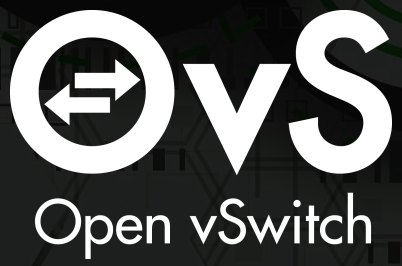


2016 FALL CONFERENCE

FAUCET -

The open source SDN Control Plane for production networks

Shivaram Mysore
Shivaram@TrustStix.com

2

Faucet

- What?
  - Drop in **replacement for L2/L3 switch with extra SDN** based functionality
  - Developed as an application for Ryu SDN Controller
  - Written in **Python** with **Apache 2 License**
- Whom?
  - **Enterprise & Campus segments** - primary focus
  - **Personas**:
    1. Network Operator - Regular Linux sysadmins, no special SDN controller ninja skills required
    2. Security Team - Network & Application Security teams
    3. Operations Manager - high level network ops & manages network Ops
    4. Business Users - need for Operational stats
    5. Application Developers - Python Developers
- Why?
  - SDN enabled switches provide numerous advantages for network operators

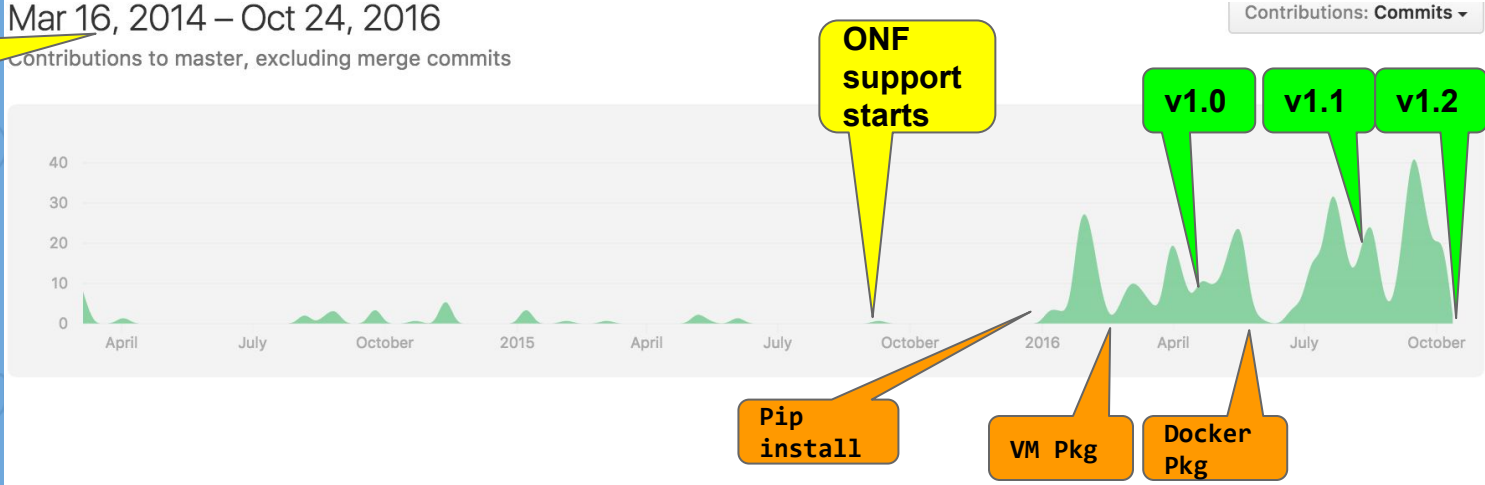
3

Faucet Differentiation

- Ease of **installation**: < 30min, drop in replacement
- Faster **upgrades** than non SDN (can upgrade controller in <1sec while network still runs and without rebooting the hardware) → Important with increasing number of **zero day attacks**.
- **Built-in support for Network Operations**
  - Much **easier to automate and integrate configuration** (you write a YAML file under Linux - no more expect scripts).
  - **Real-time database** integration for **stats** → Grafana dashboards
  - **NoSQL database** integration for **flows**
- **Greater control** of layer 2 than non-SDN (eliminating unicast flooding, defeating rogue DHCP servers, broadcast storms, etc).
- **Applications +**
- **Built-in Test suite (Mininet + Hardware)**

4

History



Core Team:

- Josh Bailey
- Brad Cowie
- Chris Lorier
- Shivaram Mysore

THE UNIVERSITY OF WAIKATO
Te Whare Wānanga o Waikato

Google

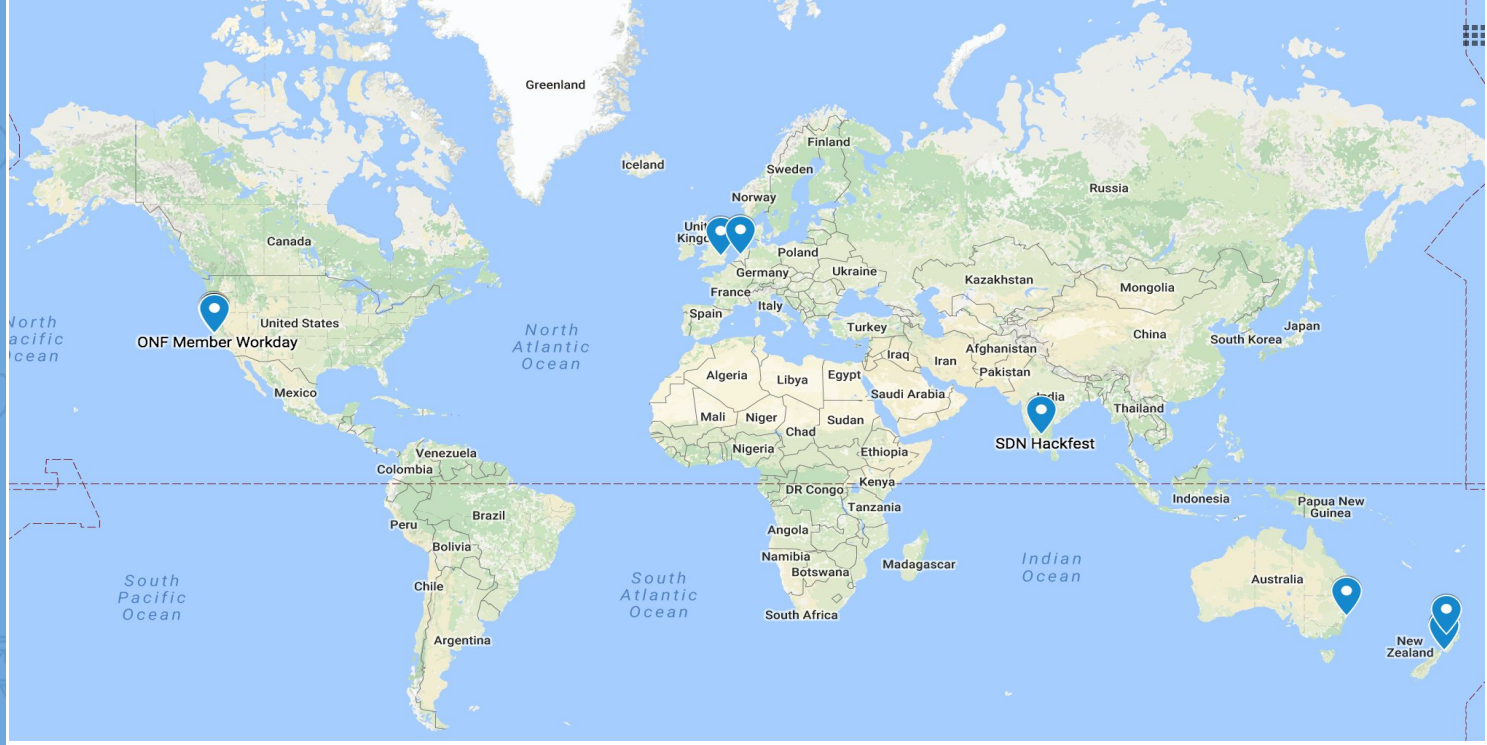
REANNZ

OPEN NETWORKING FOUNDATION

faucet-dev@openflowsdn.org; faucet-users@lists.geant.org

5

Worldwide Deployment - October 2016



SITES: ONF, REANNZ(**2 years**!), AARNet, ESNet, GEANT, GEANT HQ, Victoria University of Wellington, Allied Telesis, WAND Group Waikato University

EVENTS: SDN Hackfest, ONF Member Workday

`https://www.google.com/maps/d/u/0/viewer?mid=1MZ0M9ZtZOp2yHWS0S-BQH0d3e4s&hl=en`

6

Production Quality Code

- 5,200+ lines of code + documentation
- 2,600+ lines of test code
  - Mininet & hardware support
- 40+ devs contributed code
- Language: python
- Delivery:
  - Python pip install
  - Virtual appliance - VMDK, OVF, ISO
  - Docker package
- Dataplane support

7

Design Philosophy - 1/6

- Vendor Agnostic support
  - Openflow 1.3 with Multiple Tables & Group Table support required.
  - Minimize packet_ins (ex. host learning)
- Division of Labor architecture
  - **Faucet** - Maintain switch connectivity, pipeline and flow table management, packet_ins
  - **Gauge** - Read and store flows (NoSQL) and stats info (Time Series) for north-bound applications
- Complete control-plane upgrade
  - … while the network is running
  - … in a fraction of a second
  - “`service faucet restart`”

8

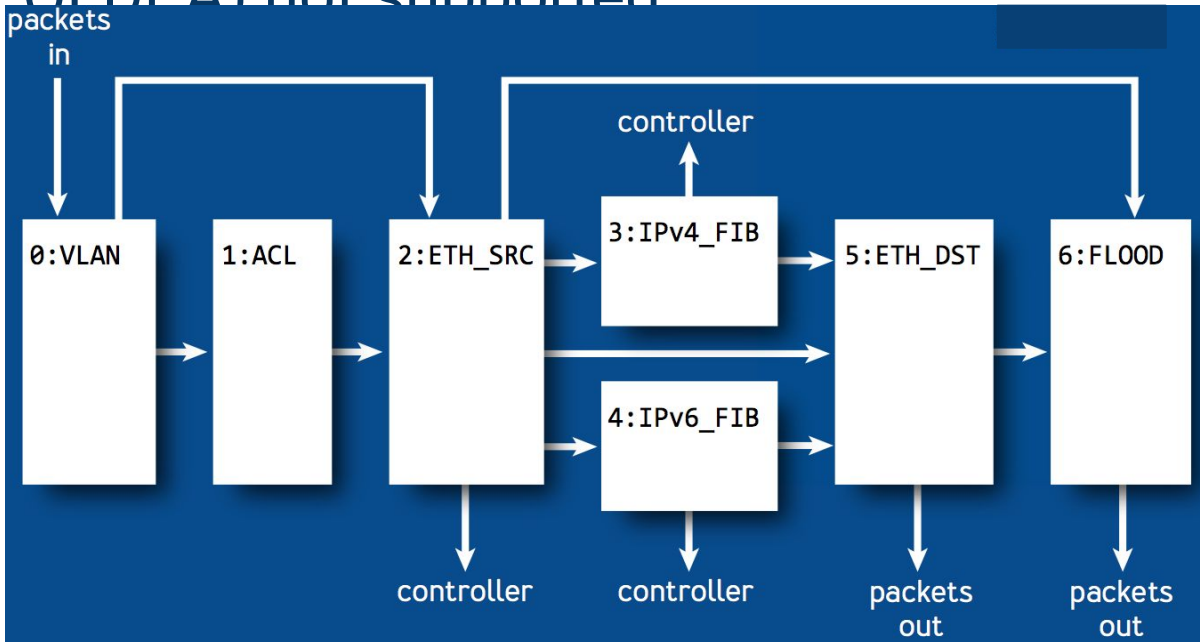
Design Philosophy - 2/6

- API
  - No external API for Add/Delete flows. Native applications have full access.
  - Program to NoSQL (CouchDB), Time Series (Influx) DB, Grafana, APIs
- “Push on Green”
  - Built-in unit test framework for Mininet & Hardware
- SDN Configurability
  - Learning - ex. Unicast flooding
  - Routing algorithms
  - ACLs, Policy Based Forwarding (PBF) based on OpenFlow matches
  - Stacking of Switches (Fabric)

9

Design Philosophy - 3/6

- Pipeline
  - OFA pipeline built on Faucet and pushed to Switches
  - Vendor specific pipelines (ex. Broadcom OFDPA) not supported



10

Design Philosophy - 4/6

Control Plane Security

- Switch-controller is connected by a dedicated port from the switch. Channel is secured by TLS or Certificates or 802.1AE MACSec
- On connection to Switch, Faucet programs “default-deny” rules - drop all unknown traffic
- Flows periodically timeout and are refreshed by the controller
- Switch can be configured for
  - “**fail-secure**” - keep forwarding and using currently programmed flows until they expire (default - expected mode for Faucet)
  - “**fail-standalone**” - revert to being a non-programmable switch
- Faucet implements expiry times on all flows. Forwarding will cease if no controller can be reached for a configurable period of time.

11

Design Philosophy - 5/6

High Availability via Idempotency

- Configure 2+ Faucet instances with the same config for the same switch (fabric)
- No inter-controller configuration or communication required

Scalability

- Faucet eschews PACKET_IN, controller scaling decoupled from switch fabric scaling
- @SDN Hackfest, with AT x930, we had 150+ hosts and 24,000+ flows
- Faucet v1.2 controls fabric of switches, and programs intra-switch data plane

Simple, Declarative Configuration

- No information is specified twice
- Nothing is specified that can be derived
- Reduces opportunities for error

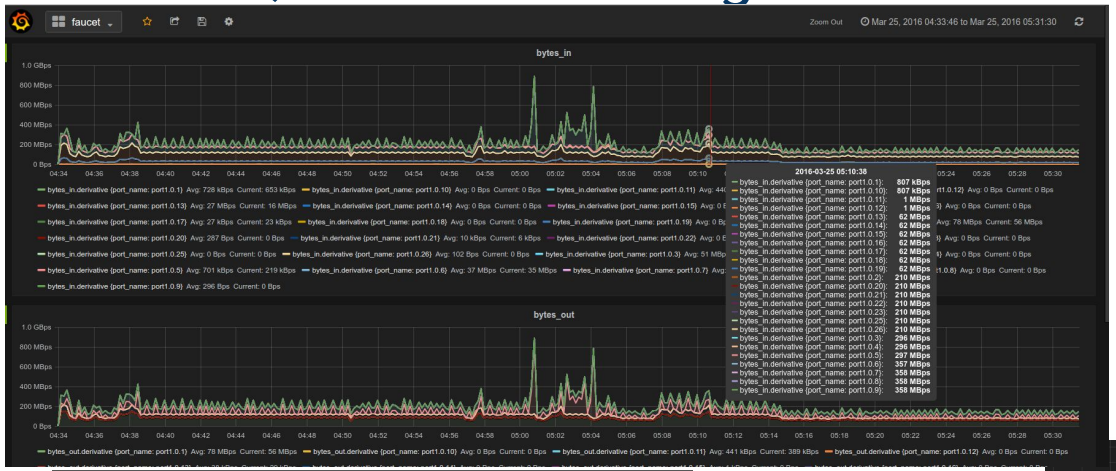
Data Plane for NFV

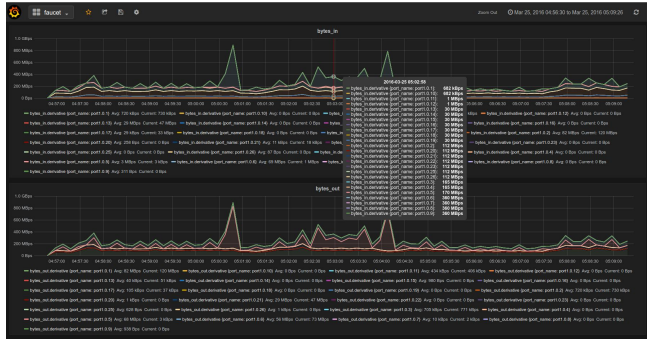
- Faucet's minimal PACKET_IN is headed to 0
- Virtualized functions like 802.1X, DHCP, NAT, IDS, etc. use parallel data path to controller
- NFVs can rewrite FAUCET configuration and apply changes in fraction of a second
  - Dynamic segmentation based on 802.1X

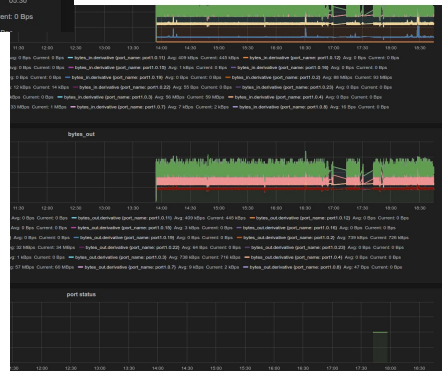
13

Network Operations & Analytics

- Real-time database integration for stats
- Grafana dashboards
- NoSQL database integration for flows

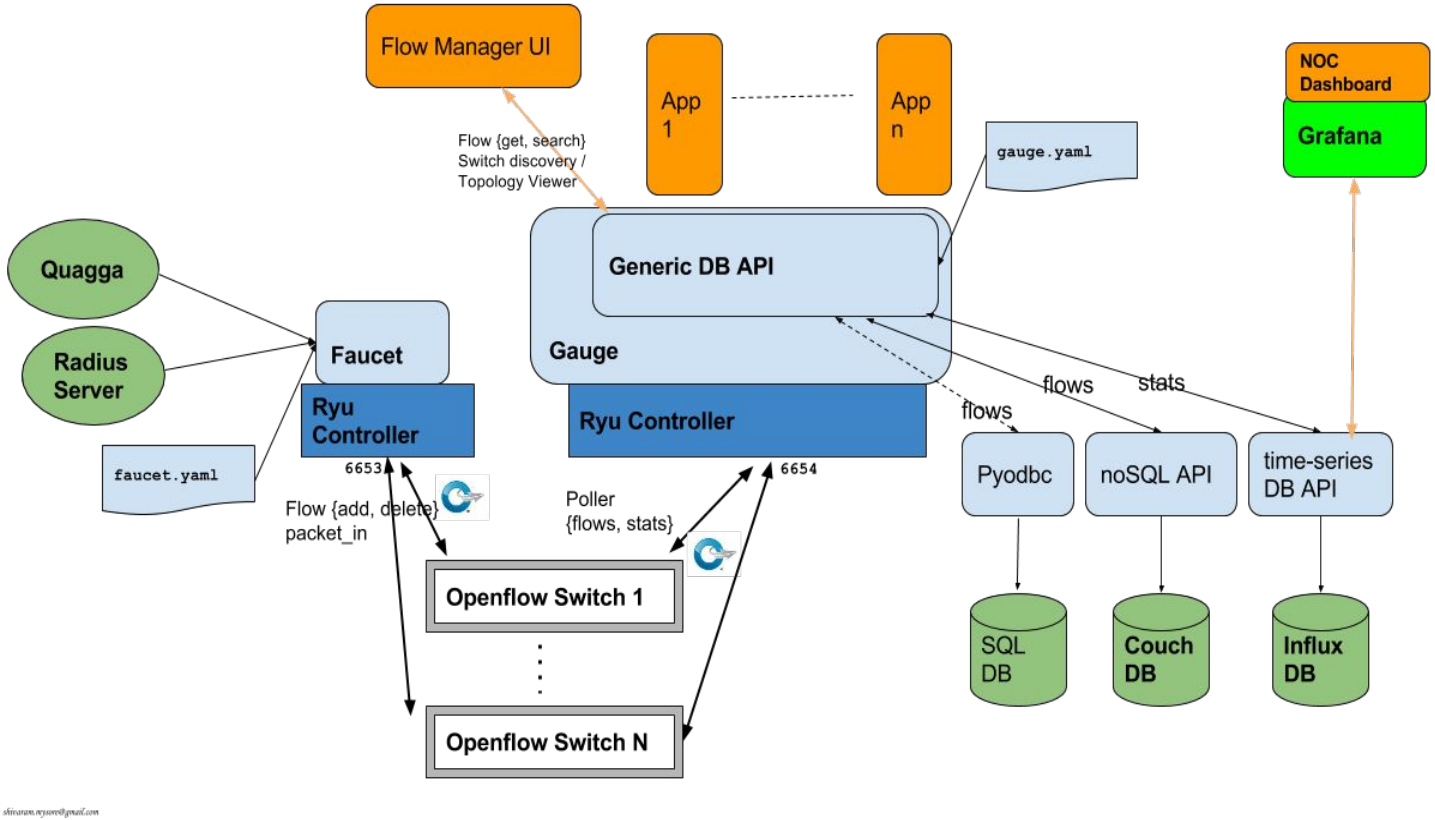






14

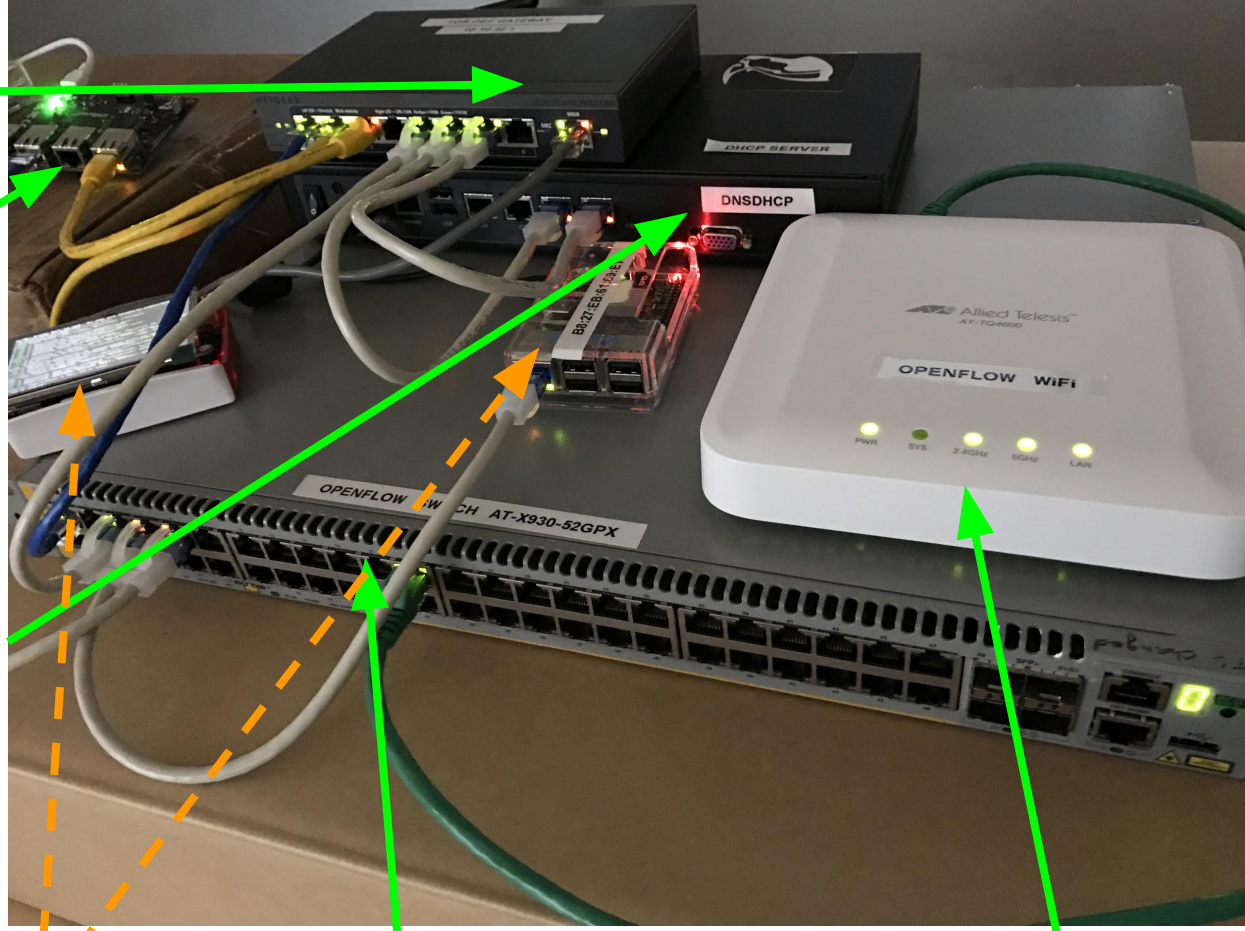
Faucet Architecture



@author: shivaram.mysore@gmail.com

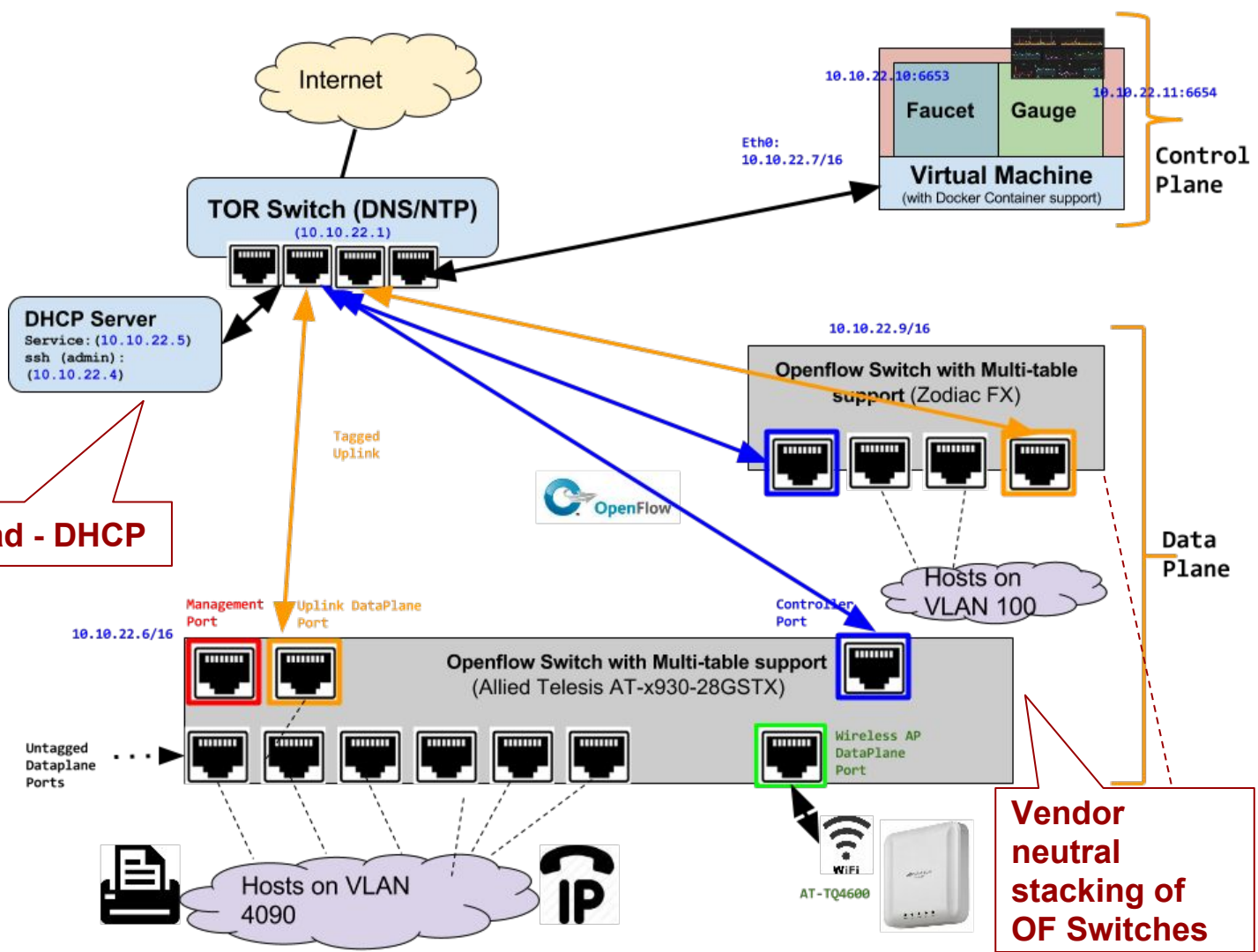
15

Deployment



16

Network Overlay Deployment Diagram



NFV Offload - DHCP

Vendor neutral stacking of OF Switches

17

Developer Profile

Python Programmer

- Knowledge of networking +
- Knowledge of NoSQL, Time Series database integration +
- Understanding of network deployments +

Standard software development practices

- Git, writing test cases, documentation
- Linux usage, network configuration, Mininet
- Install and config - virtual machine, docker

Advanced skillset (for specific projects)

- Technologies: BGP (Quagga, exabgp, Bird), OpenVSwitch, Openflow Spec, Radius, 802.1x, DHCP, DNS, Ethernet frames(L2), IP (L3), MPLS, Access Control List, Firewall,
- Software: HTML5, JavaScript, Grafana
- Deploying & configuring Cisco switches, enterprise/ campus networks

18

References

- Github Repo - https://github.com/REANNZ/faucet
- Installation
  - Python pip - https://pypi.python.org/pypi/ryu-faucet
  - Docker - https://hub.docker.com/r/faucet/
  - Virtual Machine - https://susestudio.com/a/ENQFFD/ryu-faucet
- YouTube - https://www.youtube.com/playlist?list=PL2co5JVVb0LC2rz_Ygyk8OTAnWQCGnh_8
- Blog - https://faucet-sdn.blogspot.com/
- Publications: ACM Queue (Sept/Oct 2016) - Faucet: Deploying SDN in the Enterprise
- Excellent Faucet related tutorial articles on https://inside-openflow.com/

19

Faucet SDN: Call To Action

1. Deploy Faucet SDN based wireless access (ex. Guest network) today
2. Provide us your use cases
3. Help us with code contributions (applications), sponsorships for Hackfests
4. If you are organizing an event, we can do SDN WiFi - contact me :-)

Thanks to all the vendors, developers and users for your continued support to Faucet