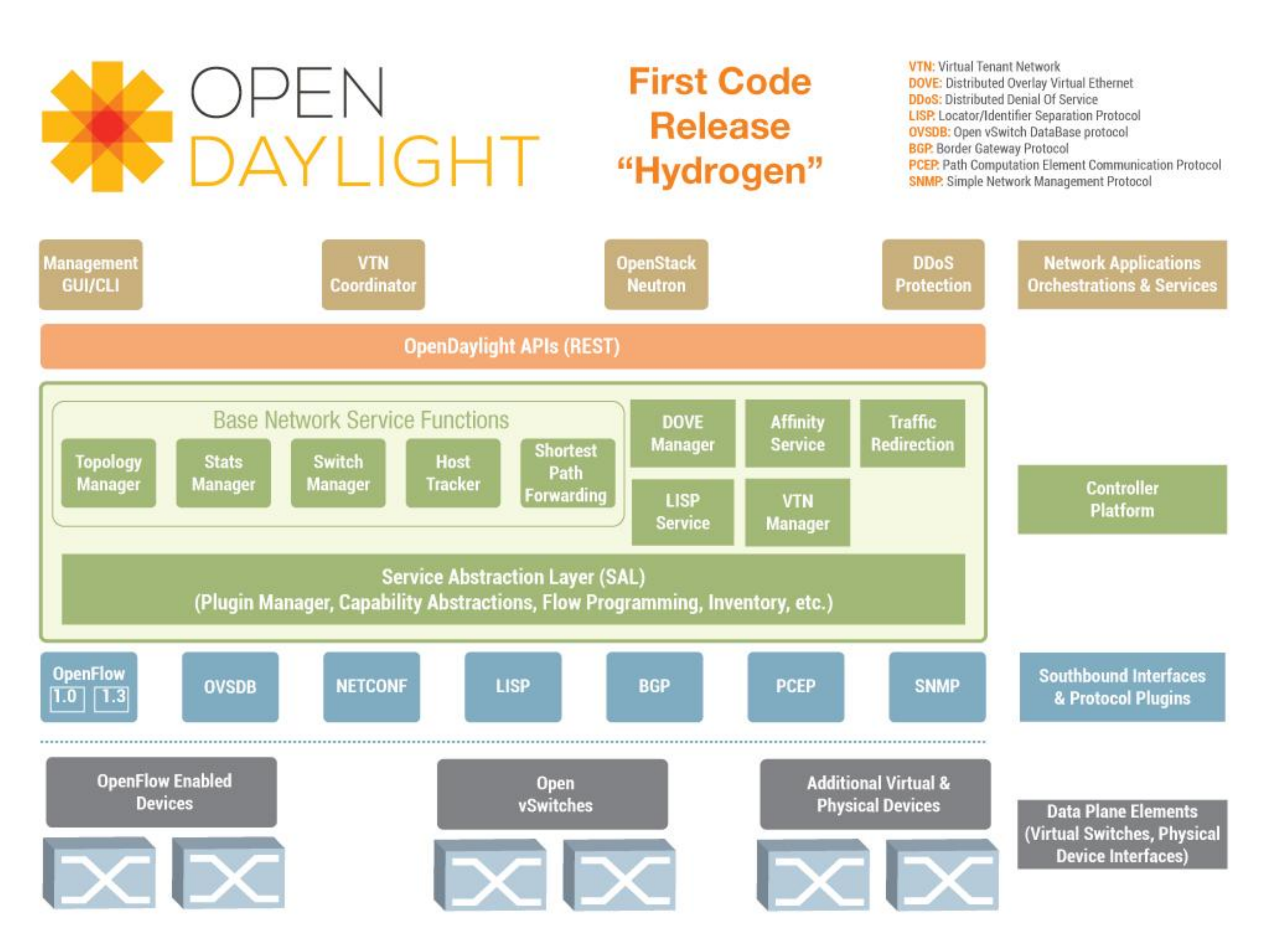
OF Messages, Examples

- Symmetric
  - Hello
  - Echo (Request/Reply)
- Controller-to-Switch
  - Features (Request/Reply): switch capabilities
  - Configuration: e.g. max new flow bytes to controller
  - Modify-State: e.g. add/delete/modify flow entries and to set switch port properties
- Asynchronous
  - Packet-In, Packet-Out
  - Port-Status: e.g. link down

OpenDaylight

- Open source project
- Platform for
  - Software-Defined-Networking
  - Network Function Virtualization
- ODL project
  - Diverse sub projects / modules, APIs
  - Java-based implementation, OSGI interface
  - Southbound interfaces, e.g. OpenFlow
- Current release: Hydrogen – 3 editions: Base, Virtualization, Service Provider

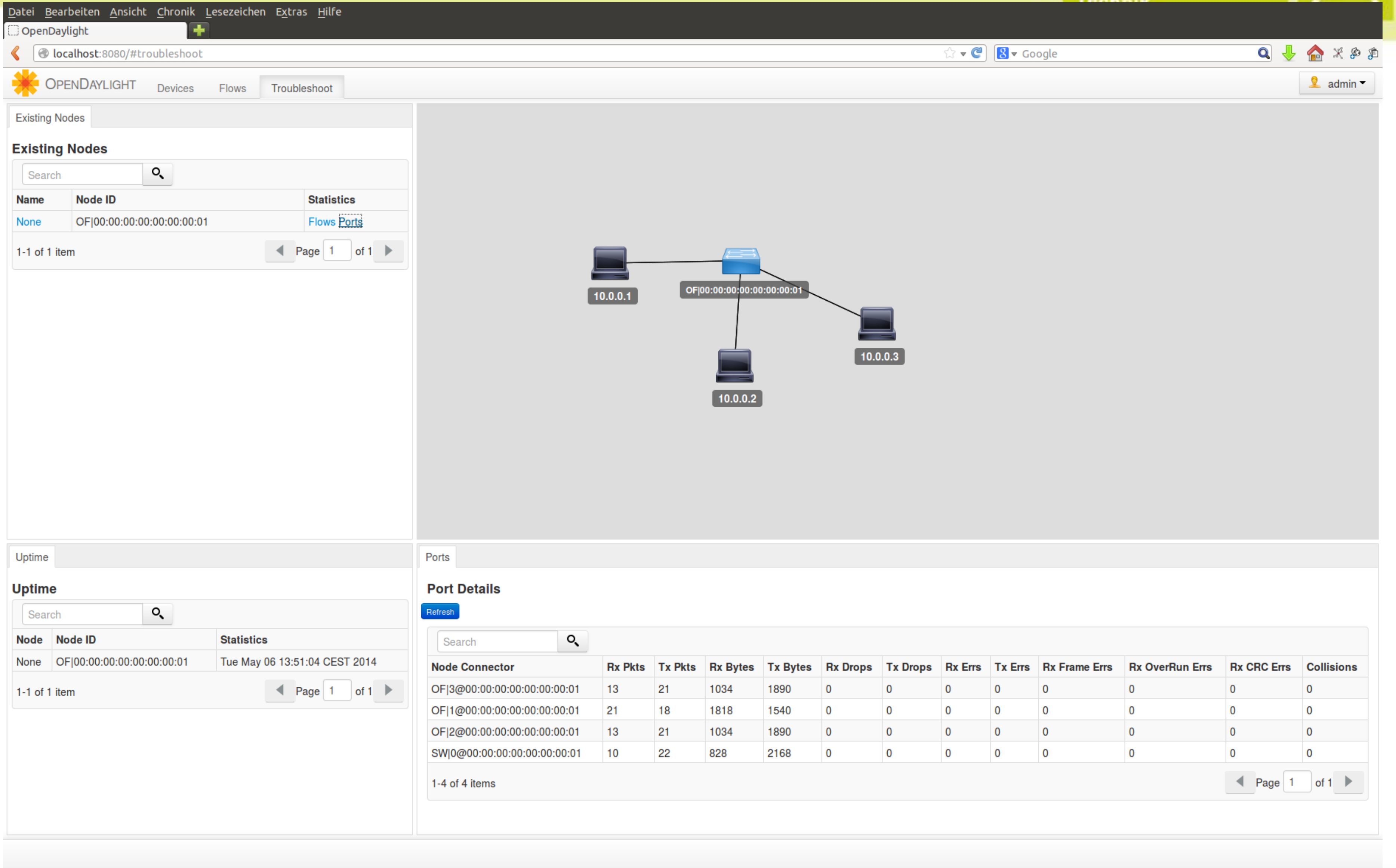
OpenDaylight



OPEN DAYLIGHT
First Code Release "Hydrogen"
Management GUI/CLI
VTN Coordinator
OpenStack Neutron
DDoS Protection
Network Applications Orchestrations & Services
OpenDaylight APIs (REST)
Base Network Service Functions
Topology Manager
Stats Manager
Switch Manager
Host Tracker
Shortest Path Forwarding
DOVE Manager
Affinity Service
Traffic Redirection
LISP Service
VTN Manager
Controller Platform
Service Abstraction Layer (SAL)
(Plugin Manager, Capability Abstractions, Flow Programming, Inventory, etc.)
OpenFlow 1.0 1.3
OVSDB
NETCONF
LISP
BGP
PCEP
SNMP
Southbound Interfaces & Protocol Plugins
OpenFlow Enabled Devices
Open vSwitches
Additional Virtual & Physical Devices
Data Plane Elements (Virtual Switches, Physical Device Interfaces)

VTN: Virtual Tenant Network
DOVE: Distributed Overlay Virtual Ethernet
DDoS: Distributed Denial Of Service
LISP: Locator/Identifier Separation Protocol
OVSDB: Open vSwitch DataBase protocol
BGP: Border Gateway Protocol
PCEP: Path Computation Element Communication Protocol
SNMP: Simple Network Management Protocol

OpenDaylight GUI



Existing Nodes

Name	Node ID	Statistics
None	OF|00:00:00:00:00:00:00:01	Flows Ports

1-1 of 1 item   Page 1 of 1

Uptime

Node	Node ID	Statistics
None	OF|00:00:00:00:00:00:00:01	Tue May 06 13:51:04 CEST 2014

1-1 of 1 item   Page 1 of 1

Port Details

Node Connector	Rx Pkts	Tx Pkts	Rx Bytes	Tx Bytes	Rx Drops	Tx Drops	Rx Errs	Tx Errs	Rx Frame Errs	Rx OverRun Errs	Rx CRC Errs	Collisions
OF|3@00:00:00:00:00:00:00:01	13	21	1034	1890	0	0	0	0	0	0	0	0
OF|1@00:00:00:00:00:00:00:01	21	18	1818	1540	0	0	0	0	0	0	0	0
OF|2@00:00:00:00:00:00:00:01	13	21	1034	1890	0	0	0	0	0	0	0	0
SW|0@00:00:00:00:00:00:00:01	10	22	828	2168	0	0	0	0	0	0	0	0

1-4 of 4 items   Page 1 of 1

- References
  - *http://www.openflow.org/wk/index.php/OpenFlow_Tutorial*
  - *http://www.mininet.org*
  - *http://www.openvswitch.org*
  - *http://www.opendaylight.org*
  - *http://sdnhub.org/tutorials/*