The Journey to a Fully Automated IP Remote Production Network

Case Study: NEP Australia

Broadcast network transition to IP is in full flight. There are increasingly more new ST-2110 networks on-air every few months. Some have been on-air for nearly two years now, such as NEP Australia.

NEP Australia commenced trials with ST-2110 in October 2017 as they developed their remote production facilities in Sydney and Melbourne. It was early days with the technology and the skills of engineers at the time were quite discrete. There were broadcast engineers and network engineers. Learning curves on both sides were steep, but there was joint success through a collaborative approach to the project.

As the OB trucks were fitted out for IP and the Hub data centres and production facilities were completed, NEP was on-air with live production entirely based on IP from the Sydney Hub by March 2018.

NEP Australia now broadcasts over 800 productions every year from 39 venues around Australia, all over IP and all remotely produced from the Sydney and Melbourne Hubs.

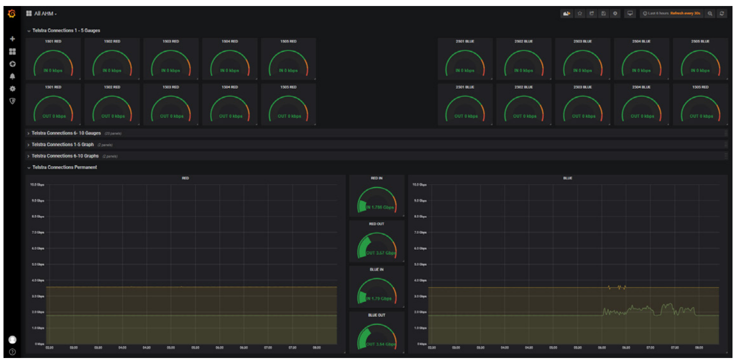
What about the network operations side of this innovative installation? How has this evolved since October 2017 and what has NEP done to develop a new paradigm around managing an IP based broadcast network?

The answer is, that it has been incredible to witness the emergence of new hybrid engineers combing skill sets from not two, but three disciplines: Broadcast, Network and Development. By leveraging these cross-purpose skills, NEP Australia has gone from basic network configuration scripts and traditional network monitoring tool sets, to innovating with modern automation models and real-time streaming telemetry, in a very short amount of time.

The aim of this paper is to share these learnings and highlight the benefits of a modern approach to managing broadcast networks.

Phase 1: Traditional Tool Sets & Early Automation

As productions went live from the Hub, the network was monitored and managed using a combination of traditional network management tools sets and command line interface (CLI) configuration.



The traditional approach to network management has been using SNMP (Simple Network Management Protocol). An SNMP solution includes a management server, MIB (Management Information Base) definition files, and end networks devices. The SNMP server polls the network devices at a set interval and extracts all the state from the device as defined by the MIB files. The state extracted is only the elements defined in the MIB and includes elements that have not changed or are zero. This is all collected on every polling interval, which is typically every 30 seconds to 2 minutes. This information is then graphed in a tool and displayed in what can only be described as smoothed averages of what is actually going on in the network, with very little granularity.

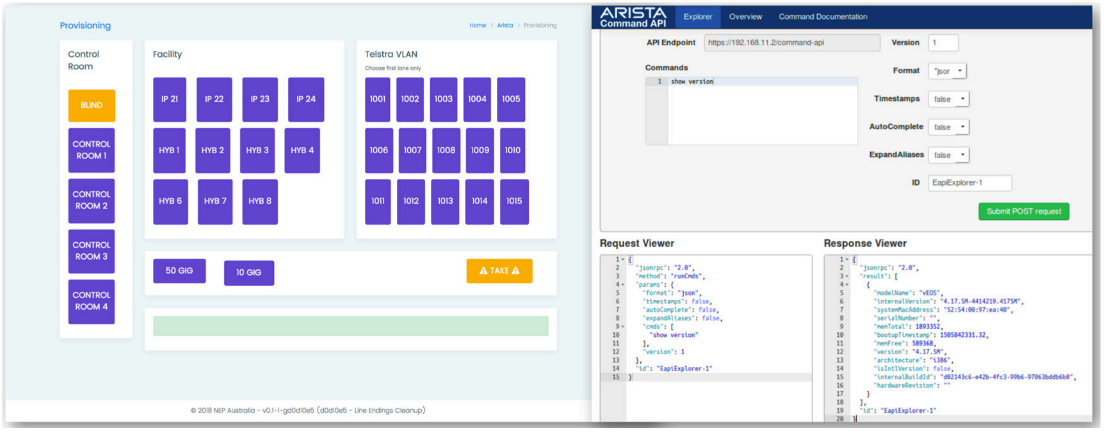
This approach includes the following disadvantages as you can probably imagine:

- CPU cycles consumed on every interval, on every device
- Network traffic generated across the entire network, on every interval
- What about state changes between the polling intervals? These are lost or delayed.
- I'm running a broadcast network, where I need to know what's going on within my network at the sub-second level, not the average over 2-5 minutes of data collected every 30 seconds or so.

This approach for NEP was enough to establish a base level of visibility of the network, but was clearly not that useful for troubleshooting during a production.

When it came to configuration ahead of a live production, there was network configuration changes to make based on which OB truck was used, what venue was hosting the event, and which Hub was in production for the event. One of the goals of introducing IP was to reduce the number of personnel at an event, not increase by now adding a network engineer. CLI configuration was initially used for these changes required, but it was clear that adding automation to this provisioning process was going to make their lives a whole lot easier and remove the human error element.

Arista EOS features a JSON-RPC API (eAPI) through which programmatic access to the network devices is provided. NEP was able to leverage eAPI to start automating this provisioning process through the use of scripts. Now when an OB truck came online, the centrally located network engineer was able to execute the scripts necessary to being the truck online ahead of the remote production.



This approach though, was still very network-centric and wasn't particularly user friendly for the broadcast engineers on location.

The next goal was to address the shortcomings of this phase 1 and introduce some modern techniques through Arista CloudVision.

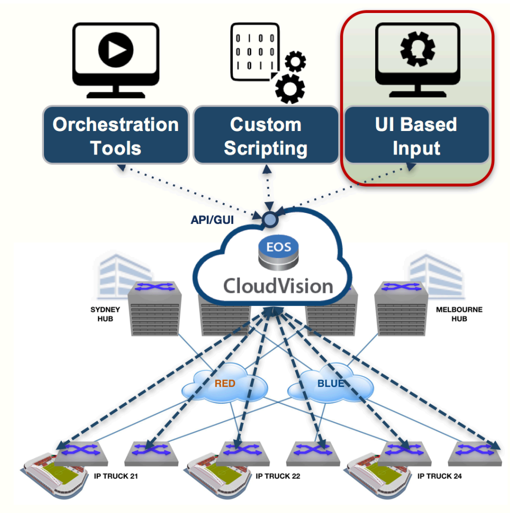
Phase 2: Automation & Visibility through Arista CloudVision

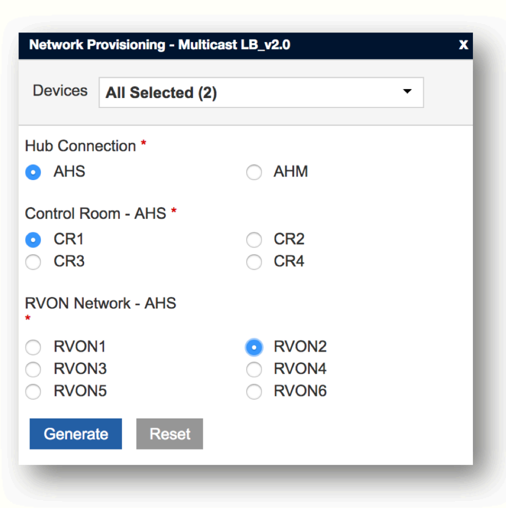
As part of the Arista network deployment, NEP implemented Arista CloudVision. CloudVision is an automation and visibility tool that introduces modern provisioning and telemetry elements, in a turn-key solution.

On the network visibility side, Arista EOS features a real-time state streaming capability available in all Arista switches. EOS features a simple state database whereby all state within the switch is available in a simple tree data structure, called NetDB. The streaming telemetry agent running within EOS subscribes to all state elements and is notified when an element changes. As these state elements change, this information is streamed from the switch to a remote destination. In this case, it is streaming to CloudVision. Contrast this to SNMP. Instead of the management server interrogating every device, for all state, on a set polling interval, the device is instead responsible for sending the state information towards the management server (CloudVision in this case). Not only does this provide real-time updates of state for devices as state changes, but it removes the need to send all the state, all the time. Only the state elements that change are sent and only if they are non-zero. T

his is a very different approach and immediately proved invaluable in the NEP broadcast network. PTP is one of the state elements streamed to CloudVision and so we now understand the entire network path for the PTP domain and understand immediately if there are changes. This information was used to quickly identify grandmaster clock changes which were not expected in the network.

Now armed with increased visibility, NEP looked to leverage some automation for the provisioning of the configuration changes required when an OB truck came online from a venue. CloudVision features the ability to create custom forms used for user input and apply some programmatic logic behind that form to instantiate the correct configuration template for a given device. An example is shown below of a CloudVision Configlet Builder developed by NEP for this provisioning use case. The goal was to provide the broadcast engineers with a simple user interface with a pre-defined set of parameters that could be selected prior to the event.





Configlet Builders within CloudVision are very flexible and the underlying Python based scripting engine provides a very powerful capability. These Configlet Builders may be executed from the form-based user interface or programmatically from an external source calling the REST API available for all CloudVision functions.

The NEP operation was starting to realise some of the benefits from these modern day telemetry and provisioning techniques. The skills available within the NEP Australia team were starting to evolve and the broadcast engineering and the network engineering were creating synergies. The third element NEP were fortunate to have access to through in-house skills available was a broadcast engineer with a software development background.

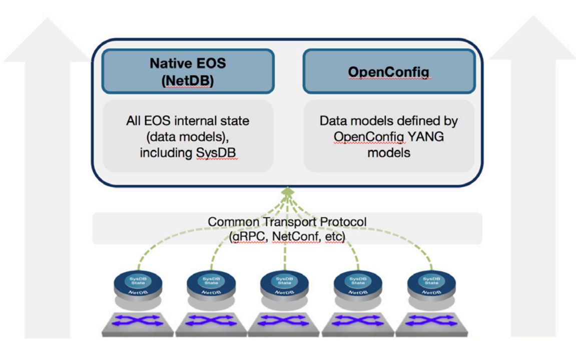
NEP had the vision of taking this streaming telemetry and automation capability within the Arista platform and extend this to create a closed loop for provisioning the broadcast network services ahead of an event.

Enter Openconfig and Phase 3.

Phase 3: Customised Open Source Control

One of NEP's goals in developing an in-house solution was to create something that would take advantage of open standards to ensure that the investment in time and resources would not lock them into something proprietary.

Arista EOS and CloudVision are built upon open standards. While all state can be sent to CloudVision to take advantage of the turnkey analytics engine for telemetry, the state from EOS may also be streamed to third party or open source tools. This state may be represented in Openconfig data models or native EOS data formatted in a standardised way. For example, the state could be streaming to a Kafka bus, which in term could be used to message to another upstream tool.



For NEP, the use case was, in addition to streaming the telemetry data to CloudVision for a holistic view of the network, also send selected state from the devices to an open source tool using Openconfig models. This tool would consume the state in real-time, just as CloudVision does, and understand the current utilisation of links from the OB truck towards the Hub across the telco network.

This information would be displayed in a real-time graph and be used as a means to determine the path onto which to provision multicast flows, based on the bandwidth available. The provisioning is automated through an API call to the Arista switch via eAPI.

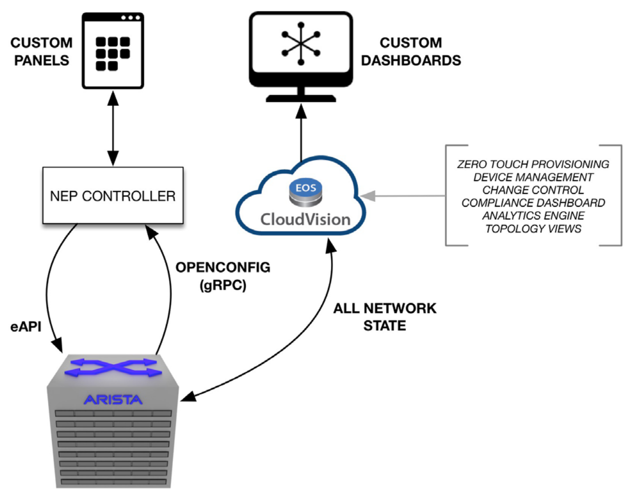
So now NEP has a closed loop of understanding current network state and automating the provisioning of flows based on the active utilisation. The even nicer element is NEP has developed a simple frontend, based on open source, that presents the broadcast engineer with a touch panel which gives them all the information and control they need to stand up the network connections for the OB truck before the event.

A Deeper Look into the Openconfig Workflow and Benefits for NEP

The primary benefits gained by NEP in developing this system based on Openconfig included:

- Scalable real-time monitoring
- Faster decision making for automation
- Less overhead with lighter collectors of the network state
- Forward compatibility and vendor interoperability
- User defined models for the network state
- To observe the network state as part of the provisioning process

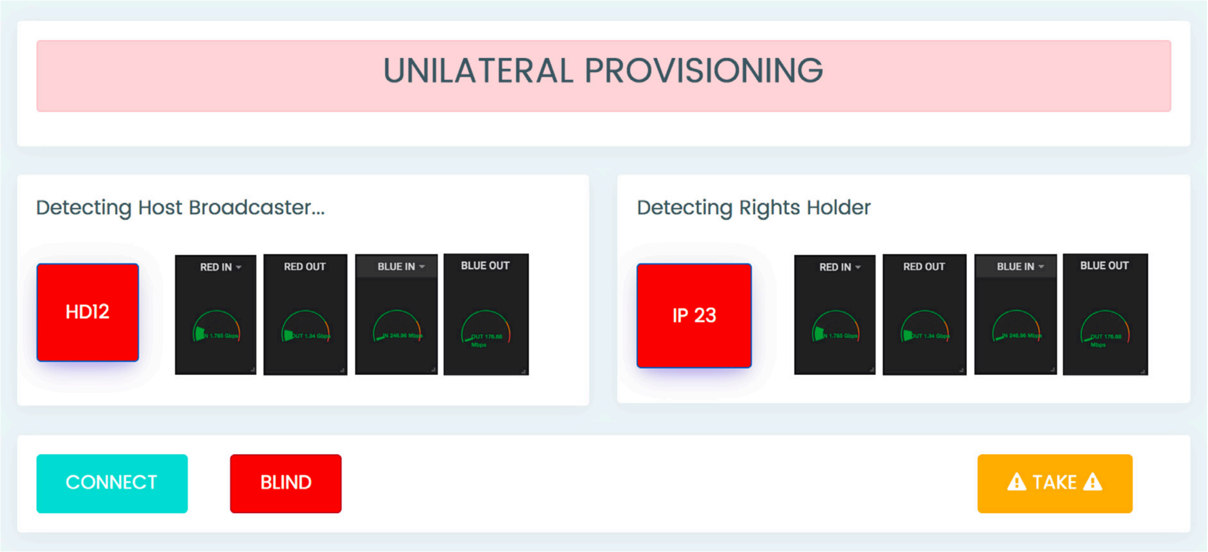
The mechanism used to connect Arista EOS to the NEP control system is through the open gRPC (Google Remote Procedure Call) protocol. This transport provides for serialisation of structured data in what's known as a Protobuf (Protocol Buffer) data transmission. The gRPC connection is established using HTTP/2 transport layer and supports a bidirectional communication model, load-balancing, flow-control and call-cancelling.



The management protocol over which the streaming telemetry and configuration data is carried over is called gNMI (gRPC Network Management Interface). gNMI leverages the gRPC framework and is implemented in Golang (an open source programming language) on Arista EOS through the gNMI Library available on the Arista Network github site (https://github.com/aristanetworks).

There are various open source tools available to receive the streaming telemetry data in Openconfig form. NEP leverages a combination of Prometheus and Grafana for consuming the telemetry data and making it available in graphical form. Prometheus provides a time series database and Arista provides a client (ocprometheus) to enable Openconfig data from EOS devices to be consumed by Prometheus. These elements allow for integration with monitoring and alerting tools such as Slack.

This has allowed NEP to extend the provisioning process further through enabling Unilateral Provisioning with a customised interface as shown below.



Summary

The richness and openness of Arista EOS and CloudVision has helped NEP to take some operational challenges, and in a very short amount of time, extend this capability to realise some real business value.

The primary benefits to NEP, by choosing to invest time and resources on developing these systems with Arista and Openconfig, are summarised as follows.

- Greater visibility and control over the network - resulting in a simpler operational model with less room for human error
- Increased granularity and speed of provisioning
- Real-time alarming of events occurring within the network - minimising the delay in troubleshooting an issue
- An opportunity to take a positive step forward in integrating broadcast control with network control that has resulted in real business benefits

Santa Clara—Corporate Headquarters
5453 Great America Parkway,
Santa Clara, CA 95054

Phone: +1-408-547-5500
Fax: +1-408-538-8920
Email: info@arista.com

Ireland—International Headquarters
3130 Atlantic Avenue
Westpark Business Campus
Shannon, Co. Clare
Ireland

Vancouver—R&D Office
9200 Glenlyon Pkwy, Unit 300
Burnaby, British Columbia
Canada V5J 5J8

San Francisco—R&D and Sales Office
1390 Market Street, Suite 800
San Francisco, CA 94102

India—R&D Office
Global Tech Park, Tower A & B, 11th Floor
Marathahalli Outer Ring Road
Devarabeesanahalli Village, Varthur Hobli
Bangalore, India 560103

Singapore—APAC Administrative Office
9 Temasek Boulevard
#29-01, Suntec Tower Two
Singapore 038989

Nashua—R&D Office
10 Tara Boulevard
Nashua, NH 03062