Streamlining common operational lifecycle tasks @ ebay

So, what’s this all about?

ebay has operated a multivendor 'routed' network since approximately 2009. Numerous challenges have presented themselves during the transition away from the traditional L2+VLAN network.

We’ll present ways we solved some of these problems with help from our partner vendors, and how we used automated management to orchestrate several common operational lifecycle tasks.

A key to our approach is to work with our partner vendors to make these features available in the industry so everyone can take advantage

So, what’s this all about?

In this talk:

- switch-build automation: ZTP (Zero Touch Provisioning)
- simplifying administrative cost-out of links in a BGP-as-IGP network
- Hitless code upgrade on TOR switches: ISSU (In Service Software Upgrade)
- What are we working on for the future?

ebay’s approach

How are we able to get ‘wishlist’ features implemented?

- RFP
  - Multivendor is key to driving features
  - Table-stakes: Becomes requirement if N vendors support it
  - Advertise requirement 1-2 years ahead of time
  - Make it a differentiating feature
- Cultivate partner relationship with vendors

ebay’s approach

How are we able to get ‘wishlist’ features implemented?

- Explain why we need the feature! Say things like…
  - This isn’t just helping ebay.. It helps YOU
  - Everyone will have this problem. We just have it first
  - I can’t afford to operate a network based on your gear
  - Look how much work this saves us
  - This is table stakes, not a paid add-on (eg ZTP)

ebay’s approach

- Bring it up with all decision makers
- Bring it up with all decision makers
- Bring it up with all decision makers

…

- Bring it up with all decision makers

Seriously. We tell our problems to everyone who will listen, ask them to help us solve them.

Building/provisioning - ZTP

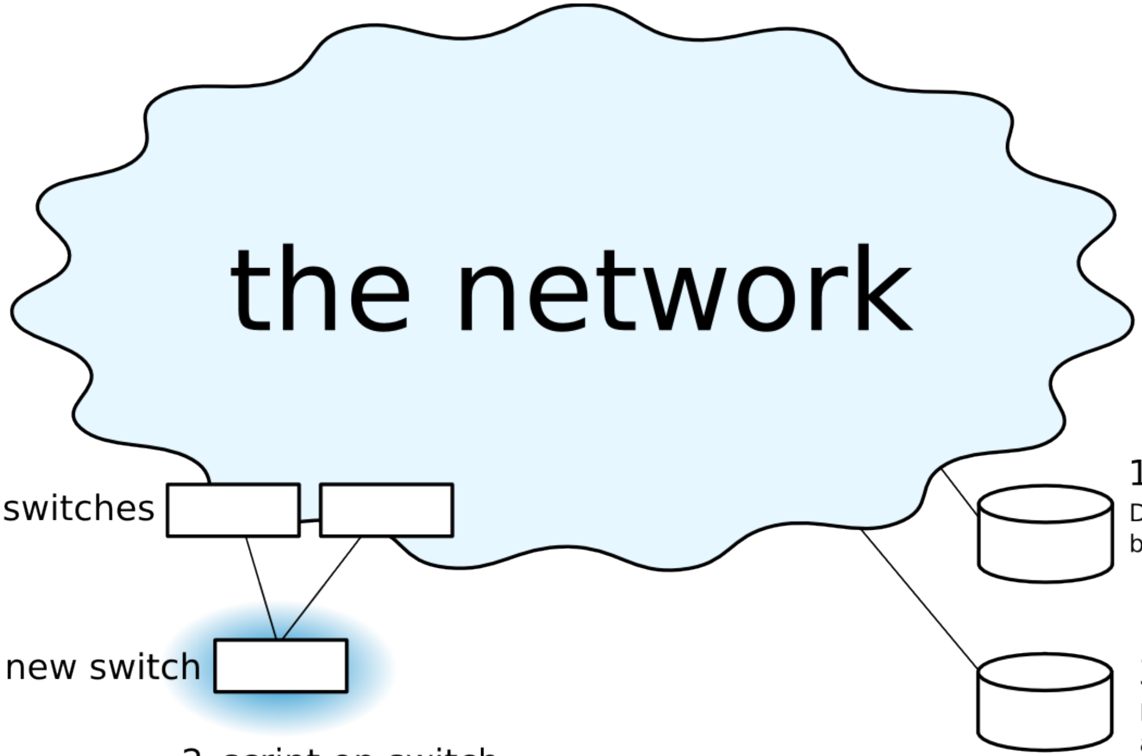
- What challenges did we face?
  - 2009: 6500 / EOR design moving to TORS
  - combat the 'another device to manage' mentality.
  - Solve inconsistent implementation
  - Don’t have humans doing repetitive tasks!

- Our ZTP concept
  - 100% automation. The only human is involved is the one that racks the switch
  - Use industry standard tools that have been around forever
  - Configuration AND code-image
  - Run a script on the switch to configure it

Building/provisioning - ZTP

- basic ZTP
  - Configure using DHCP options only
- orchestrated ZTP
  - Require minimum information about the hardware in advance
  - Automated config generation
  - no personalization of any devices until the last moment
- RMA
  - When replaced, gear assumes identity of failed device

Building/provisioning - ZTP



1: DHCP

Detects switch model, returns
bootstrap script

2: script on switch

Requests config/build-info from webserver

3: web-script

Determines switch identity,
and generates configuration

Reports build event to
upstream tools

4: script on switch

Determines code-version based on config file.
Downloads if necessary

Building/provisioning - ZTP

- industry influence and challenges
  - Make the feature
  - boot mode challenge
  - 10g/40g autosense
  - Nextgen: LLDP for identity instead?

- ZTP in-a-box. Ask your favorite vendor!
  - Package ZTP and required tooling in a VM for ease of deployment

Building/provisioning - ZTP

Lessons learned

- Don’t boot in L2 mode! All ports in vlan1, DHCP from vlan1 interface may be easy to implement, but isn’t the right approach.
  - Attached hosts compete for IP addresses. There are more of them!
  - Network protection features disable uplink ports: BPDU-guard, STP mismatch, trunk mode mismatch, etc
  - Unintended adjacencies (eg OSPF) may form between upstream switches
- Restart from the beginning if anything goes wrong
  - Autobuild is an automated process. Fix it on the backend if it’s broken, and it’ll pick up the changes on the next retry.

BGP cost-out simplification

We are in transition from OSPF to BGP. Our L1/L2 techs are familiar with ‘draining’ in OSPF, but what happens when we switch to BGP?

Current situation: OSPF

- fairly straightforward, well understood
- Apply a metric to the interfaces on both sides
- oops, now add add ipv6. Have to cost-out *two* address families per link
- max-metric and associated commands for whole-box draining.

BGP cost-out simplification

- Challenges switching from OSPF to BGP
  - Costing out links is not interface based anymore
  - Look up which neighbors are on the interface and apply a route-map to them
  - This takes longer and is more error-prone

- How can we make this simpler?
  - It’d be nice to handle both families at once
  - How about draining traffic in both directions from ‘one side’
  - Do we have to do neighbor lookups? We really just want to cost a link out

BGP cost-out simplification

- This isn’t just an ebay problem.
  - Other companies may have different ways of costing out a link in BGP
  - A user-defined route-map is needed

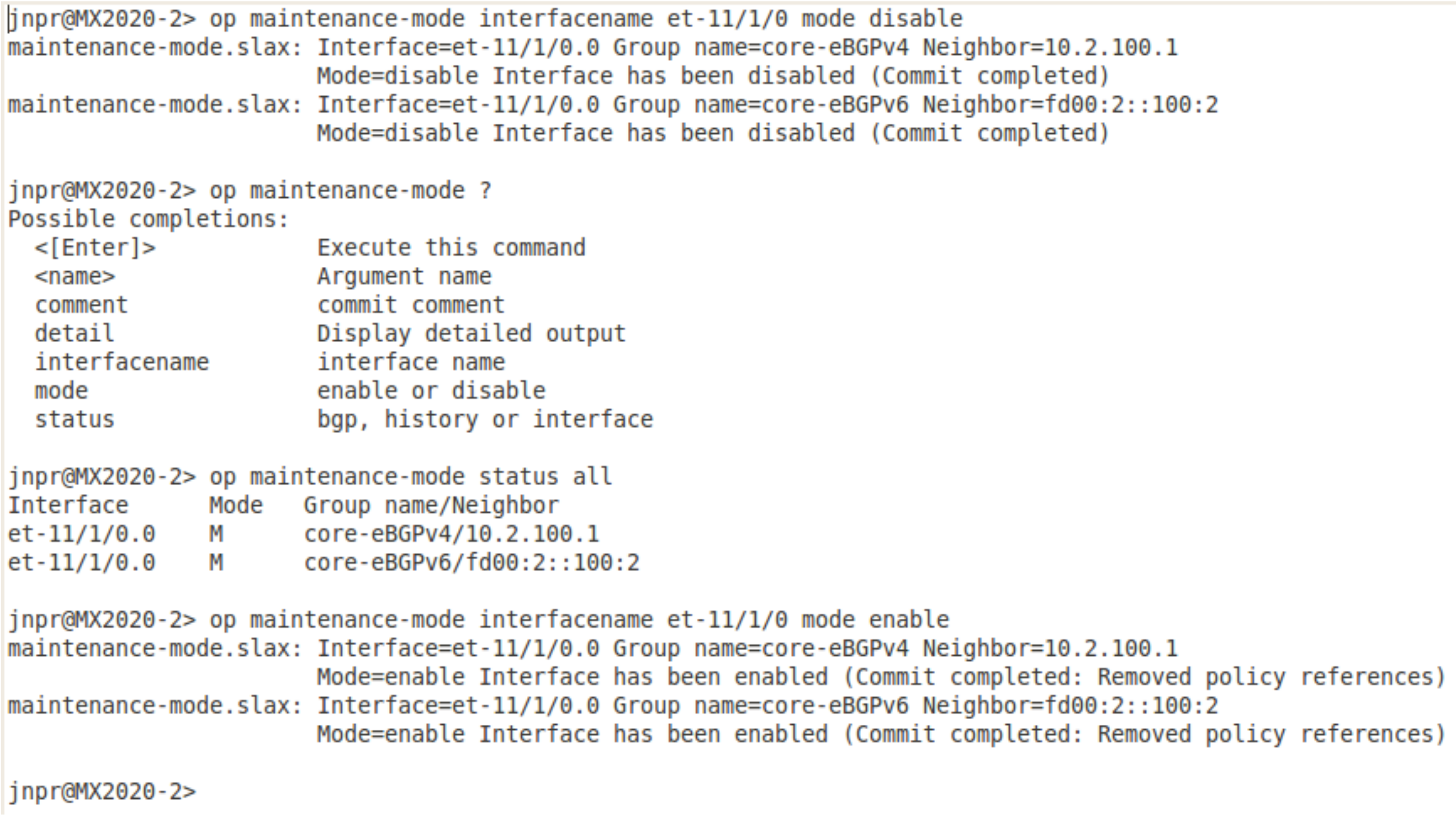
BGP cost-out simplification

Ways our partners solved this problem

- Cisco: user-script to do lookups
- Juniper: script for now, OS feature on the way
- Arista: OS feature for BGP cost-out

BGP cost-out simplification

Juniper: Today: script assisted
Future: OS feature



```
jnpr@MX2020-2> op maintenance-mode interfacename et-11/1/0 mode disable
maintenance-mode.slax: Interface=et-11/1/0.0 Group name=core-eBGPv4 Neighbor=10.2.100.1
                       Mode=disable Interface has been disabled (Commit completed)
maintenance-mode.slax: Interface=et-11/1/0.0 Group name=core-eBGPv6 Neighbor=fd00:2::100:2
                       Mode=disable Interface has been disabled (Commit completed)

jnpr@MX2020-2> op maintenance-mode ?
Possible completions:
  <[Enter]>            Execute this command
  <name>               Argument name
  comment              commit comment
  detail               Display detailed output
  interfacename        interface name
  mode                 enable or disable
  status               bgp, history or interface

jnpr@MX2020-2> op maintenance-mode status all
Interface      Mode   Group name/Neighbor
et-11/1/0.0    M      core-eBGPv4/10.2.100.1
et-11/1/0.0    M      core-eBGPv6/fd00:2::100:2

jnpr@MX2020-2> op maintenance-mode interfacename et-11/1/0 mode enable
maintenance-mode.slax: Interface=et-11/1/0.0 Group name=core-eBGPv4 Neighbor=10.2.100.1
                       Mode=enable Interface has been enabled (Commit completed: Removed policy references)
maintenance-mode.slax: Interface=et-11/1/0.0 Group name=core-eBGPv6 Neighbor=fd00:2::100:2
                       Mode=enable Interface has been enabled (Commit completed: Removed policy references)

jnpr@MX2020-2>
```

BGP cost-out simplification

Cisco: Today: script assisted
Future: OS feature + GSHUT

- Python Script for automated COST-OUT and COST-IN
- # cli alias name bgpmod source bgp-oos-policy-v2_3.py
- Usage CLI:

```
COST-OUT:
    CMI-D-N7009-1# bgpmod -i all -a apply
    CMI-D-N7009-1# bgpmod -i eth3/1 -a apply
COST-IN:
   CMI-D-N7009-1# bgpmod -i all -a remove
   CMI-D-N7009-1# bgpmod -i eth3/1 -a remove
```

- Configuration Overview

```
template peer-policy link-out-of-service
    route-map out-of-service-out out
    route-map out-of-service-in in
!
neighbor 40.1.1.3
    inherit peer TOR
    address-family ipv4 unicast
      inherit peer-policy link-out-of-service 10
!
neighbor 2001:40:1:1::3
    inherit peer TOR
    address-family ipv6 unicast
      inherit peer-policy link-out-of-service 10
!
route-map out-of-service-out permit 10
  set as-path prepend 65302
!
route-map out-of-service-in permit 10
  set as-path prepend 65302
```

BGP cost-out simplification

What about GSHUT?

- Described in http://tools.ietf.org/html/draft-ietf-grow-bgp-gshut-05
- Technology for operational procedures aimed at reducing the amount of traffic lost during planned maintenances of routers or links.
- Either a **single** neighbour or **all** neighbors simultaneously can be set in GSHUT mode:

```
Device> enable
Device# configure terminal
Device(config)# router bgp 65000
Device(config-router)# bgp graceful-shutdown all neighbors 180 local-preference 20 community 10
Device(config-router)# bgp graceful-shutdown all neighbors activate
Device(config-router)# end
```

- **Any of this configuration can be part of maintenance mode profile**
- New BGP knobs are under work to enhance GSHUT for Maintenance mode
  - Add ‘AS-prepend’ as a configuration option to also add dynamically AS numbers in the AS-path-list

Feature velocity - ISSU

- ISSU has been around forever, why are we talking about it like it's something new?
  - ISSU on chassis is not useful in a L3 datacenter network
    - Most large networks use some variant of L3 + Clos networks
    - Upgrading a spine or core switch is easy! ISSU is irrelevant.

- ISSU on TOR switches
  - ebay has thousands of ToRs
  - coordinating upgrades of hundreds or thousands of switches in a multi-customer environment is a non-starter
  - feature velocity suffers. If you can't upgrade, you can't consume new features!
  - Plenty of opportunity to test, limited consequences

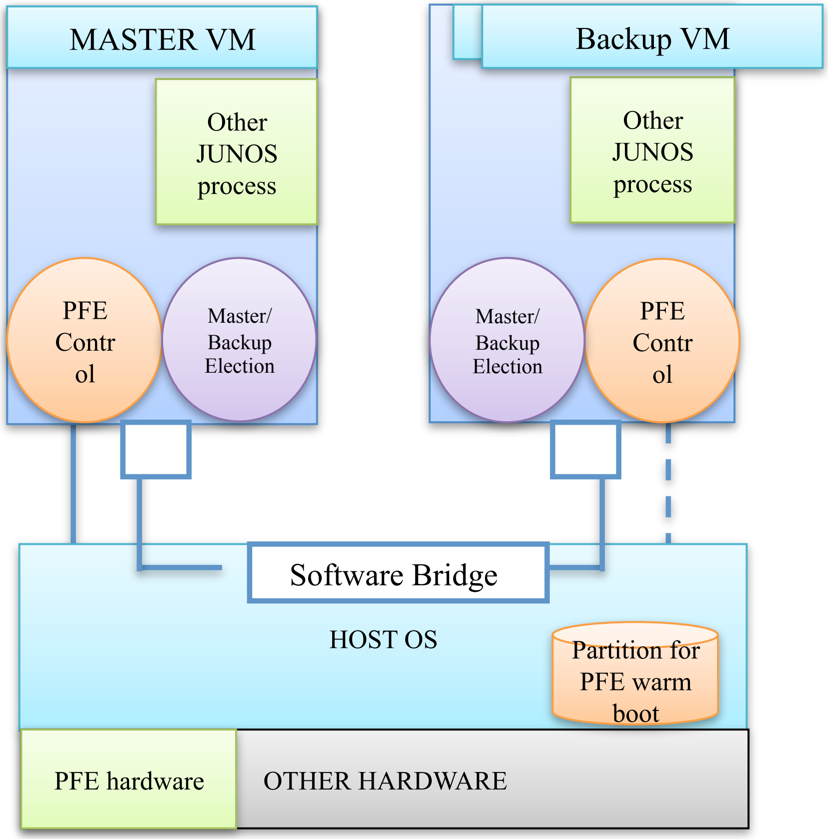
Feature velocity - ISSU

- industry influence
  - Long-term project – 2 years in the making!
  - Curiously, each vendor took a slightly different approach

Feature velocity - ISSU

Juniper:

- Master JunOS VM controls the hardware–PFE and FRU on the system
- Master issues upgrade command
- System launches a new JunOS VM with new image as backup
- All states are synchronized to the new backup JunOS
- Detach PFE from current master, then attach to backup JunOS (hot move)
- The PFE control component in new master will control the forwarding
- Stop the new backup VM



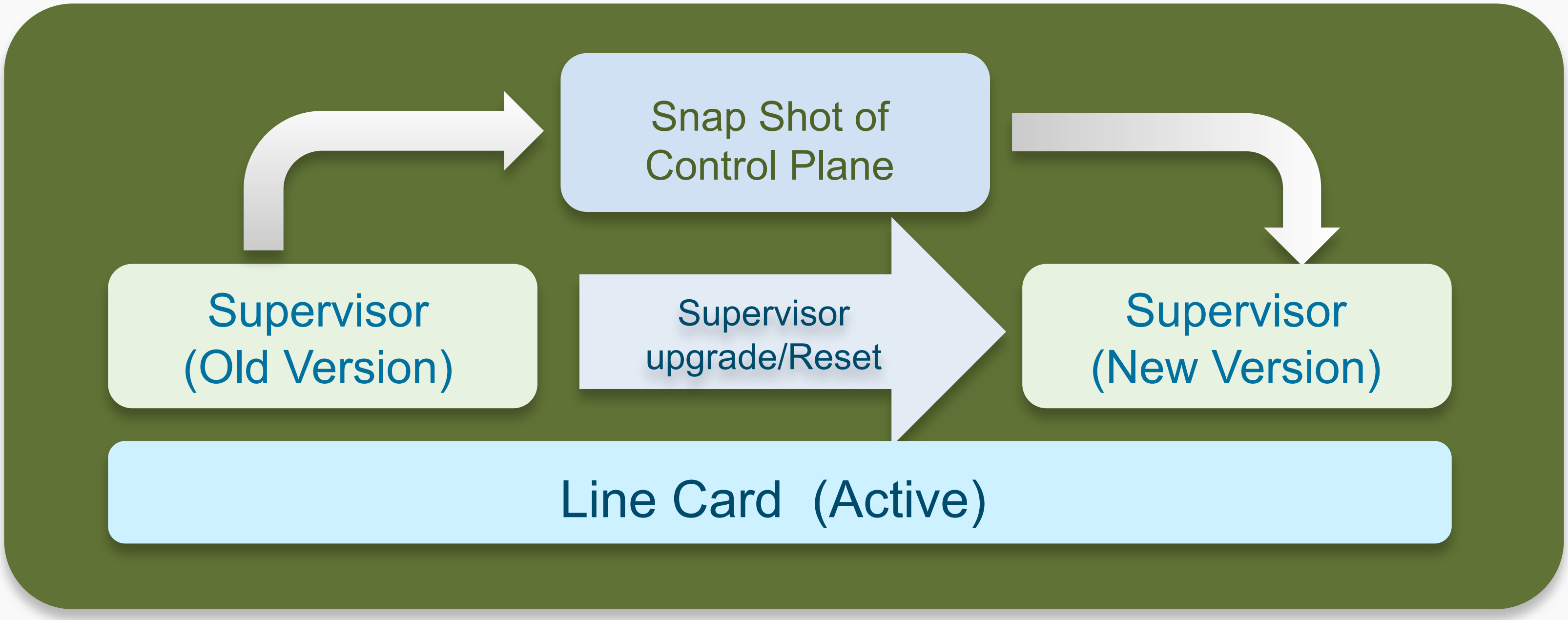
Feature velocity - ISSU

Arista:

- Has committed to releasing an ISSU-like feature that meets ebay’s requirements.
- Unable to share details publicly at this time due to SEC rules

ISSU on Nexus 9300 Series Switches

- Nexus 9300 switch is internally modeled as a modular switch with a single supervisor and a single line card.
- During ISSU, the supervisor gets reset while the line card remains up and forwarding traffic during the entire process.



Snap Shot of Control Plane

Supervisor (Old Version)

Supervisor upgrade/Reset

Supervisor (New Version)

Line Card (Active)

Feature velocity - ISSU

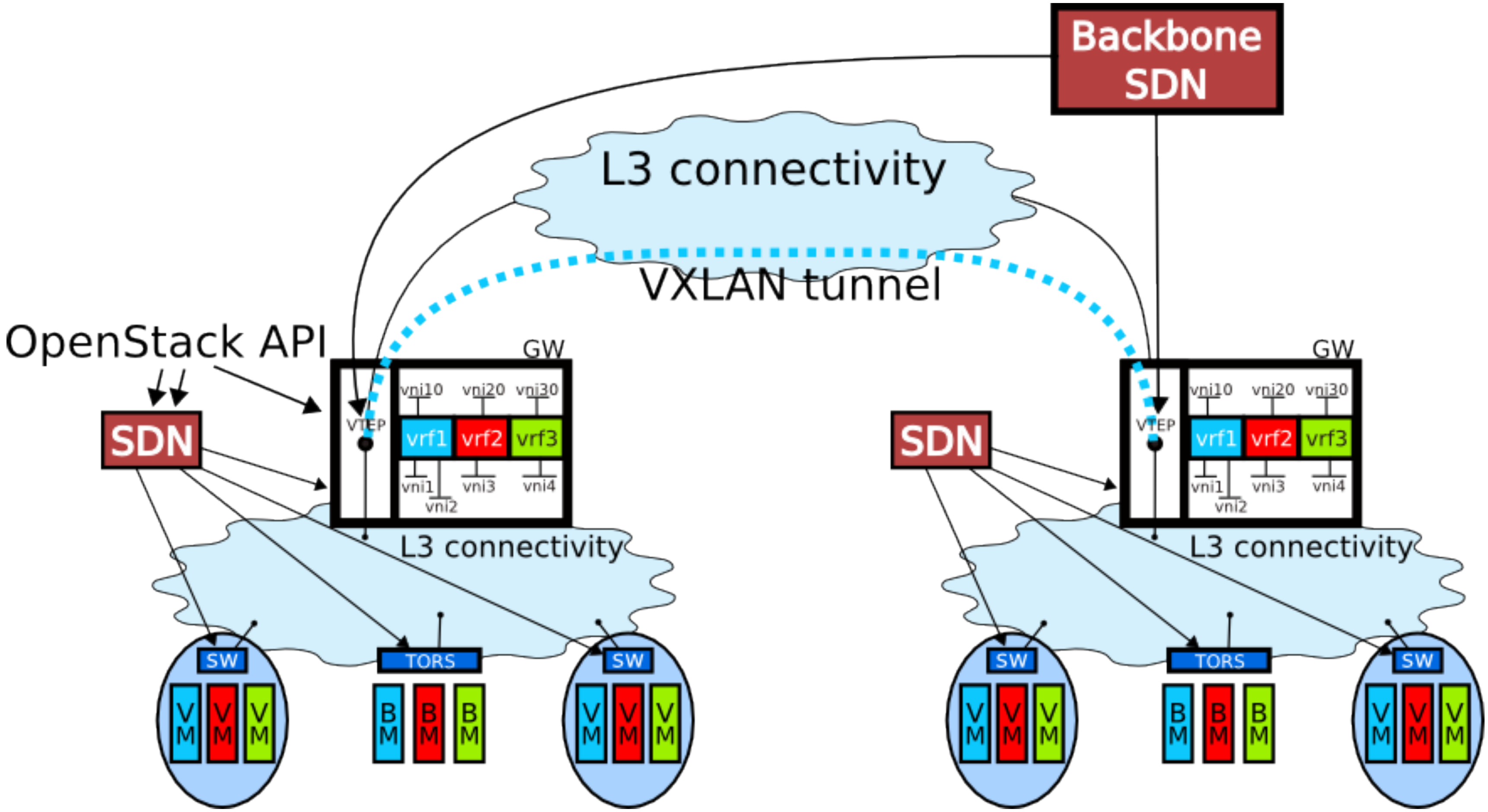
- Are we solving the right problem?
  - What about Multi-NIC?
    - 2x # of TORS
    - Less usable bandwidth vs single-nic
    - Still in the same failure domain!
  - Isn’t this just masking application design deficiencies?
  - Switch code development complexity
    - More difficult to make ISSU enabled features
    - More bugs, etc.
    - Features take longer – lower supply-side feature velocity

What’s next?

Here are a few other features we are currently in the process of promoting

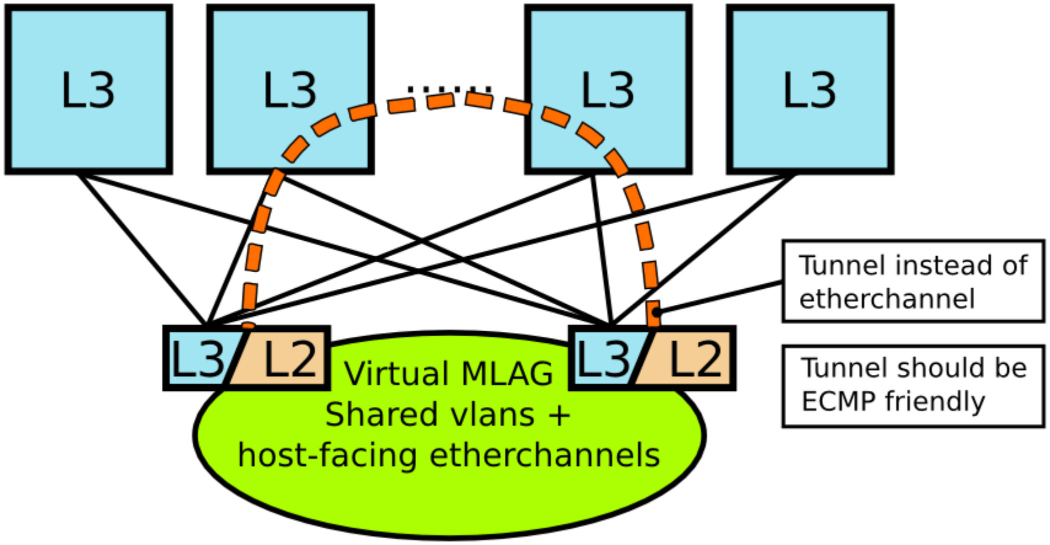
- ‘device personality’ blob for backup/restore
- High resolution metrics / ‘compute clusters’ made of switch cpus
- multi-controller Overlay/SDN support on the same switch
- Virtual MLAG (multi-switch etherchannel)

What’s next?



Backbone SDN
L3 connectivity
VXLAN tunnel
OpenStack API
SDN
GW
VTEP
vni10
vni20
vni30
vrf1
vrf2
vrf3
vni1
vni2
vni3
vni4
L3 connectivity
SW
TORS
SW
VM
VM
VM
BM
BM
BM
VM
VM
VM
SDN
GW
VTEP
vni10
vni20
vni30
vrf1
vrf2
vrf3
vni1
vni2
vni3
vni4
L3 connectivity
SW
TORS
SW
VM
VM
VM
BM
BM
BM
VM
VM
VM

What’s next?



Tunnel instead of etherchannel

Tunnel should be ECMP friendly

Questions?

I can be reached at tmk@ebay.com