How I run a production, multi-vendor, enterprise Software-Defined Network

...and why you should too.

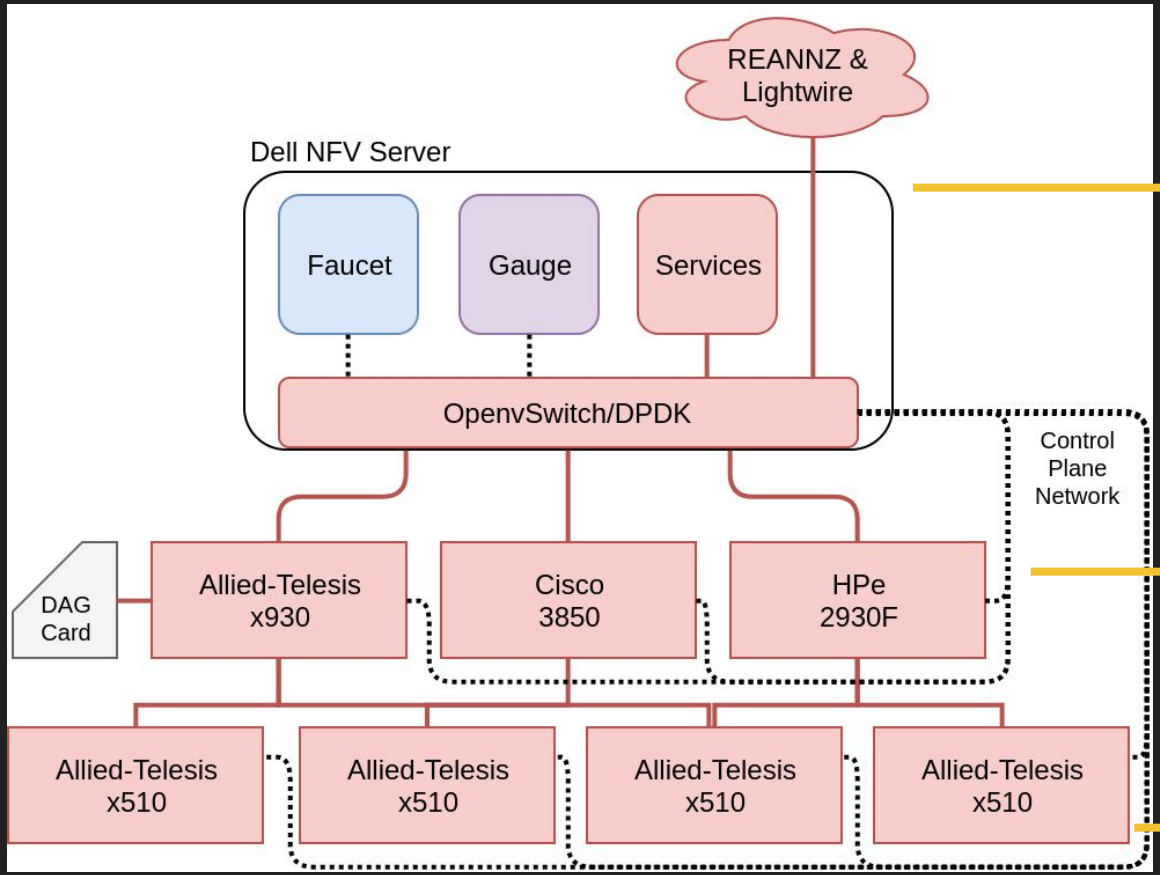
Brad Cowie
WAND Group, University of Waikato

What is FAUCET?

- Open source, well tested, production SDN controller
- Primarily developed in New Zealand
- Programs dataplane via OpenFlow to forward, route and apply policy
- Interoperates with non-SDN networks using standard protocols
- Supports multiple vendors with a single pipeline
- Provides network monitoring (GAUGE)
- Installs in <30 seconds

The WAND redcables network

- We run our own network separate from University for BYOD & Research
- Original network
  - 4x Dell PowerConnect switches
  - 1x Linux router
- Configured manually via CLI
- Replace with multi-vendor SDN network controlled by FAUCET
- Benefits
  - Less administrative overhead
  - Improve monitoring & visibility over network
  - Add network policy
  - Add L2/L3 resiliency
  - Dog-food our SDN controller



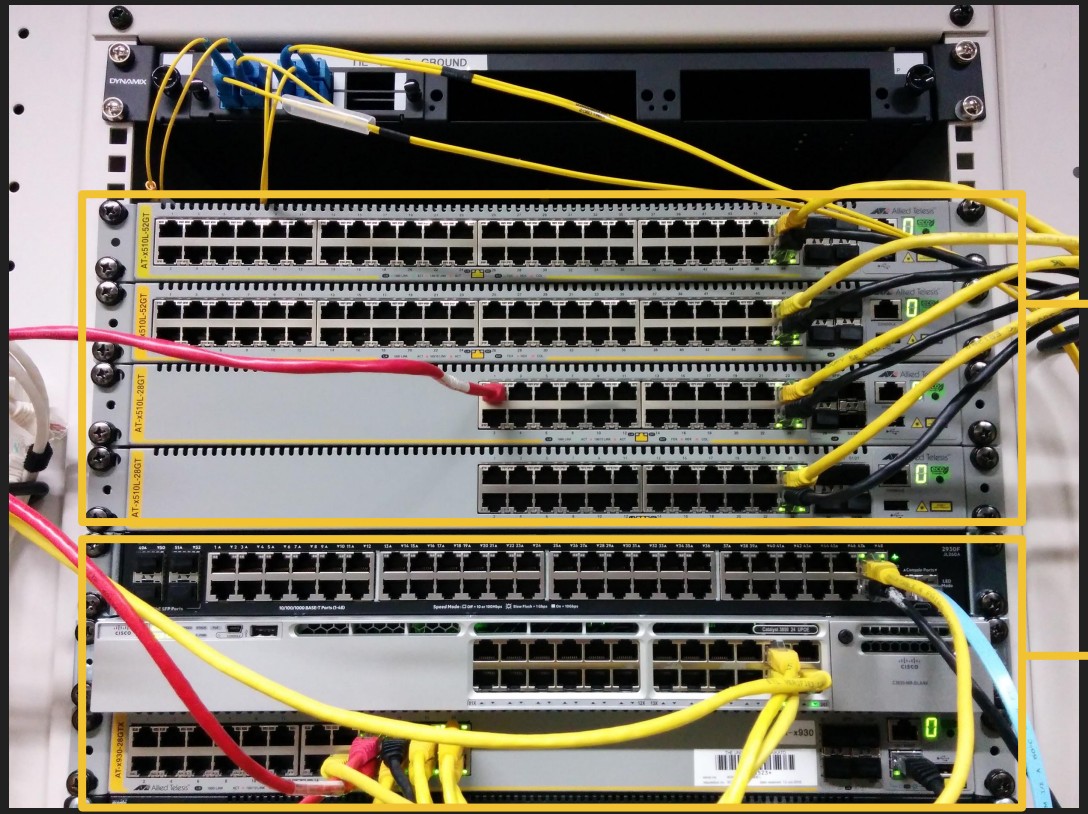
Core

- Routing
- BGP
- DHCP
- DHCPv6
- IPv6 RA
- Services

Aggregation

- Forwarding
- Policy

Access



248 OpenFlow ports

Network config management with ansible

Configure every network element with ansible stored in git repo

- Each commit represents a different network state
- Git makes rolling back & peer review simple
- Use different ansible inventories to separate staging from production
  - Deploy to a “staging” mininet topology first to test network functions
  - Deploy to a canary network first to validate configuration
- Do sanity checks as part of ansible playbook, only apply if things look good
- If you feel confident with your review process, push on green!
- Redcables ansible repo is open-source on GitHub

Network testing & validation

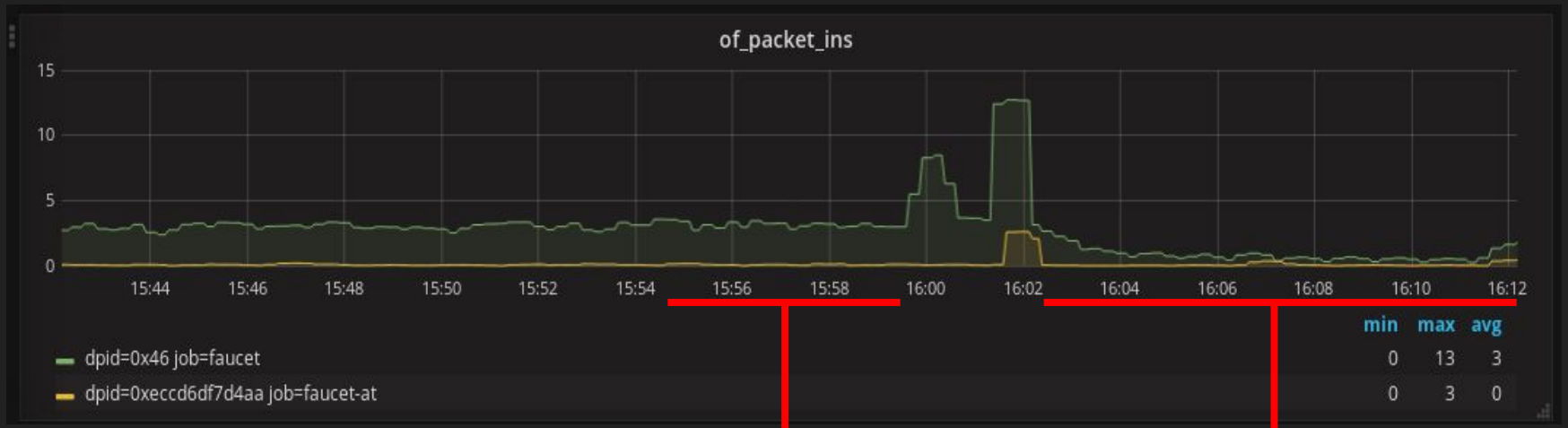
- FAUCET includes a test suite
- Test suite performs 139 different test scenarios
  - Includes real topologies
  - Includes real traffic
- All new commits to FAUCET are automatically tested
- We implement our own tests for features in use on redcables
- We can qualify new network kit with test suite to validate features
- No more attempting to parse vendor documentation
- Automate your RFP process

Network visibility with Grafana and Prometheus

Better network visibility reduces response time to faults

- Let’s not reinvent the wheel and write our own
- We use two collection/storage systems
- Prometheus (polls FAUCET)
  - MAC table, Port state, OpenFlow channel utilisation, instrumentation, etc
- InfluxDB (GAUGE polls switches for statistics and stores here)
  - Port counters (bytes in/out, packets in/out, errors)
- Grafana provides dashboards & real-time graphs of Prometheus/InfluxDB
- Prometheus also provides alerting

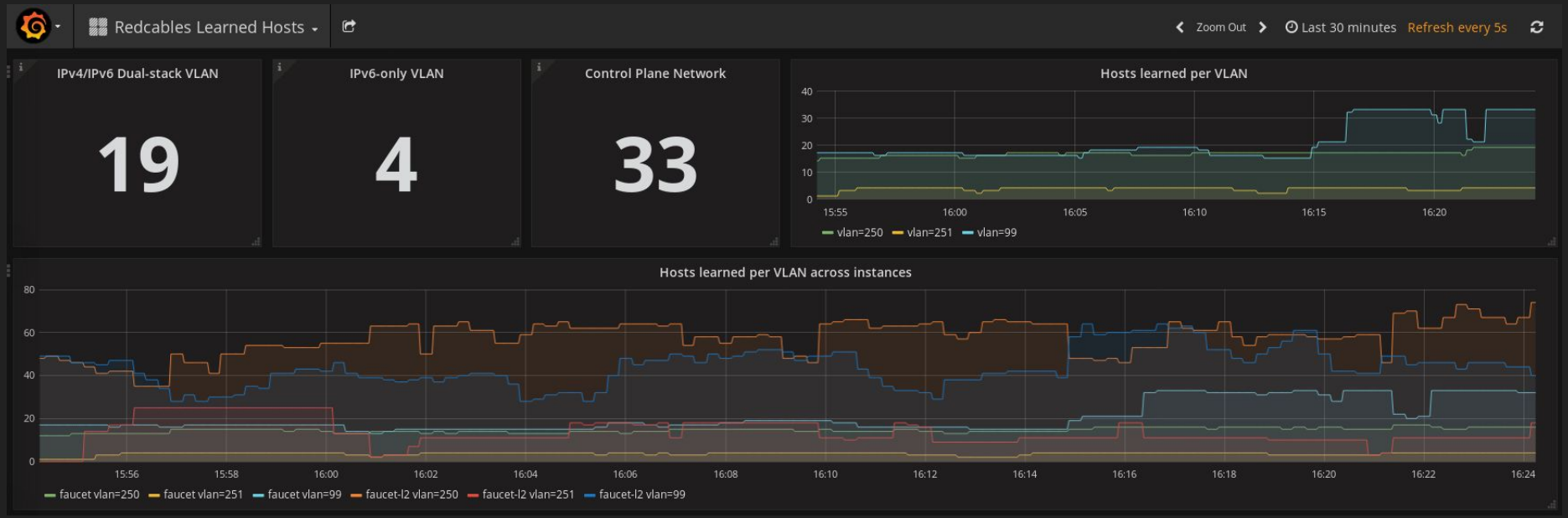
Real-time OpenFlow statistics



Old Code

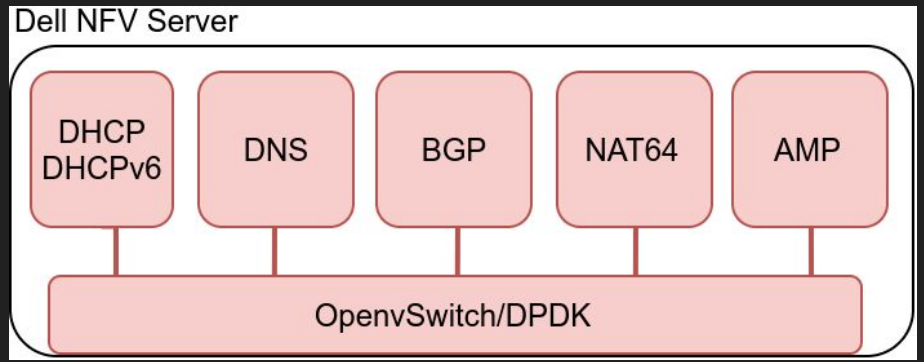
New Code

Real-time learned hosts on VLANs



Network services

- Easily deploy services on network server as VMs
  - WAND's AMP (Active network monitoring)
  - Catalyst's Are We DDoS'd Yet? (DDoS monitoring)
  - Jool NAT64 (IPv6 only network)
  - isc-dhcp-server (DHCP and DHCPv6)
  - bird (BGP)



Implement network policy with FAUCET ACLs

- A FAUCET ACL has a match and action
  - Matches anything OpenFlow can
  - Action can be DROP, ALLOW, OUTPUT, MODIFY
- Port-based ACLs
  - DHCP and DHCPv6 spoofing protection
  - IPv6 Router Advertisement Guard
  - BCP38
  - NFV offload, output 802.1x EAPOL frames to NAC
- VLAN-based ACLs
  - Drop anything that doesn’t have the IPv6 ethertype on our IPv6-only network

Example: mitigating a security vulnerability

- While I was building network there were some large security vulnerabilities
  - Intel AMT
  - WannaCry / SMB 1.0
- I was doing incident response on corporate University network for these
- Central firewall architectures only get you so far
- With FAUCET you can instantly deploy port ACLs to every port

```
- rule:
    dl_type: 0x800  # ipv4
    nw_proto: 6     # tcp
    tcp_dst: 16992  # intel-amt-http
    actions:
      allow: 0      # drop
```

Example: Intrusion Detection System

- Policy-based packet inspection
- Fine-grained IDS
- Carve packets off a (large) link and direct at Endace DAG capture card
- No longer have to inspect entire links
- Distributed packet inspection (steer packets towards nearest DAG)
- Can signal DAG card with metadata about what is being captured

```
- rule:
    dl_type: 0x800  # ipv4
    nw_proto: 6     # tcp
    actions:
      output:
        port: dag
```

Future plans

- FAUCET Foundation
- Policy based routing
- OpenFlow wireless

Thanks!

- FAUCET Developers
- Vendors
  - Allied-Telesis
  - Cisco
  - HPe/Aruba
  - OpenvSwitch
  - Endace
- REANNZ
- AARNet
- WAND

Challenge

- If this talk introduced concepts you want to see on your network, I challenge you to deploy FAUCET somewhere inside your organisation to experiment

```
$ pip install faucet
$ ryu-manager faucet.faucet
```

or

```
$ docker run -d --name faucet \
    -v /etc/ryu/faucet/:/etc/ryu/faucet/ \
    -v /var/log/ryu/faucet/:/var/log/ryu/faucet/ \
    -p 6653:6653 -p 9244:9244 \
    faucet/faucet
```

Questions?

- Our network is open source
  - https://redcables.wand.nz/

- Contact me
  - brad@waikato.ac.nz

- More about FAUCET
  - http://faucetsdn.org
  - faucet-users@lists.geant.org

Bonus slide - NZNOG

- New Zealand Network Operators' Group (NZNOG17) conference network was entirely OpenFlow controlled
- 150 network engineers on our OpenFlow WiFi
- Similar architecture to WAND network
  - Dell NFV host
  - 2x Allied Telesis x930 switches
  - 6x Ubiquiti Unifi AC access points

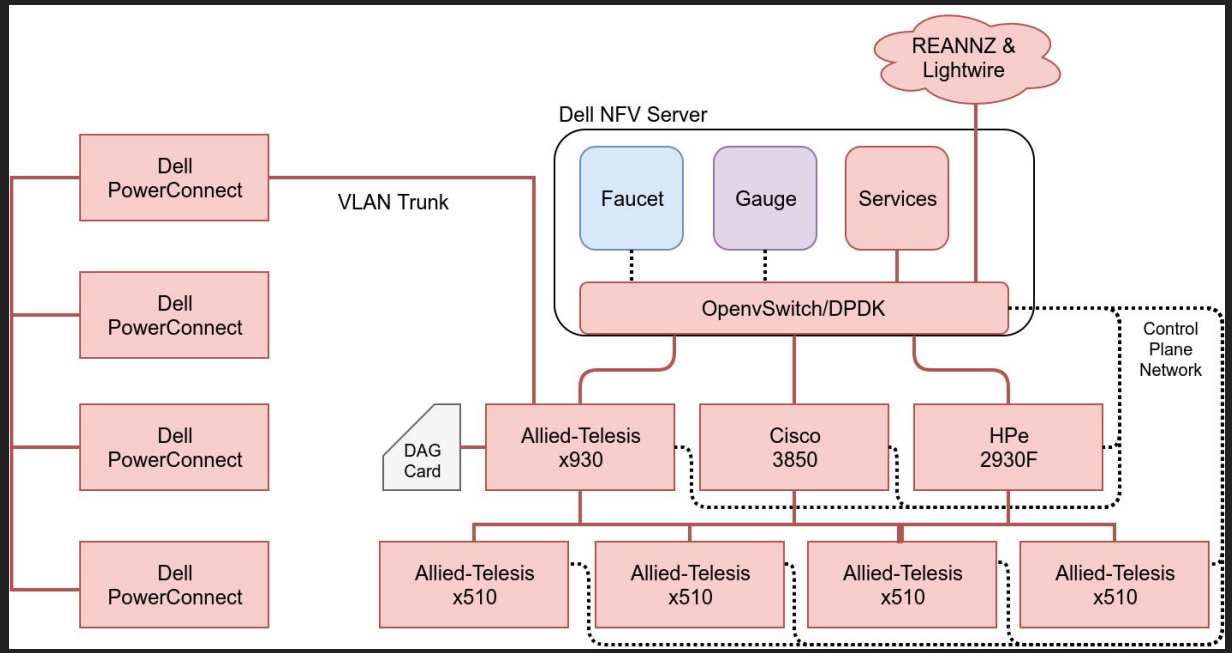
https://github.com/wandsdn/conference-sdn-nfv-network

Bonus slide - Hardware

- Dell PowerEdge R430
  - 1U, Xeon E5-2630 2.2GHz, 32GB DDR4
  - Redundant PSU
  - Intel i350-T4 1G DPDK NIC
  - Intel x710-DA4 10G DPDK NIC
- Allied-Telesis
  - 2x AT-x510L-52GT
  - 2x AT-x510L-28GT
  - 1x AT-x930-28GTX
- Cisco
  - 1x Catalyst 3850
- HPe
  - Aruba 2930F JL260A

Bonus slide - Migration

- Currently both networks are deployed side-by-side
- Share VLANs between networks via VLAN trunk



Bonus slide - Deployment experience

- What was harder than expected
  - IPv6 Stateless Addressing
  - Be careful not to overload switches when reconfiguring 248 ports at same time
  - DPDK requires a non-zero amount of tweaking
- What was easier than expected
  - Turning up BGP with REANNZ and announcing network prefixes
  - Constantly renumbering network is as easy as running `sed' over git repo
  - Debugging issues is as easy as reviewing flow table and implementing a test to cover issue

Bonus slide - What is SDN?

- SDN is a separation of control and data planes
- Dataplane (typically hardware) does high performance forwarding
- Control plane implements business logic and forwarding control
- Shift control plane from embedded CPU to high performance compute
- Separation enables the control plane:
  - To be easily tested, upgraded and scaled without dependency on data plane hardware
  - To implement same business logic on multiple vendor data planes
  - Open source, non vendor provided control plane avoids vendor lock-in
  - Can implement your own features