

Building a regional service provider network using SDN (for fun)



By  @nz_liam

Why/How? A number of favourable factors... making market entry possible to smaller players

- Chorus bond requirements have come down, (previously \$50k now approx. \$10k)
- Chorus UFB Tail Extensions
 - Total of 33 UFB Areas, 24 Chorus (70%)
 - Previously an interconnect was required in each UFB area, (plus exchange space, equipment, backhaul etc)
 - With tail extensions only 3 Interconnects are required for the North Island
 - Auckland
 - Hamilton
 - Wellington
- Cheap Backhaul
 - Chorus Launched CRT, bringing 10G DWDM P2P links down to ~\$4k / month (\$0.40 per mbit).
 - This lowered national backhaul costs substantially from the previous cost of about \$5 per mbit.

What are the requirements, and where does SDN come in?

- Lower cost structures in acquiring tails help, but its not enough.
 - You still need exchange space, hardware and software... exchange space was easy we brought it off 2Degrees.
- Hardware is the tricky (and fun) part...
 - Vendor gear is expensive and often low density.
 - We wanted;
 - Small footprint (1RU)
 - High Port Density (16 x 10G)
 - 48VDC Power (for Exchanges)
 - Converged infrastructure, (Switch, Router, BRAS, RADIUS, DNS etc)
 - Cost Effective (Cheap)

Building your own SDN appliance... (and what is SDN actually?)

- To meet our requirements (footprint, density, cost) we decided to build our own SDN appliance.
- What is SDN, does anyone actually know?
 - For me it's the combination of virtualised networking in the form of virtual routers, virtual switching, and associated virtualised applications (RADIUS, DNS etc) on converged hardware.
 - Traditional vendors (Juniper / Cisco) do a bad job of producing virtual routing software for converged hardware
 - Generally only support old OS versions
 - Require specific hardware (network cards)
 - Require specific driver patches
 - Net result – you're replacing dedicated vendor hardware with dedicated SDN hardware for a single vendor...

Our SDN Environment – Hardware

- Finding virtualisation hardware for Exchanges is hard, 48VDC is a PITA
- We actually modded some CCR's to get up services up and running quickly for a customer



Mikrotik CCR1016 with Dual 48VDC Conversion in Napier Exchange

Our SDN Environment – Hardware

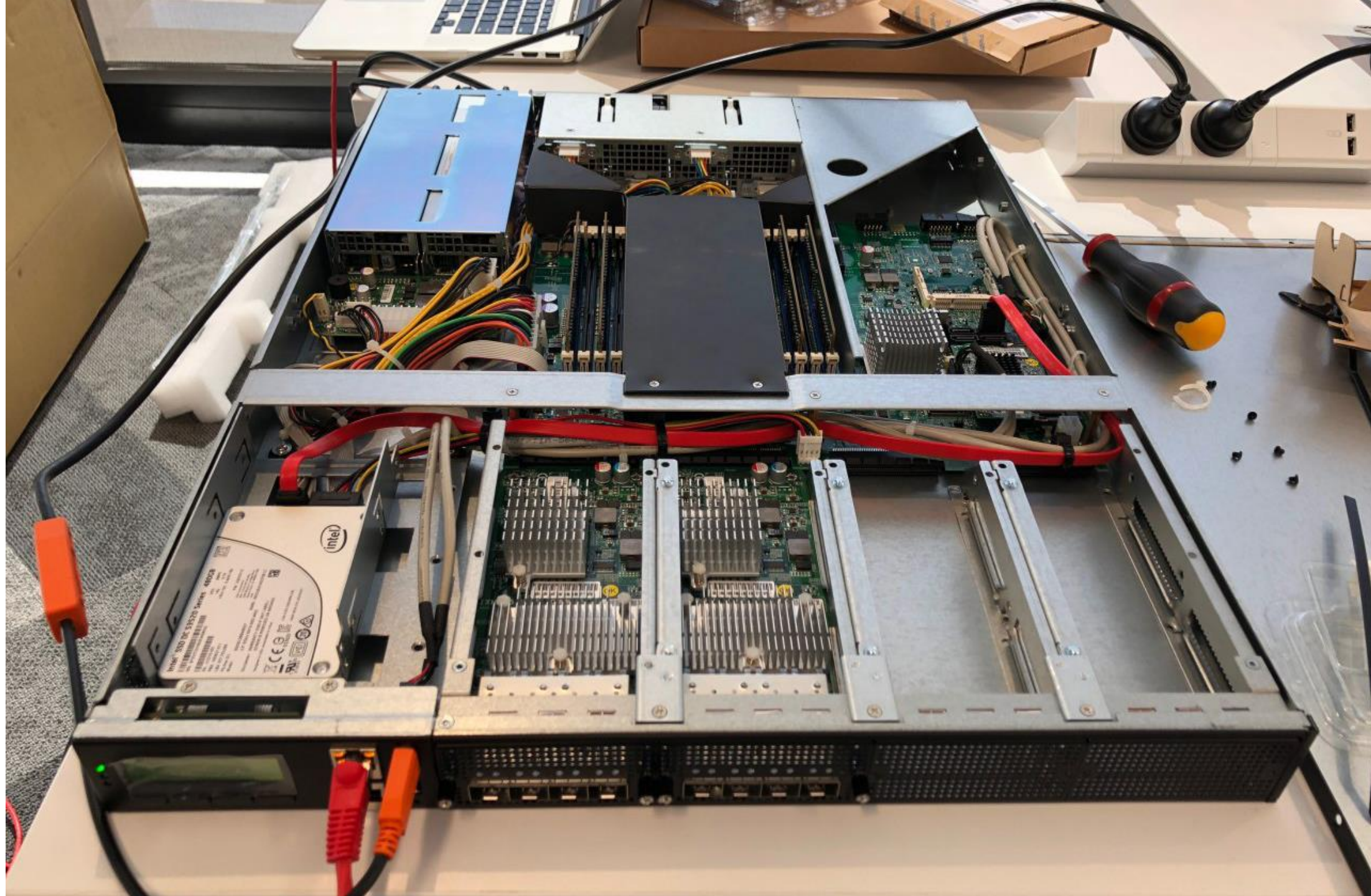
- We found a great Taiwanese OEM – Lanner
- NCA-5510 Appliance
 - 48VDC Power Option
 - 1RU
 - Up to 4 Network Modules, with 4 x 10G
 - Intel Xeon E5 Support – Max 95W TDP
 - Up to 256GB ECC DDR4 RAM
 - Dual 2.5” SSD
 - IPMI (OOB)



Our SDN Environment – Hardware

- Our Build, Intel Xeon E5-2640 v4 (10 Core @ 2.40GHz), 64GB ECC DDR4, 480GB Intel SSD, 4-16x Intel 10G Nic, Price tag approx. \$7K





Our SDN Environment - Software

- Proxmox VE 5 (KVM, QUEMU)
 - Linux hypervisor in an easy use out of the box package
 - Has commercial support
 - Runs Debian – easy to modify the HV environment
- Open vSwitch
 - OVS is Awesome (queue rant)
 - NIC's become switch ports, (access ports, trunk ports, QinQ ports etc)
 - Multiple virtual switches
 - Links between vSwitches
- For Routing - Mikrotik CHR...

Mikrotik CHR

- It runs on everything, (vmware, KVM, it just works)
 - No specific requirements, you can actually run CHR on converged hardware
- Fulfills our base requirements
 - VLANs (w/ QinQinQ Support)
 - OSPF
 - Bridging
 - MPLS / VPLS
 - BRAS
- Affordable

Some testing...

- Coffee Table ✓
- Wine ✓
- Hardware ✓
- Hypervisor (PVE) ✓
- OVS from source?
- Does QinQ work?
- What's performance like?



Some testing...

- Optics work
- OVS Works
- 1G VPLS uses 10% of 1 CPU Core
- QinQ works
- Great Success!

The screenshot displays a network management interface with a session ID of 192.168.32.112. It features an 'Interface List' table and two configuration windows for VLANs.

Name	Type	Actual MTU	L2 MTU	Tx
ether1	Ethernet	1600		34
vlan1	VLAN	1500		33
vlan111	VLAN	1500		
vlan67	VLAN	1500		
vlan123	VLAN	1500		

Interface <vlan67> Configuration:

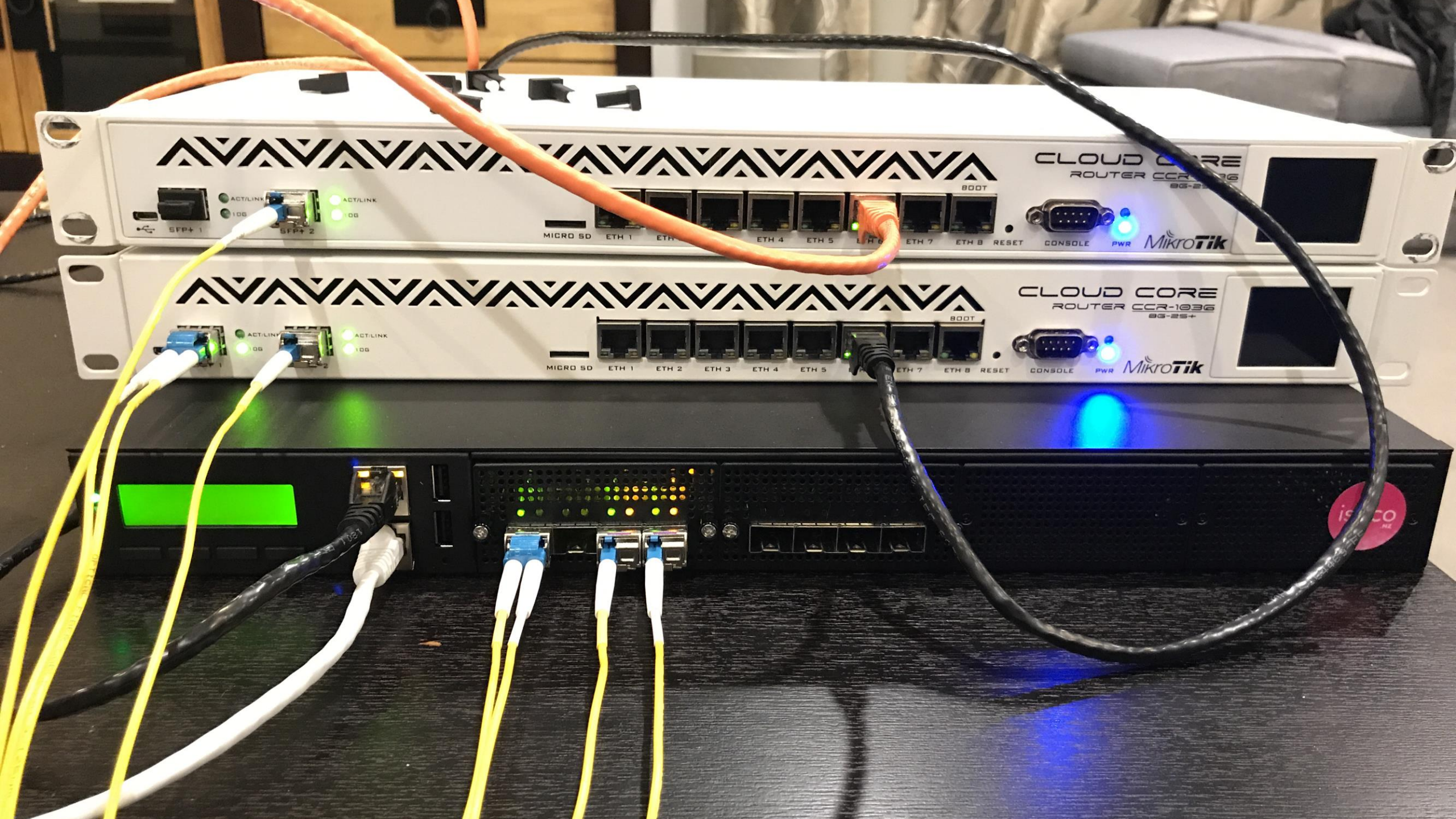
- Name: vlan67
- Type: VLAN
- MTU: 1500
- Actual MTU: 1500
- L2 MTU:
- MAC Address: 36:C9:98:8C:9E:CD
- ARP: enabled
- ARP Timeout:
- VLAN ID: 67
- Interface: vlan111
- ☐ Use Service Tag

Interface <vlan111> Configuration:

- Name: vlan111
- Type: VLAN
- MTU: 1500
- Actual MTU: 1500
- L2 MTU:
- MAC Address: 36:C9:98:8C:9E:CD
- ARP: enabled
- ARP Timeout:
- VLAN ID: 111
- Interface: ether1
- ☒ Use Service Tag

Terminal Output:

```
175 192.168.67.3 56 64 0ms
176 192.168.67.3 56 64 0ms
177 192.168.67.3 56 64 0ms
178 192.168.67.3 56 64 0ms
179 192.168.67.3 56 64 0ms
sent=180 received=180 packet-loss=0% min-rtt=0ms avg-rtt=0ms
max-rtt=0ms
SEQ HOST SIZE TTL TIME STATUS
180 192.168.67.3 56 64 0ms
181 192.168.67.3 56 64 0ms
182 192.168.67.3 56 64 0ms
183 192.168.67.3 56 64 0ms
184 192.168.67.3 56 64 0ms
185 192.168.67.3 56 64 0ms
186 192.168.67.3 56 64 0ms
```

Exchange & Handover time!

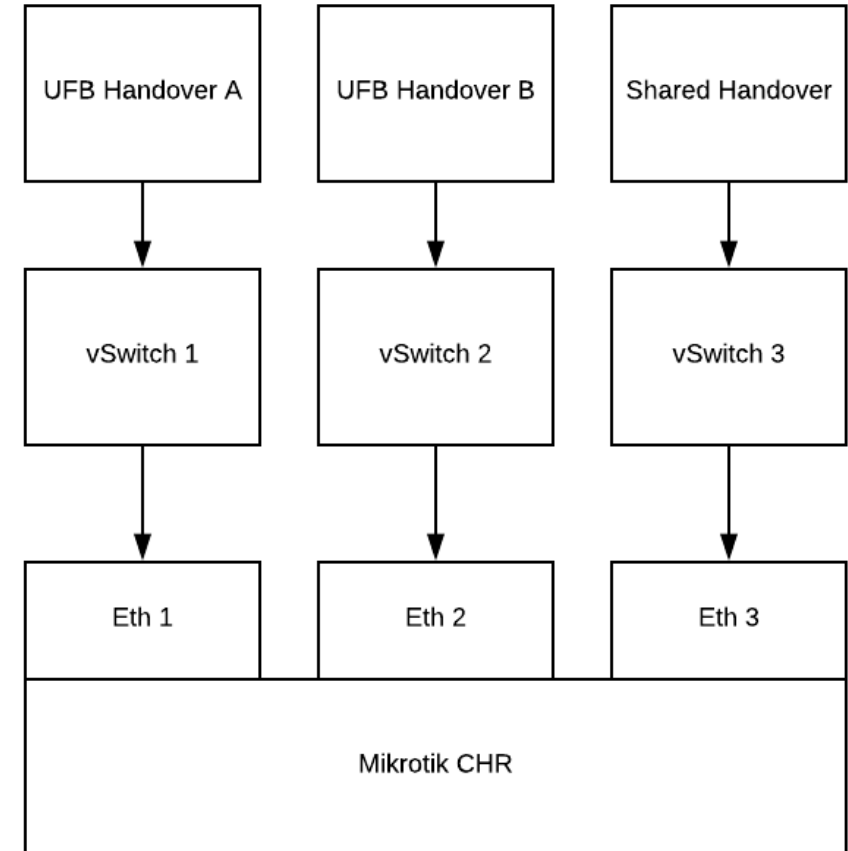
- Chorus is complicated...
- So many portals... (each with different logins)
 - Wireline
 - Portal (Fibre)
 - Checkmate
 - SSP (Self Service Portal)
 - SP (Service Provider Portal)
 - Gateway (Service Performance Manager)
 - Legacy Portal
 - CCM (Maps)

Exchange & Handover time!

- Some challenges
 - MDR had no more 10G Handover Ports... Handover & DFAS from Mt Eden
 - 1G BiDi optics don't work well in 10G cards, (even in an unsupported mode)
 - We ordered a 1G Shared Handover (ADSL / VDSL)
 - FedEx in a 1G Network Card to resolve
 - It crashed... (networking stopped passing traffic)
 - Hard to diagnose stuff when everything is broken
 - Suspect a Virtio Driver / Kernel / vSwitch Issue
 - Updated vSwitch, Kernels and Virtual routers, problem solved
 - EUBA & WVS is a pain, particularly EUBA with PPPoA

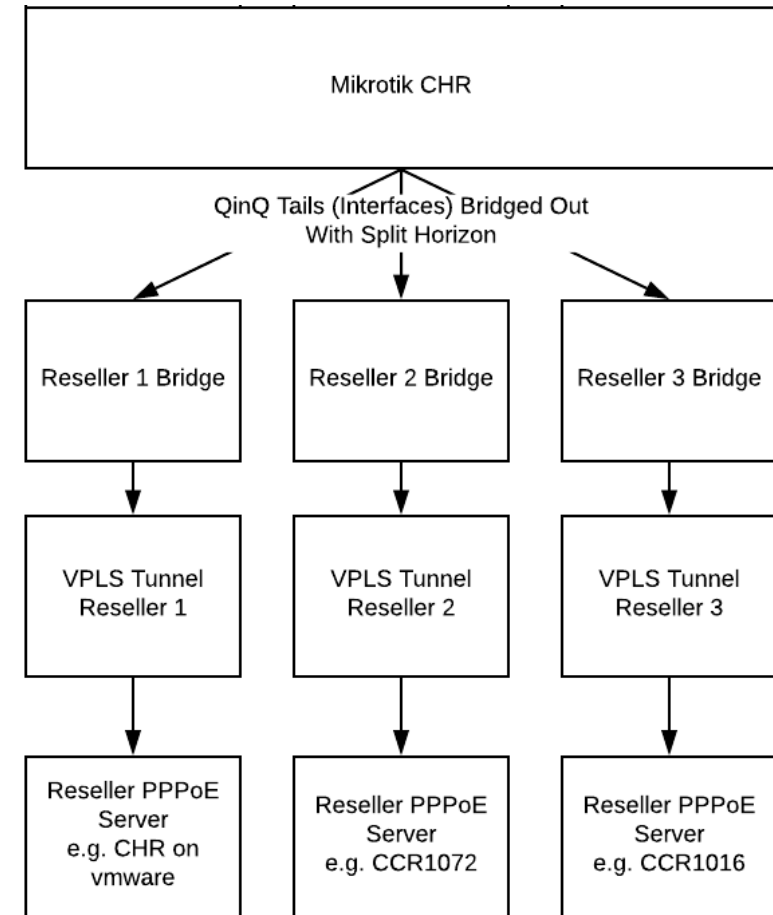
Dealing with QinQ Handovers

- Each customer tail comes as QinQ (or QinQinQ)
- OVS QinQ doesn't use the right ethertype (0x88a8 not 0x8100)
- Separate each handover (OVS port) into a new vSwitch for collision avoidance



Dealing with QinQ Handovers – Part 2

- QinQ Means you have a S-VID (Outer Tag) and C-VID (Inner Tag) for each circuit (tail)
- Connect each tail C-VID to a bridge for the specific reseller
- Use split horizon bridging to prevent tails talking to each other & STP
- We VPLS each resellers bridge back to Datacenter for PPPoE Termination on their specific BRAS



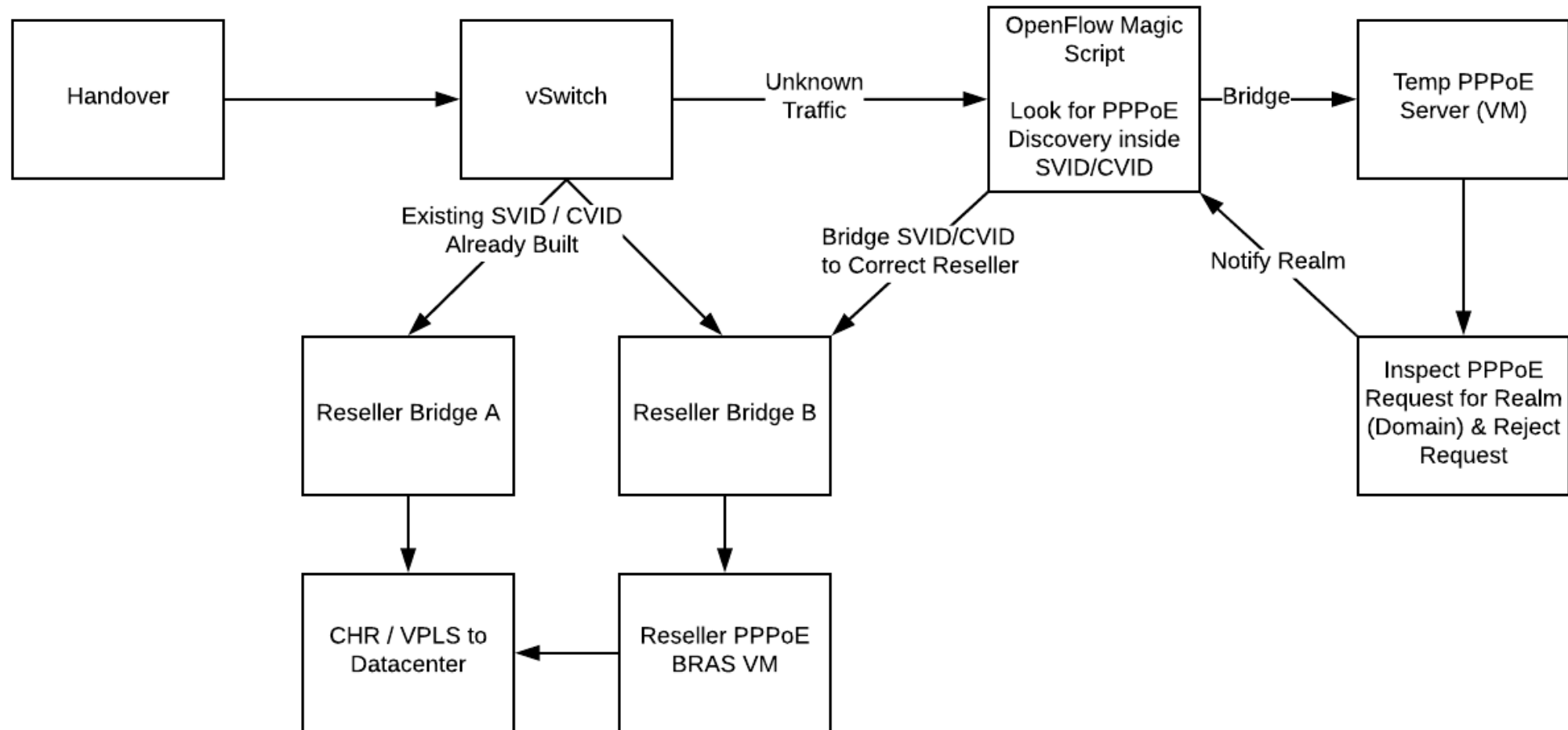
Challenges with our design

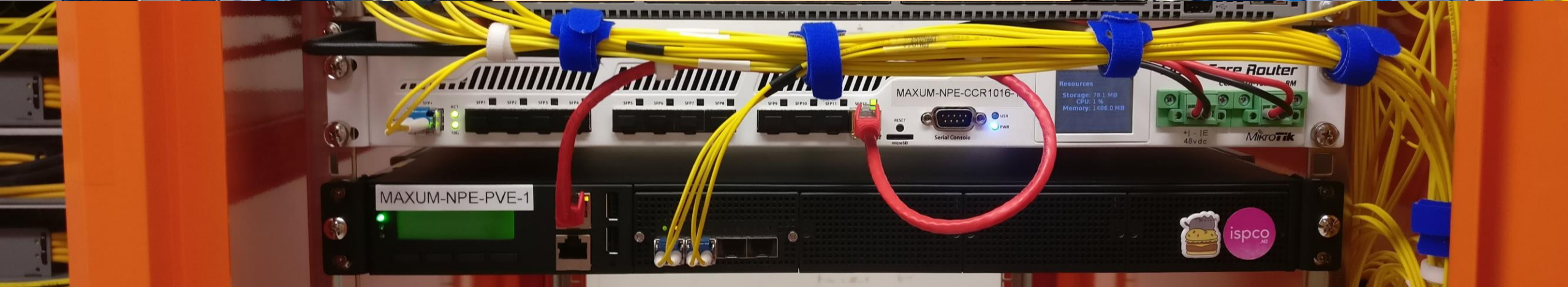
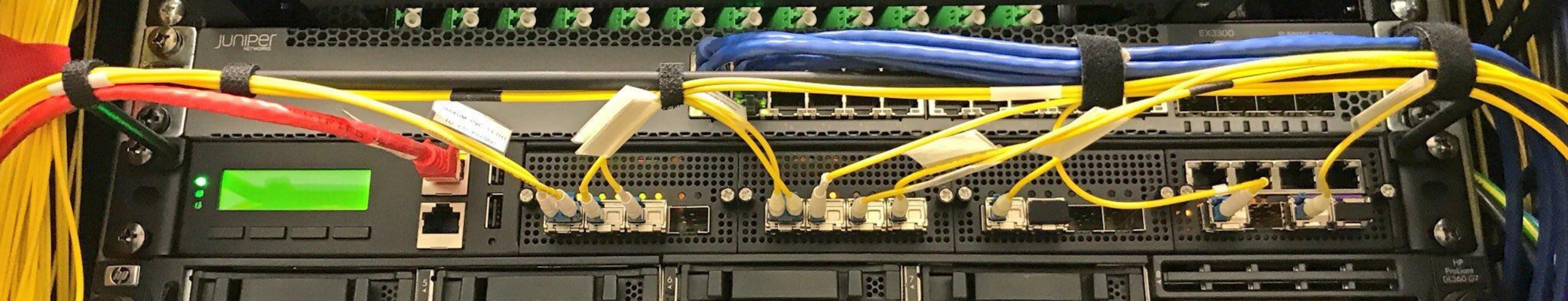
- Each customer circuit has to be built manually
 - S-VID Interface, C-VID Interface, Bridge Port...
- UFB is Easy SVID / CVID's are sequential
- EUBA / VVS (ADSL/VDSL) SVID / CVID's are NOT sequential
 - Different S-VID for each Exchange, random C-VIDs based on line card position
 - Line cards explode... CVID's change...
- EUBA is even worse
 - Using PPPoE / VLAN 10 you get the assigned C-VID by Chorus
 - Using PPPoA you get the assigned C-VID + 1600
 - Most shitty modems use PPPoA, but we also use PPPoE when bridging...

Future design improvements

- More performance using Open vSwitch w/ DPDK (and proper benchmarking)
- Custom Build of OVS with 0x8100 QinQ ethertype
- Making the front LCD do something (tell jokes)
- Hosted infrastructure for resellers
 - BYO Handover
 - Use our hardware / virtualisation platform / backhaul
- Dynamic VLAN Creation
 - NZNOG + Beer + WAND = Something Cool is Coming

Dynamic VLAN Creation





The End Product

- Chorus Handovers
 - Auckland
 - Hamilton
 - Napier
 - Wellington
- Unison Handover
 - Napier
- Datacenter POP's
 - 220 Queen Street
 - SkyTower
 - Vocus Albany

Questions? @nz_liam on Twitter

