FaucetCon 2019
SDN Traffic Engineering for Wireless ISPs

Overview: Using Faucet to control flows and manage bandwidth in a dynamic RF environment where capacity is elastic and not static.

Kevin Myers
Senior Network Architect

Sajan Parikh
Lead Developer

About
Kevin Myers

FAUCETCON 2019

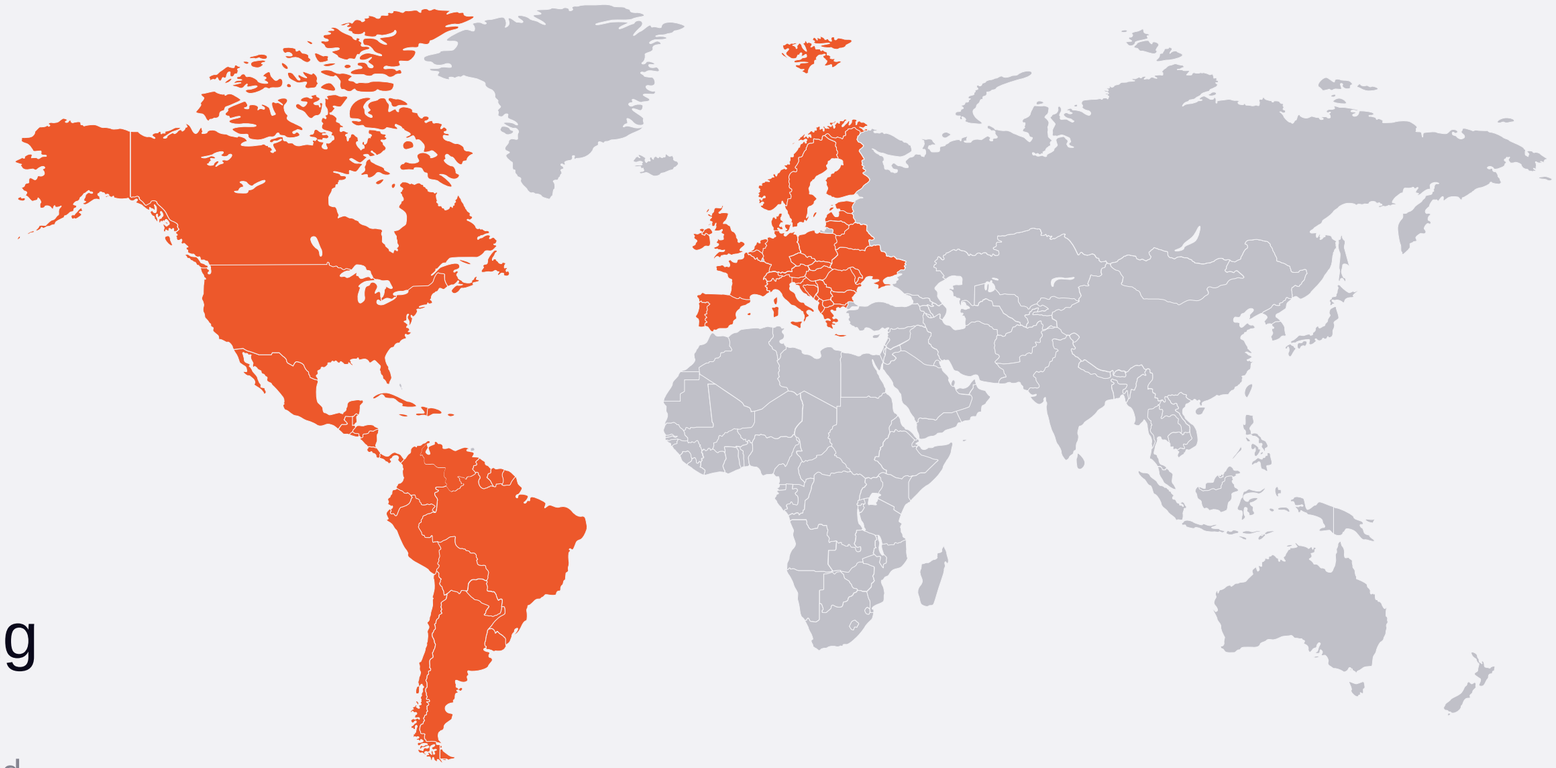
Background

- Senior Network Architect and co-founder of iparchitechs.com
- Experience in WISP, Large ISP, Fortune 500 Enterprise and Data Center
- Blog at stubarea51.net
- Twitter @stubarea51
- Contributor at thenetworkcollective.com, techfieldday.com and packetpushers.net
- **Email:** kevin.myers@iparchitechs.com

About
IP ArchiTechs Managed Services

FAUCETCON 2019

IP ArchiTechs is a global network engineering and design firm that covers a wide spectrum of environments: We provide expert consulting to Service Provider, Enterprise and Data Center network teams



Global

Offices in the US, Europe and South America

ONIE

Open Networking Focused

Whitebox, Disaggregation and Commodity networking

IP ArchiTechs MANAGED SERVICES

Overview
SDN Traffic Engineering for Wireless ISPs

FAUCETCON 2019

01 Defining the RF traffic problem

02 Lab concept

03 DrainPipe

04 Prod testing

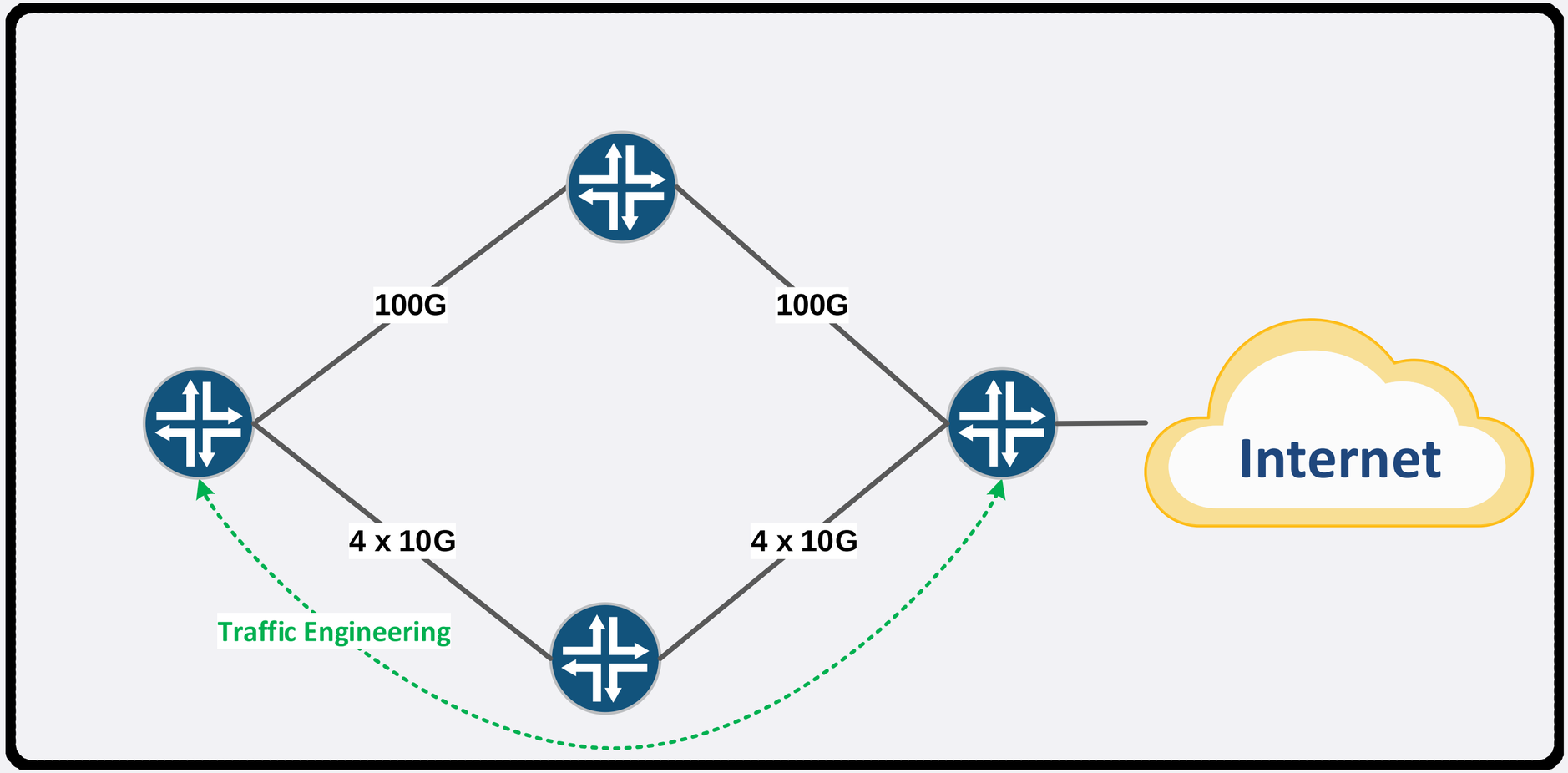
05 Next steps

FAUCETCON2019

01 Defining the RF traffic problem

SDN Traffic Engineering for Wireless ISPs
Defining the RF traffic problem

FAUCETCON2019

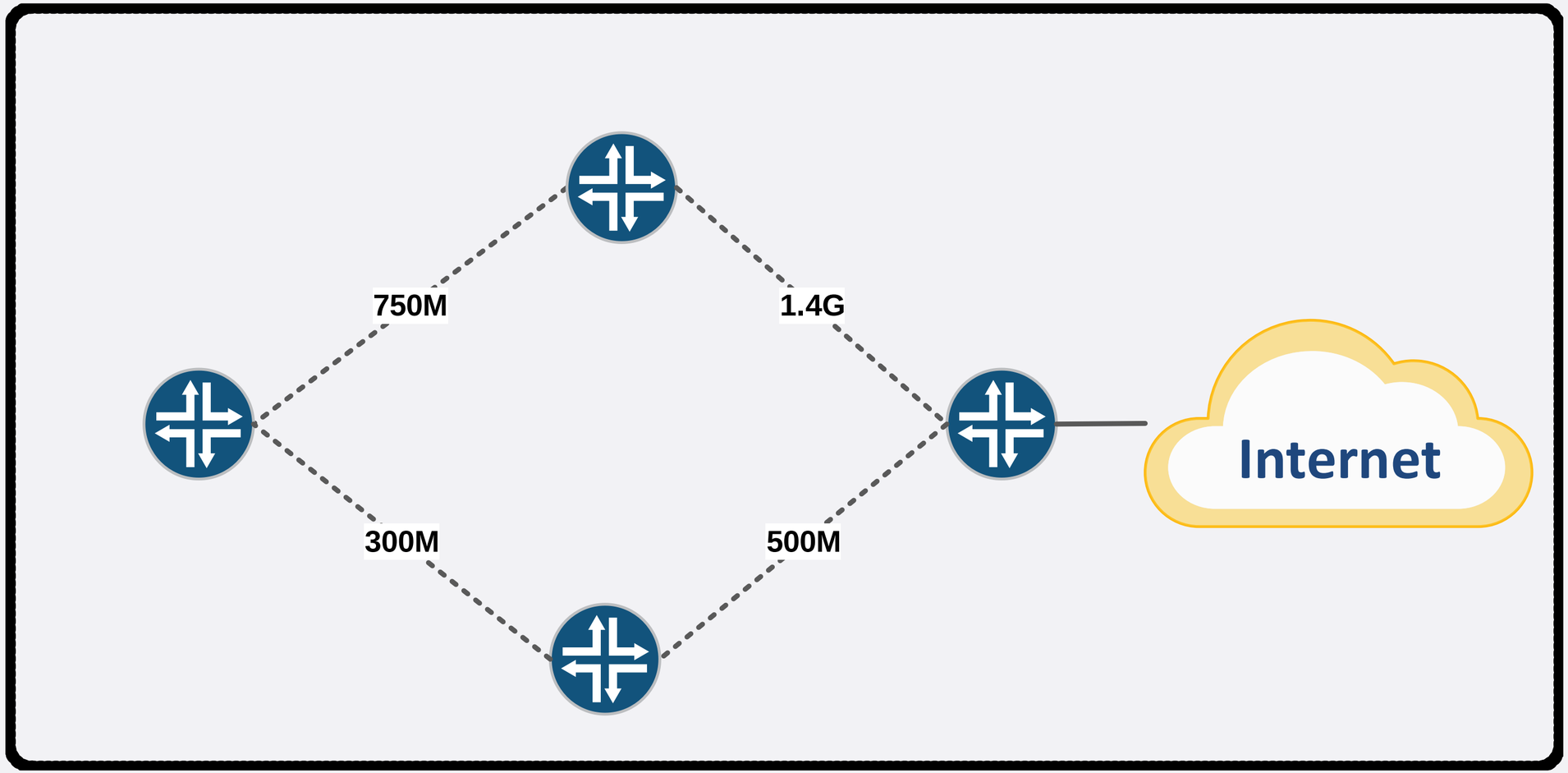


Solving the problem in a fiber ISP network

- Policy Based Routing
- MPLS TE
- Segment Routing

SDN Traffic Engineering for Wireless ISPs
Defining the RF traffic problem

FAUCETCON2019



A harder problem in a WISP

- Bandwidth is not static
- Capacity change due to weather events, atmospheric conditions, interference, etc
- Current methods limited to LTE

FAUCETCON 2019

Conventional traffic engineering has limitations

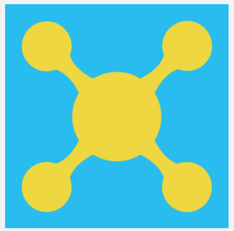
MPLS TE or Segment Routing cannot easily account for the link capacity frequently changing

FAUCETCON 2019

02 Solution proposal & lab concept

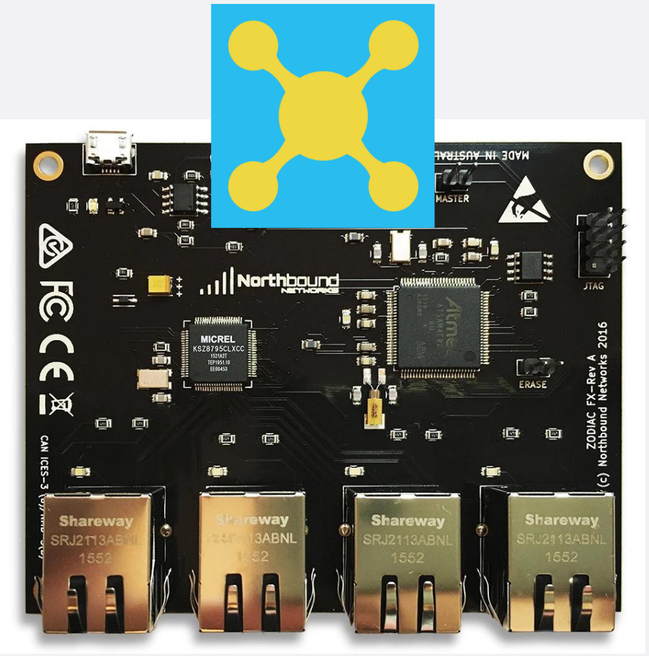
Working towards a solution

Why not use Faucet + OpenFlow to determine available bandwidth and influence traffic?



FAUCETCON2019

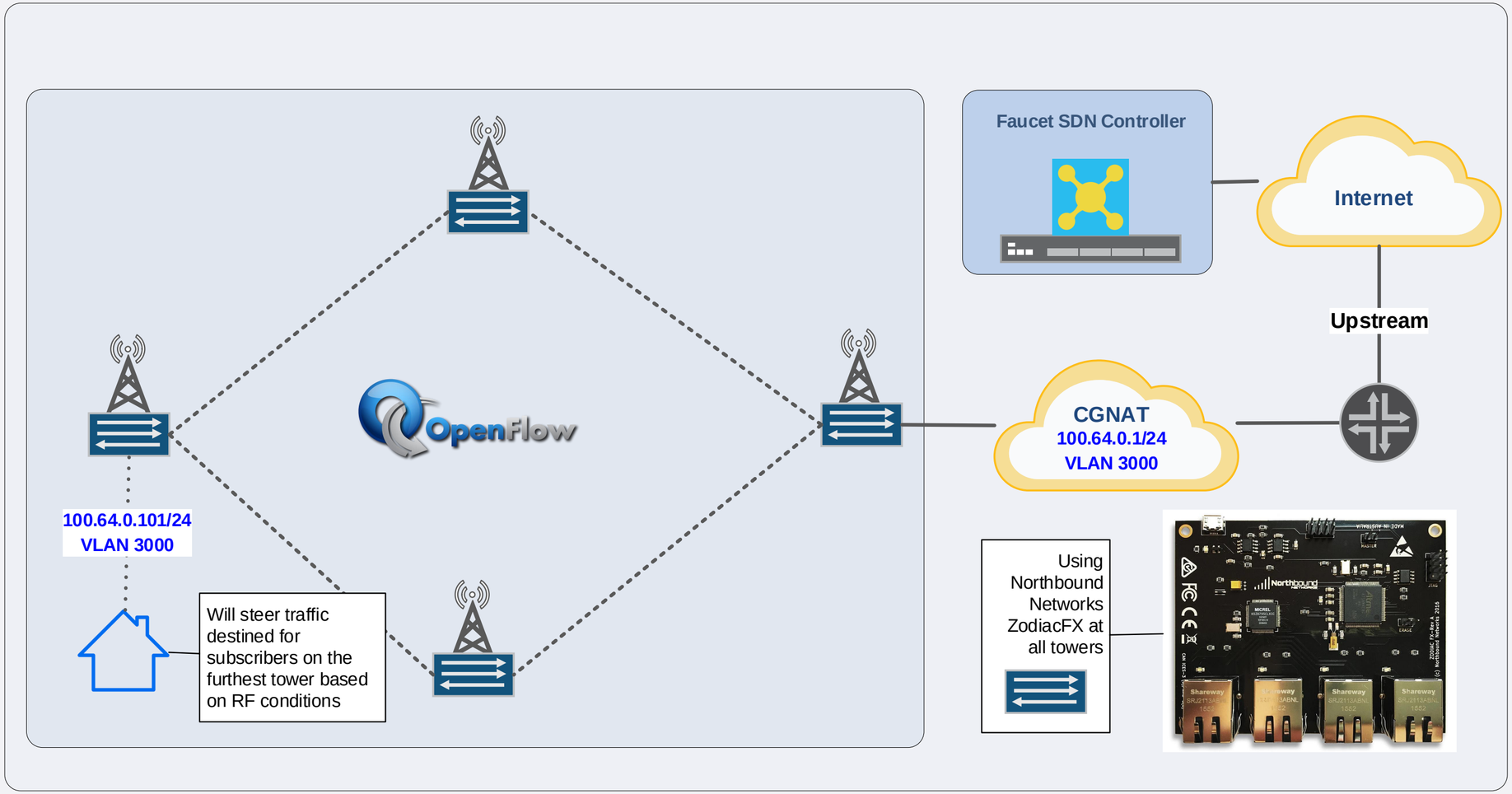
How we decided on Faucet and Zodiac FX



Needed an operational SDN controller and an inexpensive platform to perform POC testing

SDN Traffic Engineering for Wireless ISPs
Solution proposal & lab concept

FAUCETCON2019

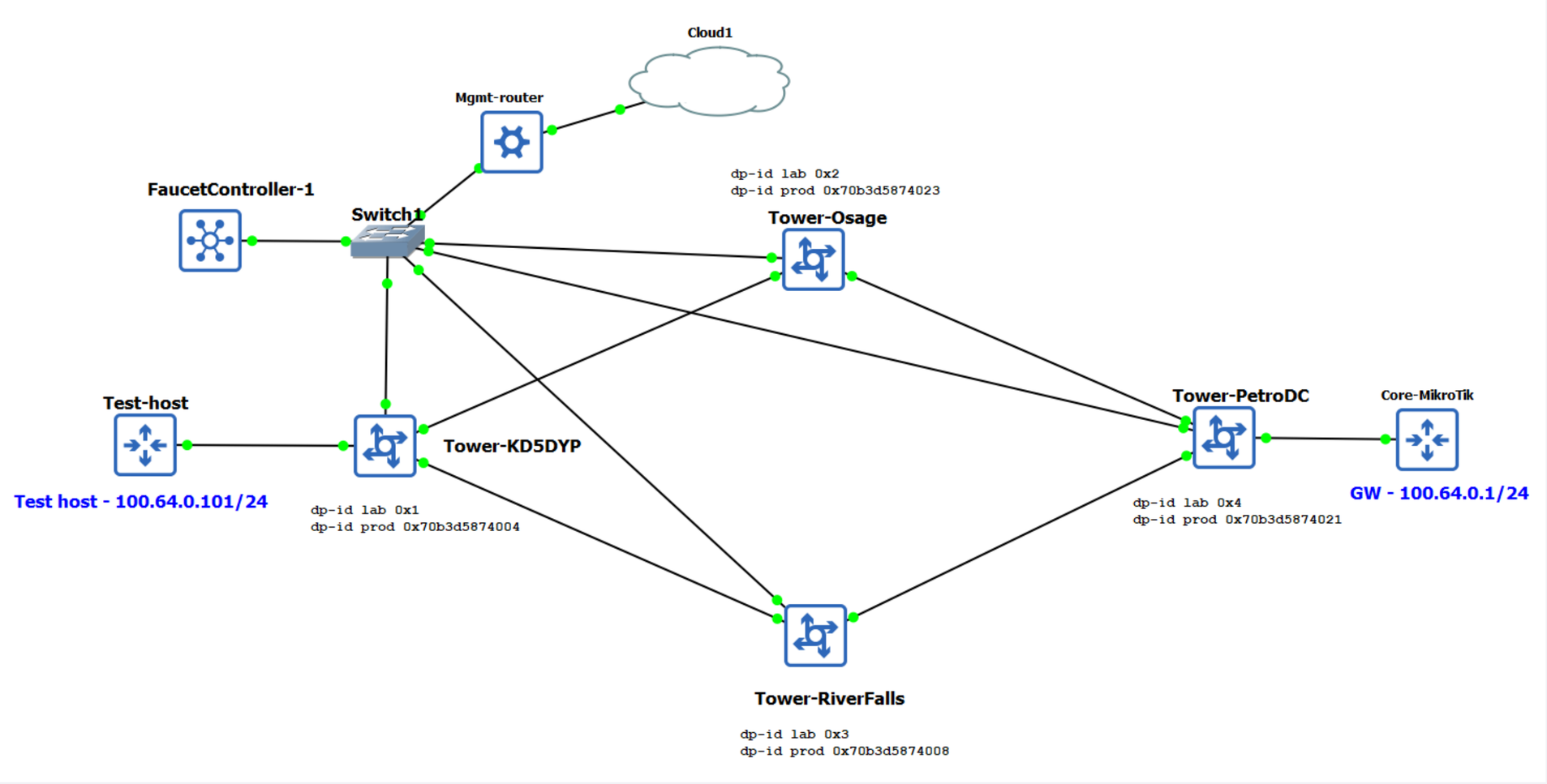


Initial concept

- L2 stack, single root
- Analyze radio capacity
- Start simple with two paths
- Move traffic to an alternate path as radio capacity changes

SDN Traffic Engineering for Wireless ISPs
Solution proposal & lab concept

FAUCETCON2019



Lab environment in GNS3

- Tried EVE-NG but had issues with OVS
- Based on future prod deployment
- Use policy-based forwarding to move traffic based on src

SDN Traffic Engineering for Wireless ISPs
Solution proposal & lab concept

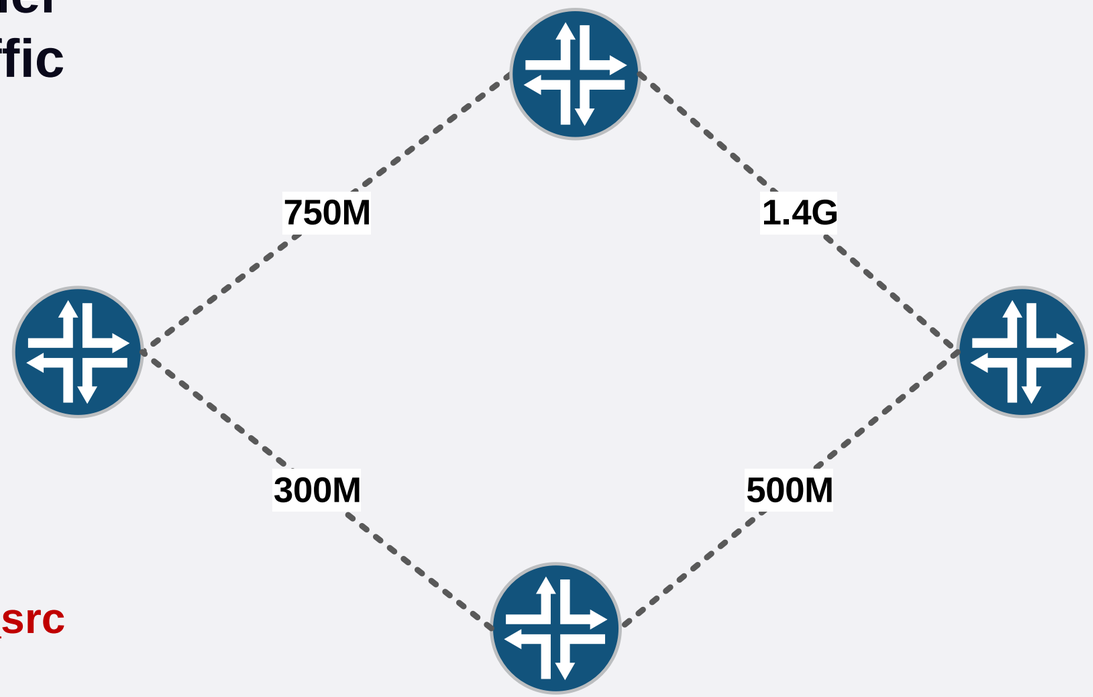
FAUCETCON2019

Break a subnet into smaller components to move traffic

100.64.0.0/24

If bandwidth reduces by 25% use a /26 to match for Faucet ACL

100.64.0.0/26
100.64.0.64/26 **<---- match ipv4_src**
100.64.0.128/26
100.64.0.192/26



Identify traffic

- Started with an easy concept
- Based on future prod deployment
- Use policy-based forwarding to move traffic based on src

SDN Traffic Engineering for Wireless ISPs
Solution proposal & lab concept

FAUCETCON2019

```
vlans:
    test-3000:
        vid: 3000
        description: "faucet test network"
acls:
    move-traffic-br0:
        - rule:
            eth_type: "0x800"
            ipv4_src: "100.64.0.101"   # Test match to move traffic
            actions:
                output:
                    port: 2            # Move traffic onto new path
                allow: True            # ALlow port traffic
    allow-all:                         # Allow all traffic
        - rule:
            actions:
                allow: True
dps:
    br0:
       dp_id: 0x1
       hardware: "Open vSwitch"
       stack:
           priority: 1
       interfaces:
           1:
               name: "eth1"
               description: "port 1"
               stack:
                   dp: "br1"
                   port: 1
```

L2 Stack (moving to L3) with ACL

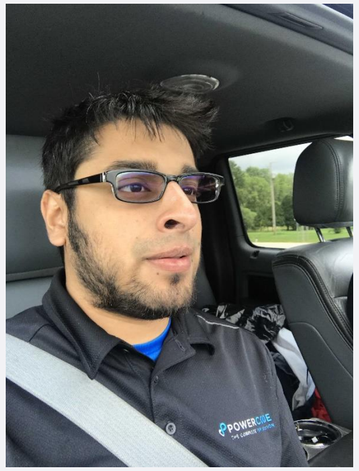
- Single root L2 stack
- ACL match on ipv4_src

FAUCETCON2019

03 Using DrainPipe to analyze RF

About
Sajan Parikh

FAUCETCON 2019

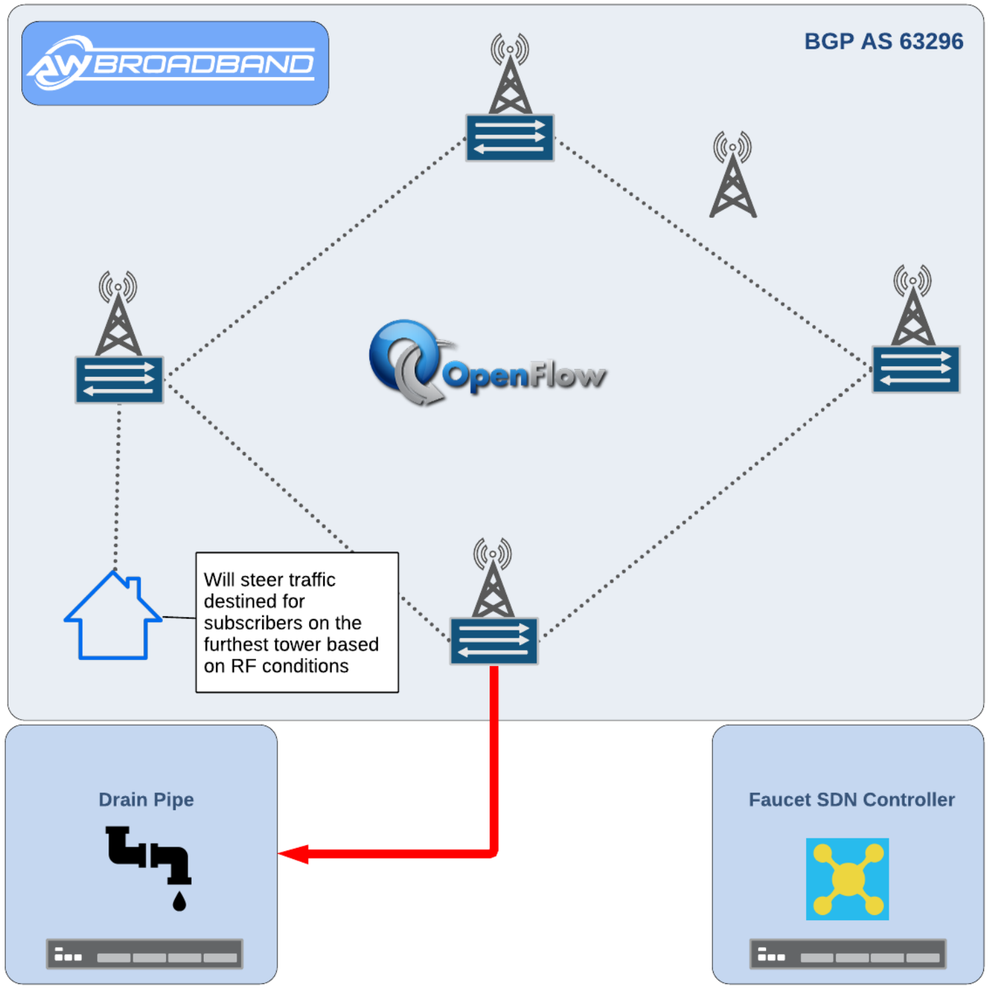


Background

- 10+ Years Software Development.
- Managed large-scale Linux deployments.
- Exposure to hundreds of WISP deployments.

- **Email:** sajan.parikh@iparchitechs.com

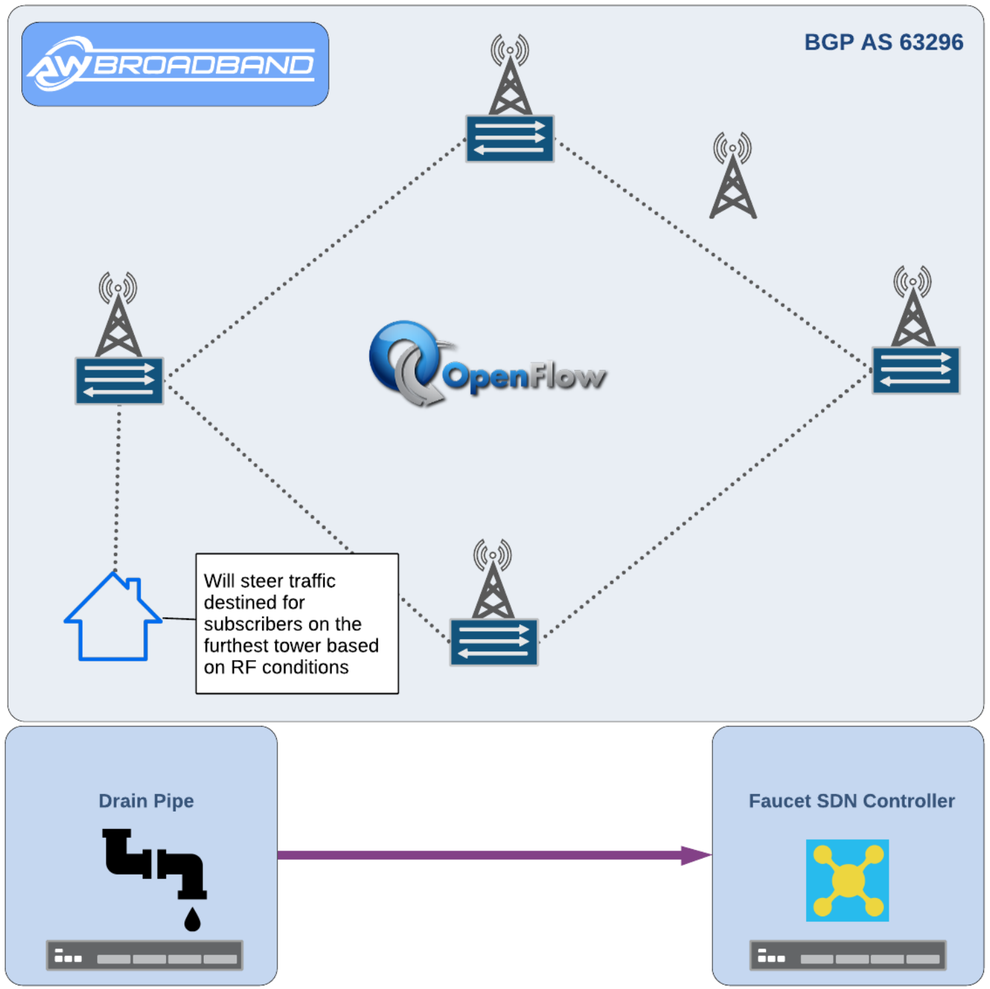
SDN Traffic Engineering for Wireless ISPs
Using DrainPipe to analyze RF



RF and Path Data

Using a variety of data sources and metrics, we continuously gather relevant data points to determine the health of RF paths.

SDN Traffic Engineering for Wireless ISPs
Using DrainPipe to analyze RF



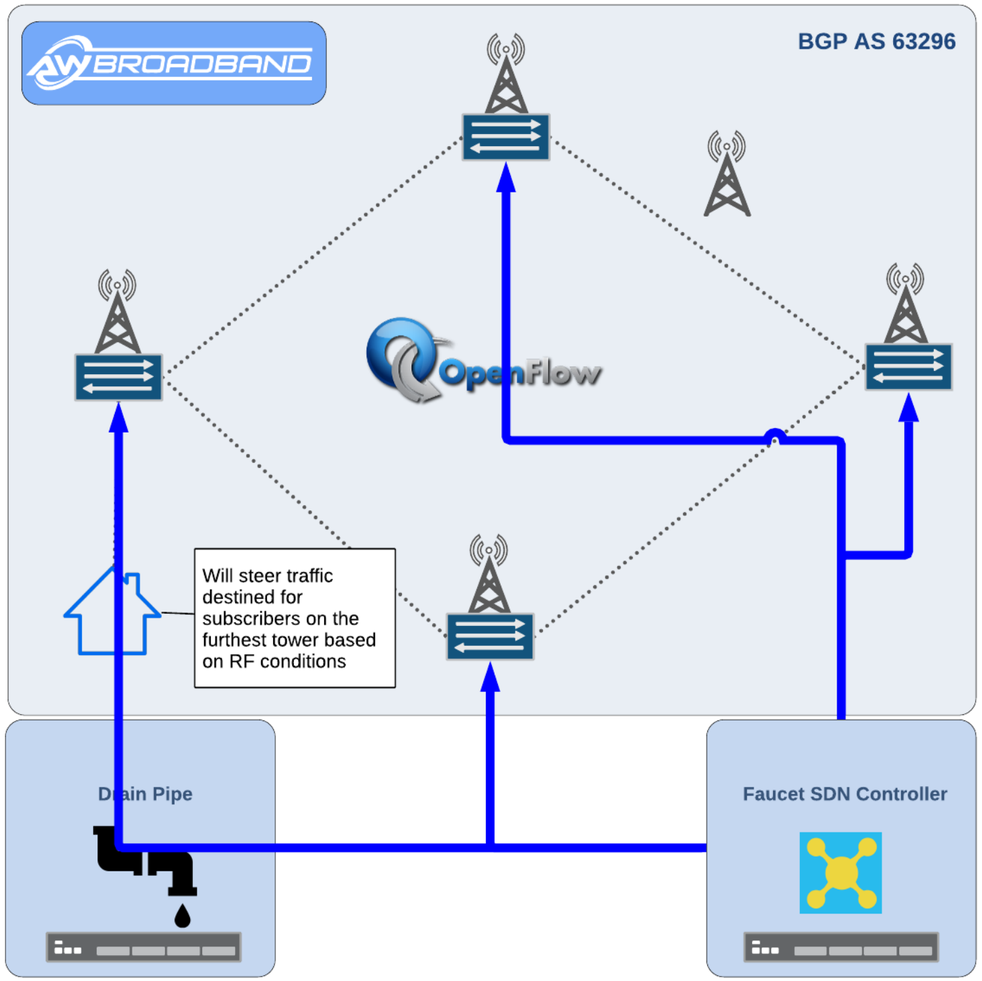
Drain Pipe Configures Faucet

Using the path health data, Drain Pipe will leverage pre-existing knowledge of the overall network architecture and topology to determine what needs to happen.

Drain Pipe will reconfigure Faucet as needed based on these changing conditions.

SDN Traffic Engineering for Wireless ISPs
Using DrainPipe to analyze RF

FAUCETCON2019



Faucet Steers Traffic In Line With WISP Business Goals

Unused bandwidth is expensive

All the Netflix!

Using the information processed by Drain Pipe, Faucet will be able to intelligently and seamlessly steer traffic through the network.

SDN Traffic Engineering for Wireless ISPs
Using DrainPipe to analyze RF

FAUCETCON2019

```
$faucet = Yaml::dump([
    'acls' => $acls,
    'vlans' => $vlans,
    'dps' => $dps
], 10);

\Storage::disk('etc-faucet')->put('faucet.yaml', $faucet);
```

SDN Traffic Engineering for Wireless ISPs
Using DrainPipe to analyze RF

FAUCETCON2019

```
sajan@faucet:~/drain-pipe$ crontab -l
* * * * * /home/sajan/drano
sajan@faucet:~/drain-pipe$
```

SDN Traffic Engineering for Wireless ISPs
Using DrainPipe to analyze RF

FAUCETCON2019

```
sajan@faucet:~/drain-pipe$ php artisan dp:plunge
Getting Capacity @ 10.172.10.140
TX Capacity: 774012160
RX Capacity:773621760
Getting Capacity @ 10.172.19.162
TX Capacity: 155648
RX Capacity:37632
Sending toward 10.172.19.162
Working through network conditionals.
Generating Faucet configuration.
Checking Faucet configuration.
Signaling HUP to Faucet
sajan@faucet:~/drain-pipe$
```

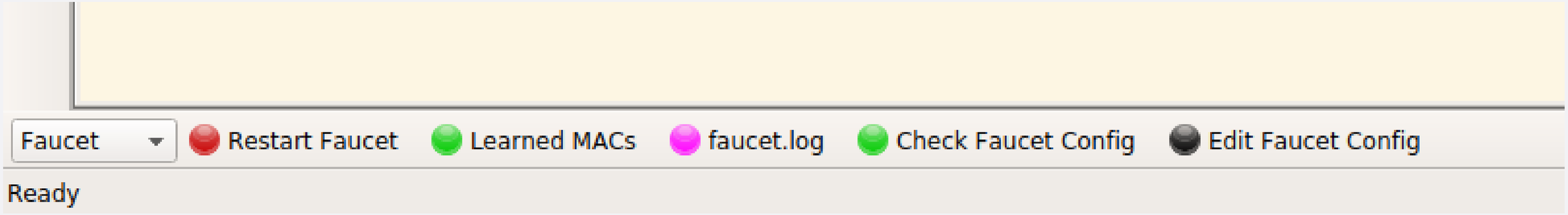
SDN Traffic Engineering for Wireless ISPs
Using DrainPipe to analyze RF

FAUCETCON2019

```
sajan@faucet:~/drain-pipe$ php artisan dp:plunge
Getting Capacity @ 10.172.10.140
TX Capacity: 773998080
RX Capacity:773538560
Getting Capacity @ 10.172.19.162
TX Capacity: 194560
RX Capacity:62720
Sending toward 10.172.19.162
Working through network conditionals.
Generating Faucet configuration.
Checking Faucet configuration.
Signaling HUP to Faucet
sajan@faucet:~/drain-pipe$
```

SDN Traffic Engineering for Wireless ISPs
Using DrainPipe to analyze RF

FAUCETCON2019



FAUCETCON2019

04 Production testing

SDN Traffic Engineering for Wireless ISPs
Prod Testing

FAUCETCON 2019

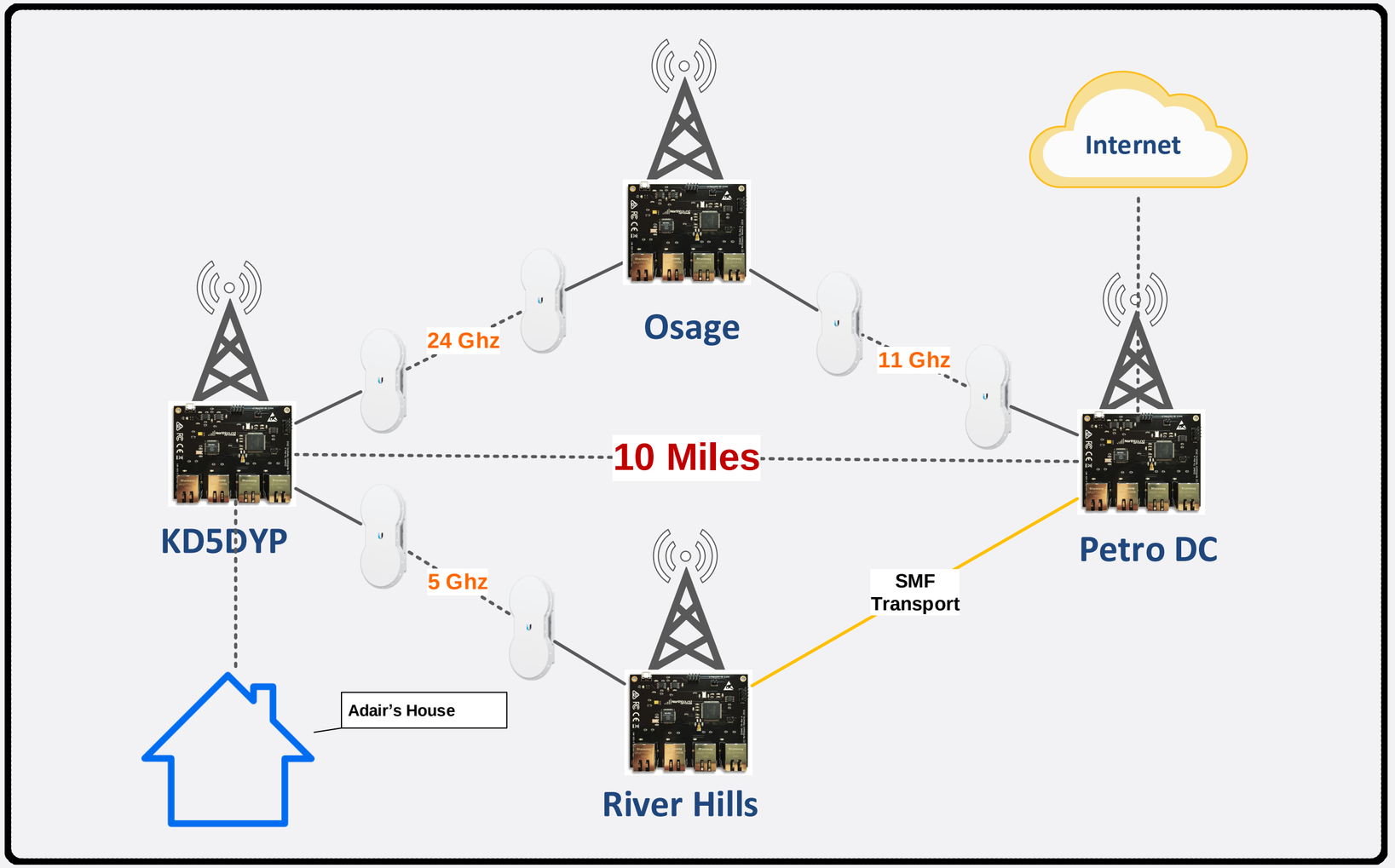
AW BROADBAND

Regional Fiber/Wireless ISP

- Serving the community of Amarillo, Texas
- Created a 4 switch test network over production microwave links
- Interested in the idea of putting the transport paths under SDN

SDN Traffic Engineering for Wireless ISPs
Prod Testing

FAUCETCON2019



Moving from lab to prod

- Deployed same YAML L2 stack into prod
- Using 5 Ghz and 24 Ghz unlicensed links
- 11 Ghz licensed links and SMF fiber

SDN Traffic Engineering for Wireless ISPs
Prod Testing

FAUCETCON2019

```
/ # ovs-dpctl dump-flows
recirc_id(0),in_port(3),eth(src=0c:34:c1:7d:70:00,dst=0c:34:c1:d3:9f:00),eth_type(0x8100),vlan(vid=3000),encap(eth_type(0x0800),ipv4(frag=no)), packets:255, bytes:18870, used:0.252s, actions:2
recirc_id(0),in_port(3),eth(src=0c:34:c1:7d:70:00,dst=0c:34:c1:d3:9f:00),eth_type(0x8100),vlan(vid=3000),encap(eth_type(0x0806)), packets:0, bytes:0, used:never, actions:2
recirc_id(0),in_port(2),eth(src=0c:34:c1:d3:9f:00,dst=ff:ff:ff:ff:ff:ff),eth_type(0x8100),vlan(vid=3000,pcp=0),encap(eth_type(0x0800),ipv4(frag=no)), packets:70, bytes:23460, used:4.352s, actions:3,pop_vlan,4
recirc_id(0),in_port(2),eth(dst=01:80:c2:00:00:00/ff:ff:ff:ff:ff:f0),eth_type(0x88cc), packets:42, bytes:2520, used:0.888s, actions:userspace(pid=4129686151,slow_path(controller))
recirc_id(0),in_port(2),eth(src=0c:34:c1:d3:9f:00,dst=0c:34:c1:7d:70:00),eth_type(0x8100),vlan(vid=3000),encap(eth_type(0x0806)), packets:0, bytes:0, used:never, actions:3
recirc_id(0),in_port(3),eth(src=0c:34:c1:7d:70:00,dst=ff:ff:ff:ff:ff:ff),eth_type(0x8100),vlan(vid=3000),encap(eth_type(0x0800),ipv4(frag=no)), packets:83, bytes:27958, used:1.172s, actions:2
recirc_id(0),in_port(2),eth(src=0c:34:c1:7d:70:00,dst=ff:ff:ff:ff:ff:ff),eth_type(0x8100),vlan(vid=3000,pcp=0),encap(eth_type(0x0800),ipv4(frag=no)), packets:84, bytes:28304, used:1.172s, actions:3,pop_vlan,4
recirc_id(0),in_port(3),eth(dst=01:80:c2:00:00:00/ff:ff:ff:ff:ff:f0),eth_type(0x88cc), packets:42, bytes:2520, used:0.888s, actions:userspace(pid=3894905306,slow_path(controller))
recirc_id(0),in_port(2),eth(src=0c:34:c1:d3:9f:00,dst=0c:34:c1:7d:70:00),eth_type(0x8100),vlan(vid=3000),encap(eth_type(0x0800),ipv4(frag=no)), packets:254, bytes:18796, used:0.252s, actions:3
/ # ovs-dpctl dump-flows
recirc_id(0),in_port(3),eth(src=0c:34:c1:7d:70:00,dst=0c:34:c1:d3:9f:00),eth_type(0x8100),vlan(vid=3000),encap(eth_type(0x0800),ipv4(frag=no)), packets:257, bytes:19018, used:0.560s, actions:2
recirc_id(0),in_port(3),eth(src=0c:34:c1:7d:70:00,dst=0c:34:c1:d3:9f:00),eth_type(0x8100),vlan(vid=3000),encap(eth_type(0x0806)), packets:0, bytes:0, used:never, actions:2
recirc_id(0),in_port(2),eth(src=0c:34:c1:d3:9f:00,dst=ff:ff:ff:ff:ff:ff),eth_type(0x8100),vlan(vid=3000,pcp=0),encap(eth_type(0x0800),ipv4(frag=no)), packets:72, bytes:24152, used:0.188s, actions:3,pop_vlan,4
recirc_id(0),in_port(2),eth(dst=01:80:c2:00:00:00/ff:ff:ff:ff:ff:f0),eth_type(0x88cc), packets:42, bytes:2520, used:2.996s, actions:userspace(pid=4129686151,slow_path(controller))
recirc_id(0),in_port(2),eth(src=0c:34:c1:d3:9f:00,dst=0c:34:c1:7d:70:00),eth_type(0x8100),vlan(vid=3000),encap(eth_type(0x0806)), packets:0, bytes:0, used:never, actions:3
recirc_id(0),in_port(3),eth(src=0c:34:c1:7d:70:00,dst=ff:ff:ff:ff:ff:ff),eth_type(0x8100),vlan(vid=3000),encap(eth_type(0x0800),ipv4(frag=no)), packets:83, bytes:27958, used:3.280s, actions:2
recirc_id(0),in_port(2),eth(src=0c:34:c1:7d:70:00,dst=ff:ff:ff:ff:ff:ff),eth_type(0x8100),vlan(vid=3000,pcp=0),encap(eth_type(0x0800),ipv4(frag=no)), packets:84, bytes:28304, used:3.280s, actions:3,pop_vlan,4
recirc_id(0),in_port(3),eth(dst=01:80:c2:00:00:00/ff:ff:ff:ff:ff:f0),eth_type(0x88cc), packets:42, bytes:2520, used:2.996s, actions:userspace(pid=3894905306,slow_path(controller))
recirc_id(0),in_port(2),eth(src=0c:34:c1:d3:9f:00,dst=0c:34:c1:7d:70:00),eth_type(0x8100),vlan(vid=3000),encap(eth_type(0x0800),ipv4(frag=no)), packets:256, bytes:18944, used:0.560s, actions:3
```

Moving traffic

- Started with ACLs and port output
- Mixed success, still working on match to make it more intelligent.
- Need to move to L3 stacking now that we understand L2 stacking

FAUCETCON2019

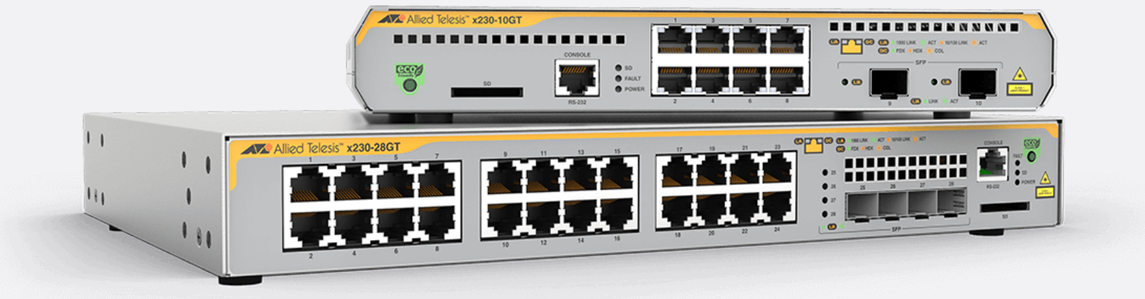
05 Next steps

SDN Traffic Engineering for Wireless ISPs
Next steps

FAUCETCON2019

Continue developing DrainPipe and move to L3 stacking, BGP and prod hardware

- Need to work on DrainPipe and how to compute available paths + bandwidth available to feed into the decision process
- Move the design to L3 stacking with BGP to integrate into the prod ISP network
- Use prod hardware to have better debugging capability, more capacity and environmentally hardened.



SDN Traffic Engineering for Wireless ISPs

Thank You

FAUCETCON2019

Faucet

AW BROADBAND

Google