Navigating the Upgrade Maze: A Tale of Modernising Router Software Versions

Jarryd Sullivan – National Engineering Manager

OPTICOMM Networks

A Uniti Group Company

Introduction

Who am I?

- Systems Engineer + Network Engineer for ~15 years
- Currently the National Engineering Manager at Opticomm
- Previously held network engineering roles at Vocus, Aussie Broadband, Over the Wire and DigitalOcean

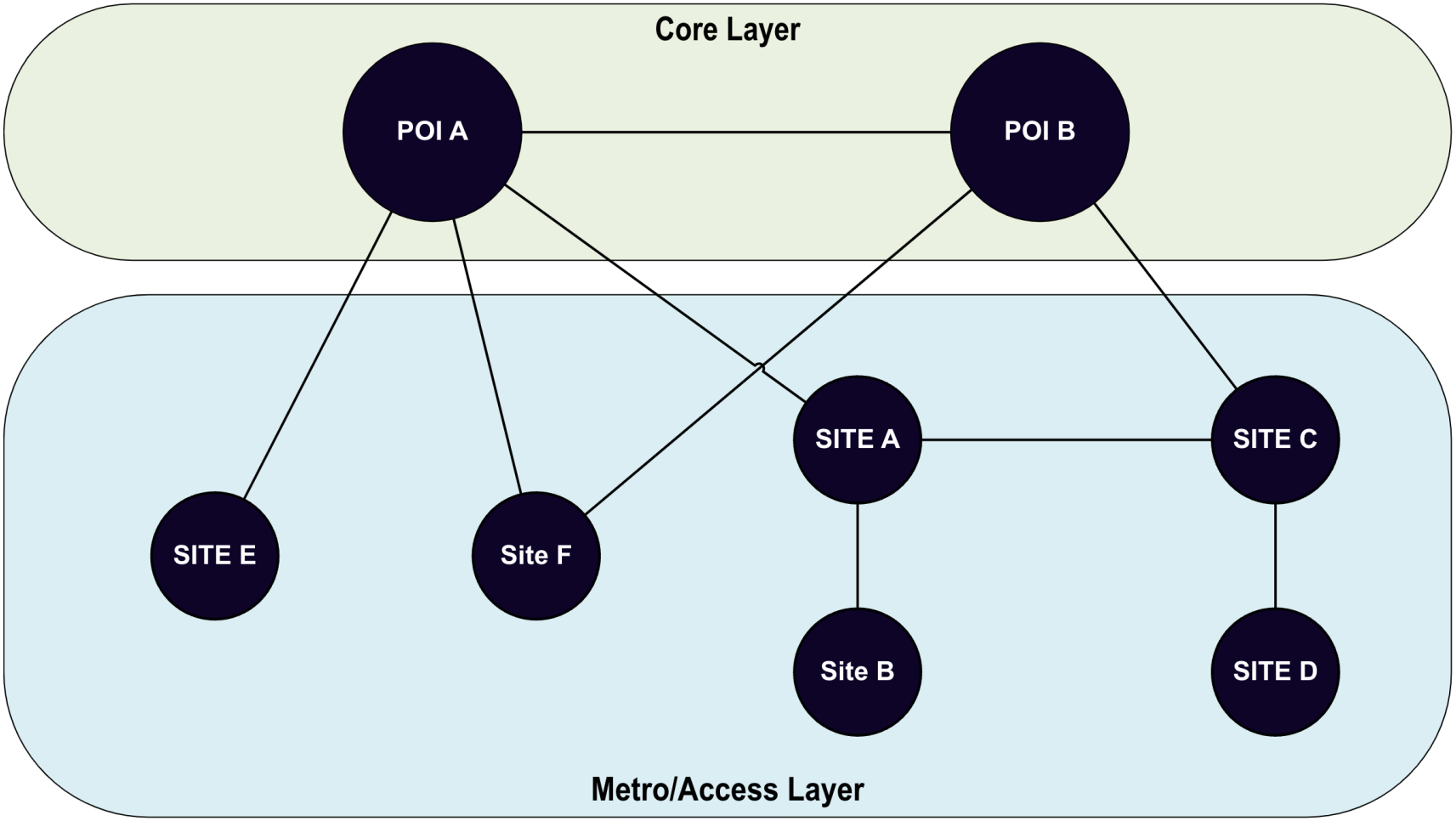
Who are Opticomm?

- One of Australia's largest independent providers of broadband access networks
- We offer a broadband access network like other national providers
- We deliver high-speed internet and communication services to end-user consumers
- We partner with our retail service providers to enable them to connect end-user consumers to the internet

What am I talking about today?

- In July 2023, a configuration change triggered an outage
- Running "extended end of life" (EEOL) software versions – albeit no longer supported
- Known software defects limiting our ability to upgrade
- Managing the process and challenges of upgrading a national access network
- The upgrade process resulted in 100% of the customer base being impacted by a maintenance window during the journey

Our Network


Core Layer
POI A
POI B
SITE A
SITE C
SITE E
Site F
Site B
SITE D
Metro/Access Layer

So, What Happened?

OPTICOMM Networks

25th July 2023

- A routine network configuration change to commission a new Opticomm area is implemented
- Our monitoring system begins to generate alarms indicating widespread outage
- Core network device responsive to SSH, however all cards in the device have stopped forwarding traffic correctly
- All FPCs showing 100% CPU utilization across the device

What do the logs look like?

Jul 25 16:23:37 mib2d[20226]: SNMP_TRAP_LINK_DOWN: ifIndex 2112, ifAdminStatus down(2), ifOperStatus down(2), ifName lsi.1447664
Jul 25 16:23:37 mib2d[20226]: SNMP_TRAP_LINK_DOWN: ifIndex 18288, ifAdminStatus down(2), ifOperStatus down(2), ifName lsi.1178429
Jul 25 16:23:37 mib2d[20226]: SNMP_TRAP_LINK_DOWN: ifIndex 9318, ifAdminStatus down(2), ifOperStatus down(2), ifName lsi.1178942
Jul 25 16:23:39 rpd[17562]: Error creating dynamic logical interface from sub-unit 1222988: Device busy
Jul 25 16:23:39 rpd[17562]: Error creating dynamic logical interface from sub-unit 1222989: Device busy
<cont>

- The logs are far too big to show in their entirety, with over 9700 log messages generated from this single device alone during the issue.

How Did We Respond?

- Roll back the network change
- Implement a network wide change freeze
- We collect the 9700+ lines of log messages and review every single message to look for clues into the problem
- Run common troubleshooting/debug/log collection commands
- Review the configuration change (human error?)
- Investigations yield no obvious signs of human error and configuration is deployed based on our standard site commissioning process
- Implement the same change in a scheduled outage window and take further logs/debugs in the event of another outage for our vendor
- The change goes through without a hitch...no outage, no alarms, no problem *insert puzzled look*

False Sense of Security

- Fast forward to Friday the 13th of October the issue has re-occurred, it had not been seen again since July
- The trigger? A backhaul interface flapping, followed by network alarms indicating a widespread network outage as well as the same log errors present from July
- This time, we took no chances, we immediately engaged our vendor for a P1 support case

What Was Wrong?

- Rewind to May 2023, we had already been working with our vendor on a software upgrade plan to a recommended release

- Limited support on current version due to being EOS

Why is that a challenge for us, you may ask?

- We now have identified a new trigger – one we have no control over, increasing risk and adding time pressures

- From our works with our vendor in May – we would experience several challenges upgrading in a ‘simple’ manner. Why?

Problem Reports
- PR1525226
- PR1528641
- PR1542211
- PR1458825

And so… the journey begins

Bugs and the Path Forward

OPTICOMM Networks

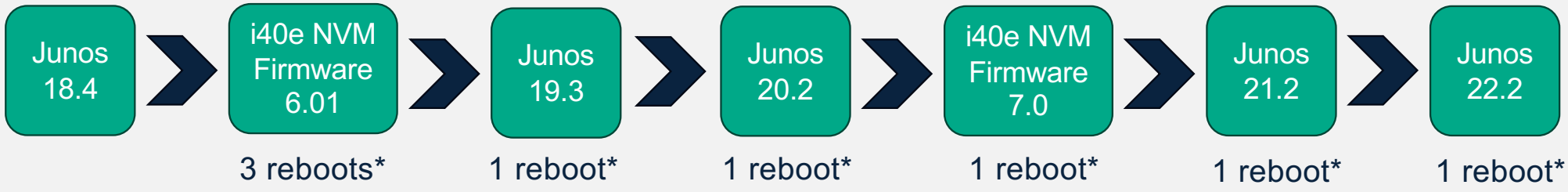
- Our vendor believes we hit either a bug or scaling problem (billions of hardware re-programming event calls during network reconvergence)
- Old code not optimised for managing this scale with our design effectively
- New code optimised to scale better with several known performance improvements
- We start a major project to test, validate and deploy version 22.2R3-Sx
- We were running version 18.4, so we knew this would be a mammoth exercise
- 4 PRs from previous slide had already prevented us upgrading in the past
- First version that all four of the PRs were fixed in was 20.2 and 20.2 would be end of support 30th of December 2023
- Must reach version 21.2 if we wanted to stay on EEOL releases
- Must upgrade from 18.4 to 21.2 in a single upgrade window to avoid hitting these known PRs

Standard Upgrade Path

OPTICOMM Networks

- Download next release, upgrade one RE at a time (noting dual RE systems are in use)
- Must adhere to rules of not skipping certain versions, resulting in a total of 6 upgrades being required
- Must complete at least 6 upgrades in a single maintenance window to avoid 4 known software bugs
- Two i40e NVM firmware upgrades required, with the upgrade to 6.01 requiring three reboots per RE. *Source - https://lkhill.com/juniper-i40e-upgrade/ Special thanks to Lindsay Hill @ Valve (AS32590) for this wonderful blog on how to handle this process*
- Total of 14 reboots required to complete full upgrade cycle
- Total upgrade is approx. 16-18 hours in our lab testing (includes copying software to device)
- Won't fit in a 12am to 6am maintenance window (obviously)

Standard Upgrade Path



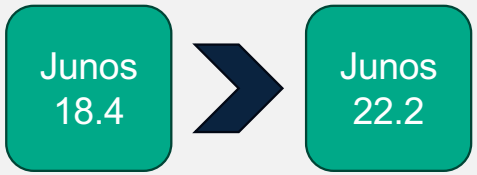
*For single RE system, double for dual REs

Expedited Upgrade Path

OPTICOMM Networks

- Pre-upgrade routing engines in our Melbourne lab
- Ship them to site and schedule window to replace both routing engines (physical swap)
- No requirements to adhere to incremental upgrade path and mitigate risk of hitting 4 PRs in a single step
- All upgrade work is done in a lab environment, with no impact to production until the maintenance window
- Can be rolled back by re-installing the non upgraded REs that were removed
- Two reboots of each RE instead of 14 reboots, and all i40e NVM firmware upgrades completed in lab before shipping equipment to site
- Will fit in a 12am to 6am maintenance window
- “Big Bang” approach, upgrading at least 4 major versions in a single reboot does however present a higher risk

Expedited Upgrade Path



2 reboots in total per RE

Lab the Upgrade

OPTICOMM Networks

- Test the upgrade process in the lab
- Validate the software works as intended and verify the network is stable post upgrade
- Our lab is a replica of our production network, albeit at a smaller scale
- Document the full upgrade process
- Test how long the upgrade takes to understand production maintenance window requirements
- Test i40e NVM firmware upgrades being done in this manner before swapping RE’s to another device
- Ensure there are no configuration changes, syntax changes, or default behavior changes between major software releases

Lab the Upgrade

OPTICOMM Networks

- Document any breaking between software releases to form part of the change process
- Build a list of validation commands for pre/post checks
- Bug scrub and validate for any longer-term memory leaks (leave lab running for several months)
- Reboot devices to ensure no loss of configuration/boot issues

Deployment Plan

OPTICOMM Networks

- Factor in logistics of shipping equipment around Australia
- Document our implementation plan, test plan, rollback plan and then peer review this via our change process
- Notify all our RSPs of the upcoming network maintenance
- Pre book our field engineers to assist with the physical works and validate no issues preventing the device being upgraded
- Mitigate risk of managing large volume of support requests in the event of unexpected issues (bolster NOC shift staff)
- For our first upgrade, we had JTAC on a call with us
- Prepare config backups, do pre checks, implement the change, run post checks, monitor the network and hand over to NOC shift

Upgrade Timeline

It worked!

Lessons Learned/Recommendations

- Shipping equipment around Australia is challenging – cost, reliability, distance/time, insurance
- Build a lifecycle plan for your network and stick to it
- Make use of a lab that's a replica of your production network, even if only at a small scale
- Regularly test new software versions in the lab
- Build a good change management process – reliable change implementation/troubleshooting when it goes wrong
- Review vendor documentation / bug information – it's sometimes useful

Lessons Learned/Recommendations

OPTICOMM Networks

- If you identify bugs/PRs don't forget about them, document it impacts your network, plan to roll out a fixed version as soon as possible
- Don't be afraid to implement a change freeze to maintain the stability of the network until you understand a problem better – but most of all communicate this with your customers
- In hindsight, USB upgrading these REs would have been another method that could have been used (potentially) but has not been tested by us
- Build a reliable OOB network
- Buy your field engineers and network engineers lots of red bull or coffee… they deserve it

Implementation Plan

OPTICOMM Networks

Upgrade Implementation Steps

- Back up primary and secondary RE configurations, SCP them to a safe location for later use
- Using a pre-defined list of commands, we'd complete pre-checks on the devices documenting the output of each (Special thanks to Pat @ AS7575 for providing recommendations on extra pre-checks we should run based on his experience completing many Junos upgrades of a similar nature)
- Take a "RSI" (*request support information*) pre upgrading, in the event we need it for JTAC cases
- Deactivate GRES + NSR
- Deactivate commit sync to prevent RE's attempting to sync commits between each other
- Gracefully power off non-active RE
- Have our field technician remove the RE and install a pre-upgraded one
- Once the RE has booted and been up for approx. 5 minutes we log into the RE and load the RE configuration from disk, before we commit the change, we deactivate NSR/GRES and commit sync as you'll remember we took the config backup before disabling them earlier
- We confirm the config loads as expected with no issues as per our lab testing and then we reboot this RE again using the "request vmhost reboot <re0|re1> ensuring you choose the non-active/standby RE. This is to ensure the "enhanced-ip" configuration applies correctly. Do not miss this step. Once the RE has booted back up and been online for at least 5 minutes we continue with the following steps.
- Move back to the active RE and check if the device is ready to complete the RE switchover by running the following command: "*request chassis routing-engine master switch check*" which should return the following expected output:
  > *warning: Traffic will be interrupted while the PFE is re-initialized*
  > *Standby Routing Engine is not ready for graceful switchover.*
- Up until this point none of this work has been service impacting, the next steps are where the first outage occurs due to the PFE re-initializing and completing any required microcode upgrades
- On the active RE run the following command: "*request chassis routing-engine master switch*" to complete an RE switchover. At this point, the PFE's/FPC's will reboot as expected
- Complete the same steps to replace the now standby RE running old code with the other pre-upgraded one, in the meantime the FPC's will boot and start forwarding traffic again after approx. 10-15 minutes

Implementation Plan

OPTICOMM Networks

Upgrade Implementation Steps Cont.

- Once the second upgraded RE has been re-installed into the system, wait for it to be detected using the “show chassis routing-engine” commands and ensure the uptime is at least 5 minutes
- Follow the same process again to load the correct configuration backup onto the RE you just installed
- Ensure you disable GRES/NSR and commit sync before you commit the change again
- Commit the configuration and then reboot the RE again using the same vmhost reboot command as before, again to ensure ”enhanced-ip” configuration takes effect
- Once the RE has booted back up with an uptime of at least 5 minutes you can log into the active/primary RE and re-enable commit sync, NSR and GRES and commit the configuration.
- On small scale networks it takes approx. 5 minutes for NSR and GRES to be completely ready, sometimes even quicker. On large scale networks please do allow at least 10-15 minutes and you should always run the following commands to check the device is ready to switchover:

> request chassis routing-engine master switch check
Switchover Ready
> show task replication
Stateful Replication: Enabled
RE mode: Master

Protocol	*Synchronization Status*
OSPF	*Complete*
MPLS	*Complete*
RSVP	*Complete*
LDP	*Complete*

- If any of the above commands show they’re not ready/complete, then do not proceed any further until they do.
- Once they all show complete/ready you can do a graceful routing engine switchover using the following command: *request chassis routing-engine master switch*

If everything has gone well, you should see no traffic loss. As always, with any process, ensure the above meet your specific needs, there may be commands that are relevant to your network that I have not included here, always reach out to your network vendor if you’re unsure about the process for your platform.

Questions?