Multi-Vendor NETCONF/YANG-Based SDN Management Interoperability Test

White Paper

Table of Contents

Editor's Note

The EANTC team is proud to release detailed results of our second NETCONF/YANG multi-vendor interoperability testing campaign. The leading manufacturers Cisco, Ciena, IP Infusion, and Nokia participated in the in-depth tests in April 2021, thoroughly verifying the readiness of their orchestrator/controller and router solutions to interoperate with each other using NETCONF as the management protocol and standardized YANG models (as much as available today).

We focused on service provider transport use cases: Device provisioning, Layer 2 and Layer 3 EVPN service creation and teardown, service teardown, monitoring, and service reliability.

What are the major takeaways?

The participating orchestrators and controllers impressed us with their efficiency and readiness for multi-vendor provisioning. Even in first-time vendor equipment combinations which had never been tested before, it took only less than a week for the participants to successfully provision advanced services together. This is due to a combination of positive developments:

- First, standardized, model-driven interfaces like NETCONF are a real success story. While NETCONF is not alone (there are some similar alternatives), we have covered NETCONF only in this event: We have witnessed the principles of separation of configuration and state, transactionality, idempotency, and the availability of existing YANG tools. Together, they form a strong foundation on top of which interoperability can be reached efficiently.
- Second, standardized high-level YANG models are becoming more mature. The OpenConfig initiative has further improved their models, eliminating some of the common areas of misinterpretation. It is now more straightforward to use OpenConfig models, although they still need to be augmented and individually adapted to achieve interoperability for non-trivial scenarios. In a number of scenarios, the use of proprietary YANG models proved to be the more pragmatic approach.
- Third, the orchestrator teams of Cisco and Nokia were very experienced both with multi-vendor scenarios and with the adaptation to new use cases. At EANTC, we sincerely appreciate the vendor experts' commitment that has greatly contributed to the success of this event. It shows that multi-vendor management integration is the new normal for leading orchestrators and that it is possible to adapt to new scenarios efficiently now.
- The participating vendors either already had extensive experience with NETCONF/YANG support or a strong commitment to standards-based network management.

Traditionally, vendor implementations are retrofitted to standard YANG models: First, the proprietary implementation existed with its own unique internal architecture. Then, YANG model support was added, facilitating some sort of internal translation. This may be a difficult undertaking with a lot of heuristics or very complex formal mathematical mapping. (I remember the issues vividly from the information theory lectures at university.) In some cases, parts of the implementation architecture were designed entirely around YANG models, using a best-of-breed approach from OpenConfig, IETF, and other sources.

We witnessed that this approach accelerated multi-vendor interoperability. Another participating vendor's setup was very closely aligned with the published OpenConfig specifications - which gained extra bonus points for standards compliance, but came at the cost of limited coverage.

Please note that the very positive results were achieved with a small number of vendors only. Support of NETCONF/YANG cannot be assumed across the whole industry yet. It is an area of active development in many areas – not only network transport and services but also broadband access and Open RAN, just to name a few.

We hope that our interoperability testing program will encourage more service providers to add NETCONF/YANG requirements to their requests for proposals (RFPs). More specifically, we recommend requesting specific standardized YANG model support in RFPs. While NETCONF is a protocol well suited for network management, there are other viable standardized options available in the market as well, such as RESTCONF or RESTful.

What was the major focus of this test campaign?

Our high-level technical goal for this interoperability event was to get closer to typical service provider automation scenarios. Beyond the basic device configuration models, we focused primarily on YANG models for Layer 2 and Layer 3 VPN services, the latter standardized by the OpenConfig initiative. We covered small-scale multiple service instance creations as well – as an initial step towards verifying the implementations' scale and robustness. In real-life deployments, the ad-hoc creation and tear-down of multiple service instances with different configs will be important.

How have we conducted the tests?

In February and March, the participants and EANTC created the detailed test plan together in preparation for the event.

The actual tests took only three weeks including all the logistics of remote connectivity, pre-staging, and formal testing. The test campaign was fully remote and distributed. Encouraged by the CoVID-19 situation, all participating vendors collaborated online across 12 time zones from California to India. EANTC hosted virtualized workloads in our internal VMware Integrated OpenStack cloud in collaboration with VMware. Hardware devices were connected via IPsec site-to-site tunnels, and remote client access was provided by OpenVPN connections. Collaboration tools such as Confluence, Rocket.Chat, and NextCloud facilitated the documentation, chat, and file exchange services. Our team used an extensive EANTC-built internal test results submission tool, enabling very efficient test execution and review workflows.

Why are we doing these tests?

We hope that our tests:

- accelerate interoperability of participating implementations;
- help to resolve any structural interoperability issues and advance protocol standards;
- encourage learning and more advanced proof of concept (PoC) testing at service provider labs, avoiding repetition of basics.

EANTC provides a vendor-neutral space to integrate solutions. Our interoperability tests provide a meeting place where new implementation aspects can be trialed with other manufacturers – many of whom may not be close partners and whose latest software versions may not be available otherwise. Vendors can learn about other implementations and can discuss technical innovations with their industry counterparts. Through our detailed test plan guidance and verification of test scenarios and results, EANTC helps to produce tangible, reproducible results published for service providers to understand the state of the market.

What are our plans for the future?

The outcome of this event encourages us to continue the series of NETCONF/YANG interoperability events – striving for more advanced scenarios and participation of additional vendors. Our future plans include orchestration aspects such as atomicity of operations, reproducibility of configuration and de-provisioning activities, exact validation of device behavior for advanced service use case scenarios, overlapping service instances, and testing of notifications. Stay tuned!

Introduction

Aim high. Reach for a new transition of SDN/NFV driven 5G network. Soon, we know that automation technology is integrated into all aspects of business today and is vital in helping service providers and operators to achieve and maintain a competitive advantage. NETCONF/YANG, through its independent configuration approach in network automation and programmability of data models, shall align itself with business strategy and deliver a clear value in solving business problems. It is critical that individual contributions in building roles of network management solutions are proficient in complying to open interfaces and supporting business goals.

In this test, we verified the interoperability between SDN controllers and network devices in a multi-vendor environment represented by Ciena, Cisco, IP Infusion, and Nokia. We selected L2 and L3 end-to-end VPN use cases with protection functions based on best practices providing add-on values to core and enterprise data centers, especially based on NETCONF key areas identified by the Telecom Infra Project. Additionally, we looked at precision time protocol (PTP) configurations with proprietary models. We did not perform any tests in optical and microwave areas as there was no equipment participation. We verified various provisioned services against YANG models according to the latest IETF RFC8299, and IETF RFC8466 service models and OpenConfig definitions to achieve zero-touch automation during the one-week test execution.

We successfully validated 28 test combinations, based on larger topologies to simplify the number of combinations, especially through hierarchical orchestration with the modular provisioning of multi-services. All results were expected and were based on OpenConfig models obtained by Ciena, Cisco, IP Infusion, and Nokia. One output generated more configuration caused by translation of NETCONF to CLI on the device under test but did not affect the test case. Otherwise, this was the only issue during the test that should be addressed by network device implementation. We observed appropriate YANG support for OpenConfig YANG models in some devices for service provisioning, monitoring, and BFD settings.

The use of native OpenConfig YANG models resulted in low test coverage. Most participating vendors implemented OpenConfig YANG extensions to increase the level of interoperability. Typically, the controller vendors are still required to use deviations and augmentations, to individually adapt the current high-level public definitions from OpenConfig to individual device implementations. As an example, the interface packet counter of OpenConfig includes the rate counter definition without any detailed information such as packet sizes. For protocol-compliant participants, a basic counter without any packet sizes was sufficient for this test—but it would not be very useful for traffic statistics.

Several devices had trouble maintaining the precision of incoming NETCONF configuration changes. In many cases, this imprecision could be worked around by letting the controller explicitly specify the value of additional YANG elements that would otherwise get a suitable default value, but get reported as ordered by the client, while they were not. With one device, we found that the device's configuration turned out different than how it was ordered through NETCONF. Generally, NETCONF automation requires that the configuration applied by the device is applied precisely, meaning that the device applies the required configuration and does not add or remove any other commands to the configuration.

If a NETCONF implementation is done as a layer on top of an existing CLI, the design makes it hard not to transfer CLI-centric behavior into the automation-centric NETCONF layer. While CLI-centric behavior, such as automatically adding configuration on behalf of the user, is best practice in the CLI-world, it may differ to the automation use case and further add complexity to participating in high levels of automation.

According to the explanation of NETCONF experts, the above-mentioned loss of configuration precision is a common problem in NETCONF designs. It is also a well-known problem to developers who implement one YANG model set, such as OpenConfig, on top of another model set, such as a proprietary/native YANG model set. EANTC performed a check/training before the actual test. The engineers of the involved system under test with the partner for each configuration had to find the extra commands applied by the system and decided manually whether to ignore them or not. These changes did not have any impact on the test results for this use case. For more advanced use cases, and during prolonged use of the system, deviations between intended and actual configuration might cause unexpected outcomes.

Finally, to manipulate configuration data for a large network and many service instances, management automation requires fast response to read and write operations. Maintaining stability by strict configuration control will help.

Test Setup and Measurement Equipment

EANTC was responsible for setting up the test environment to be used during the test, which provides the connectivity between the four vendors and their labs.

EANTC installed the virtualized IP Infusion OcNOS router components, Nokia NSP Controller components, and virtualized Cisco NSO components at EANTC's lab. Ciena routers under test were located at Ciena's lab. EANTC created and configured IPsec tunnels between the EANTC's firewall and the test sites as part of the test setup.

In collaboration with VMware, EANTC provided a VIO platform (VMware Integrated OpenStack) to host Cisco and Nokia's controllers and orchestrators. The platform has sufficient resources to host much more large-scale workloads.

EANTC created remote access accounts for all participants to reach their VMs in EANTC's lab to configure their devices and run the tests. We used a Wiki collaboration space to plan the test and document all results. EANTC provided the participants with links for digital meeting rooms to meet, discuss, and run the tests. The collective work requires technical abilities and the ability to adapt the work time and routines to reach the work goals. All participants were brilliantly able to adapt to one work-time period, which allowed this wonderful collaboration experience.

Vendor Name	Device Name	Device Role	Software/ Firmware Version
Ciena	5164	Router	10.6
Cisco	Network Services Orchestrator (NSO)	Controller and Orchestrator	5.4.0.2
	IOS XRv 9000 (XRv9k)	Virtual Router	7.3.1
IP Infusion	OcNOS	Virtual Router	4.2.56a
Nokia	Network Services Platform (NSP)	Controller and Orchestrator	20.13 (GA)

Table 1: Software and Hardware Details

L3VPN Service Provision

Network automation moves rapidly towards a service-oriented approach with network management, where many different systems support complex services. Interest in configuration control grows from managing equipment towards a situation where an operator is actively managing the various aspects of services. IETF (Internet Engineering Task Force) defined an L3VPN YANG service model in RFC8299 in 2018, which was used in the interop event to communicate between customers and network operators and deliver a Layer 3 provider-provisioned VPN service.

We verified the interoperability of the controller to render northbound and southbound interfaces and database schemas from the service and device model. Northbound was the APIs published to the orchestrator, based on the YANG service model as defined in RFC8299. Southbound was the integration point of managed devices; both proprietary/native and OpenConfig YANG models were part of the test. We observed the capability at the orchestrator and controller with the defined YANG model to configure the network as a whole rather than individual devices.

NETCONF offers support for transactional network management. We defined three different L3VPN service instances to verify the creation and deletion provisioning as a whole of each individual service across multiple network elements. A comfortable part of that built up on repeatability, dynamically adapting the service configuration solutions according to variation in the service definition without defining low-level device configuration commands. The precise goal in science was rather a verification of data synchronization to keep the service and service model synchronized. A common problem with a script-based network automation pipeline is when tearing down a service, the script does not know how to clean up the configuration data on a device when the service instances are overlapping. It is also a well-known problem by adding any service, a large amount of glue code is needed. Extreme examples involve whether to delete some shared configurations before adding new services. This test determines whether NETCONF elements could help to eliminate the device integration problem and provide a service configuration solution utilizing automatically integrated devices.

We started with an empty configuration (without any L3VPN services) as a baseline in the network and saved the configuration from each network device.

We sent traffic for the L3VPN services and expected a 100% drop for the current stage, indicating that none of the services existed. To verify the hierarchical setup with an orchestrator, we observed the required VPN information (e.g., service name, sites, IP addresses) filled in the service instances at the orchestrator's front-end, ready to initiate the service creation towards the controller via its northbound interface. We also checked the status of the controller and the YANG model at the southbound towards the network device. This NETCONF file was significantly bigger after filtering the value of key parameters. We started the service from the orchestrator in case of a hierarchical orchestration setup. In case of a direct controller setup, we started the service creation from the controller. The creation steps included single or two services at the same time until all three services were completely provisioned. In each step, we derived the running configuration from the device through the controller. We used the configuration initiated by the NETCONF as a baseline to compare the differences between both. We did not expect any changes to the running configuration indicating that the device completed the operation exactly as required by NETCONF.

The traffic passed through for each newly created service without any loss indicating a successful service creation. Especially any service creation or removal should not impact the traffic of any other existing service indicating a graceful service provision. While traffic was running for all three L3VPN services, we randomly chose to simultaneously delete all the services, a single one or two services. The traffic should be dropped for the deleted service instance(s), indicating a successful deletion. A successful deletion should not have any impact on the traffic of existing services, even if the service instances share some device level configuration. After all L3VPN services were removed from the network, we saved the configuration from the network devices and compared it with the baseline configuration as saved at the beginning. We did not expect or observe any changes in the configuration.

The following systems successfully participated in the hierarchical orchestration test: Nokia NSP acted as orchestrator, Cisco NSO as controller, and Ciena 5164 as PE router. The transport network was based on the MPLS Segment Routing (SR-MPLS).

The following systems successfully participated in the service provisioning with controllers: Nokia NSP acted as controller, Cisco XRv9k as virtual PE router. The transport network was based on the MPLS with ISIS.

Cisco NSO and Nokia NSP as controller, and IP Infusion OcNOS as virtual PE router. The transport network was based on the MPLS with OSPF.

One running configuration differed from the expected NETCONF configuration after the test and showed an additional interface configuration spontaneously added by the device.

The vendor's engineer checked with the partner each configuration manually to identify the extra commands to exclude the impact on the L3VPN services. One virtual solution did not support forwarding for VPN services. We verified the BGP VRF routing table via CLI on the DUT to ensure that all VPN routes have been learned. In addition, we observed the capture taken from the virtual link between both virtual PEs to ensure the L3VPN family type in the BGP update messages.

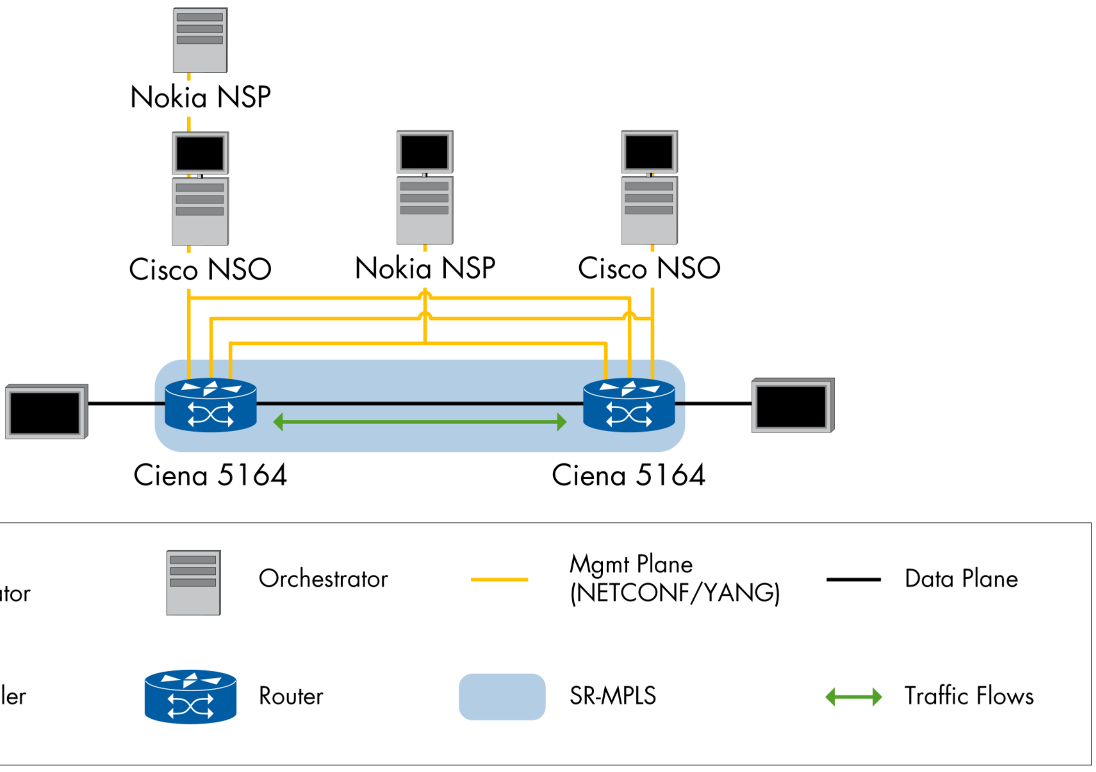

Figure 1: L3VPN Service Provision—Test Setup 1

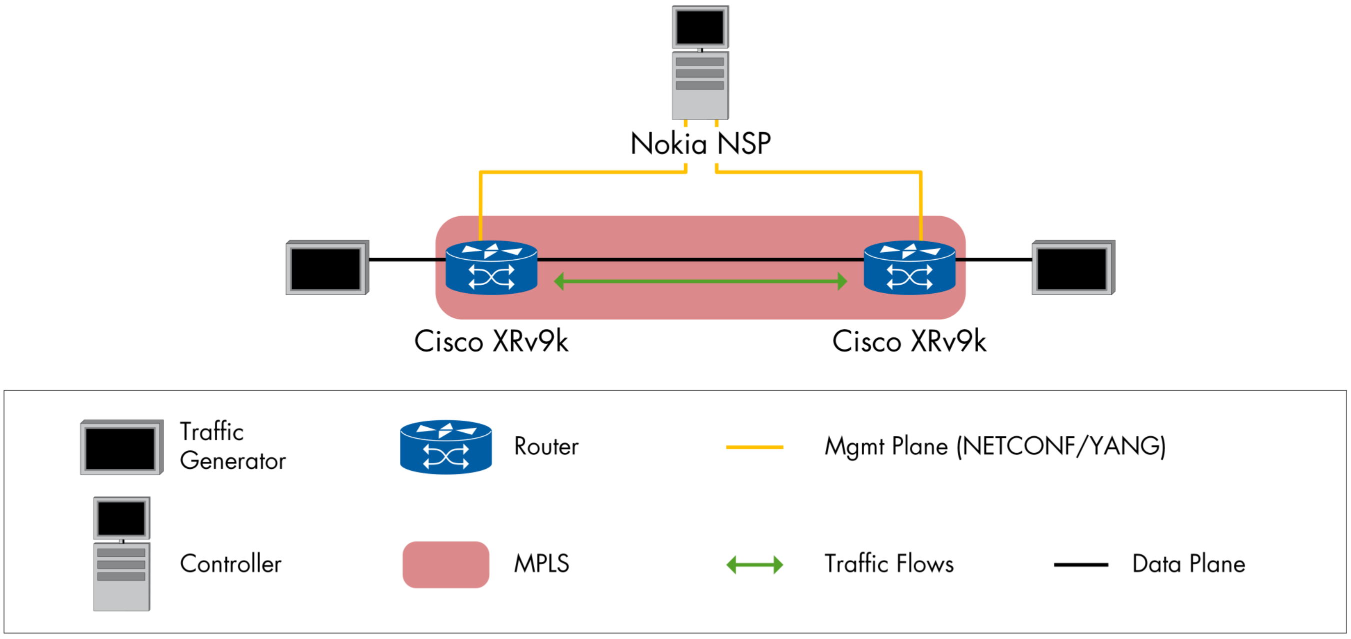

Figure 2: L3VPN Service Provision—Test Setup 2

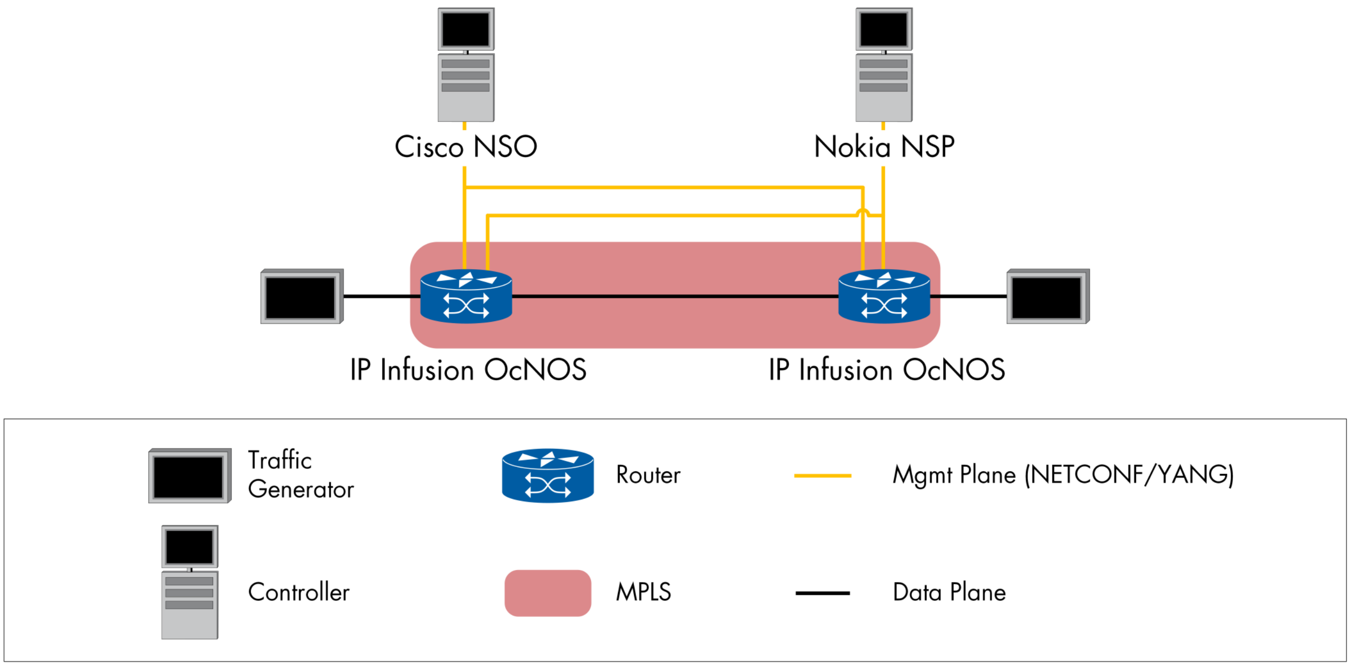

Figure 3: L3VPN Service Provision—Test Setup 3

YANG Models

We covered the following YANG models:

Vendor	Model and Path
Cisco	**NSO > Ciena 5164** Device model: /vrf in ciena-vrf.yang /classifiers/classifier in ciena-mef-classifier.yang /fds/fd in ciena-mef-fd.yang /fps/fp in ciena-mef-fp.yang /logical-ports/logical-port in ciena-mef-logical-port.yang /interfaces/interface in openconfig-interfaces.yang /bgp/instance in ciena-bgp.yang **NSO > IP Infusion OcNOS** Service model: /l3vpn-svc/vpn-services/vpn-service in ietf-l3vpn-svc.yang /l3vpn-svc/sites/site in ietf-l3vpn-svc.yang Device model: /network-instances/network-instance in openconfig-network-instance.yang /interfaces/interface in openconfig-interfaces.yang
Nokia	**NSP > Cisco NSO** Service model: l3vpn-svc (rfc8299) XMLNS: urn:ietf:params:xml:ns:yang:ietf-l3vpn-svc Paths: /l3vpn-svc/vpn-services/vpn-service /l3vpn-svc/sites/site

Vendor	Model and Path
Nokia	**NSP > Ciena 5164** A combination of Ciena and OpenConfig models have been used. Device model: vrf | xmlns="urn:ciena:params:xml:ns:yang:ciena:ciena-vrf" classifiers | xmlns="urn:ciena:params:xml:ns:yang:ciena-pn::ciena-mef-classifier" fds | xmlns="urn:ciena:params:xml:ns:yang:ciena-pn:ciena-mef-fd" fps | xmlns="urn:ciena:params:xml:ns:yang:ciena-pn:ciena-mef-fp" logical-ports | xmlns="urn:ciena:params:xml:ns:yang:ciena-pn::ciena-mef-logical-port" interfaces | xmlns="http://openconfig.net/yang/interfaces" bgp | xmlns="http://ciena.com/ns/yang/ciena-bgp" Paths: /vrf /classifiers/classifier /fds/fd /fps/fp /logical-ports/logical-port /interfaces/interface /bgp/instance Note: Ciena used extensions to augment the OpenConfig YANG models. **NSP > Cisco XRv9k** Device model: vrfs | xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-infra-rsi-cfg" bgp | xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-ipv4-bgp-cfg" interface-configurations | xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-ifmgr-cfg" Paths: /vrfs/vrf /bgp/instance/instance-as /interface-configurations/interface-configuration Note: During the staging, we tried using OpenConfig models. Not all attributes could be set using OpenConfig. For example, it was not possible to set a primary IPv4 address using OpenConfig – while the assigned IPv4 address automatically became secondary addresses. In conclusion, interfaces did not become operational. To compensate for this, we tried using a combination of OpenConfig and Cisco models. However, this did not properly work because of the overlap of OpenConfig and Cisco models. So basically, it was impossible to change both OpenConfig and Cisco models as part of the same transaction (commit). As we wanted to take advantage of the transactional behavior, we finally decided to ignore OpenConfig and integrate straight using Cisco models. **NSP > IP Infusion OcNOS** Device model: network-instances | xmlns="http://openconfig.net/yang/network-instance" interfaces | xmlns="http://openconfig.net/yang/interfaces" Paths: /network-instances/network-instance /network-instances/network-instance/protocols/protocol /network-instances/network-instance/interfaces/interface /network-instances/network-instance/encapsulation /network-instances/network-instance/tables/table /network-instances/network-instance/table-connections/table-connection /network-instances/network-instance/route-targets/route-target /interfaces/interface /interfaces/interface/subinterfaces/subinterface Note: IP Infusion OcNOS looks almost like a vanilla implementation of OpenConfig.

EVPN

IETF RFC8466 defines a YANG service model for L2VPN service delivery. We verified E-Line service provisioning using the standard YANG model northbound in a hierarchical orchestration setup. All test procedures followed the same method as described in the L3VPN service provision with 3 different L2VPN service instances.

The following systems successfully participated in the hierarchical orchestration test: Cisco NSO as controller, Nokia NSP acted as orchestrator, and Ciena 5164 as PE router. The transport network was based on the SR-MPLS.

The following systems successfully participated in the service provisioning with the controller: Nokia NSP acted as controller, Cisco XRv9k as virtual PE router. The transport network was based on the SR-MPLS.

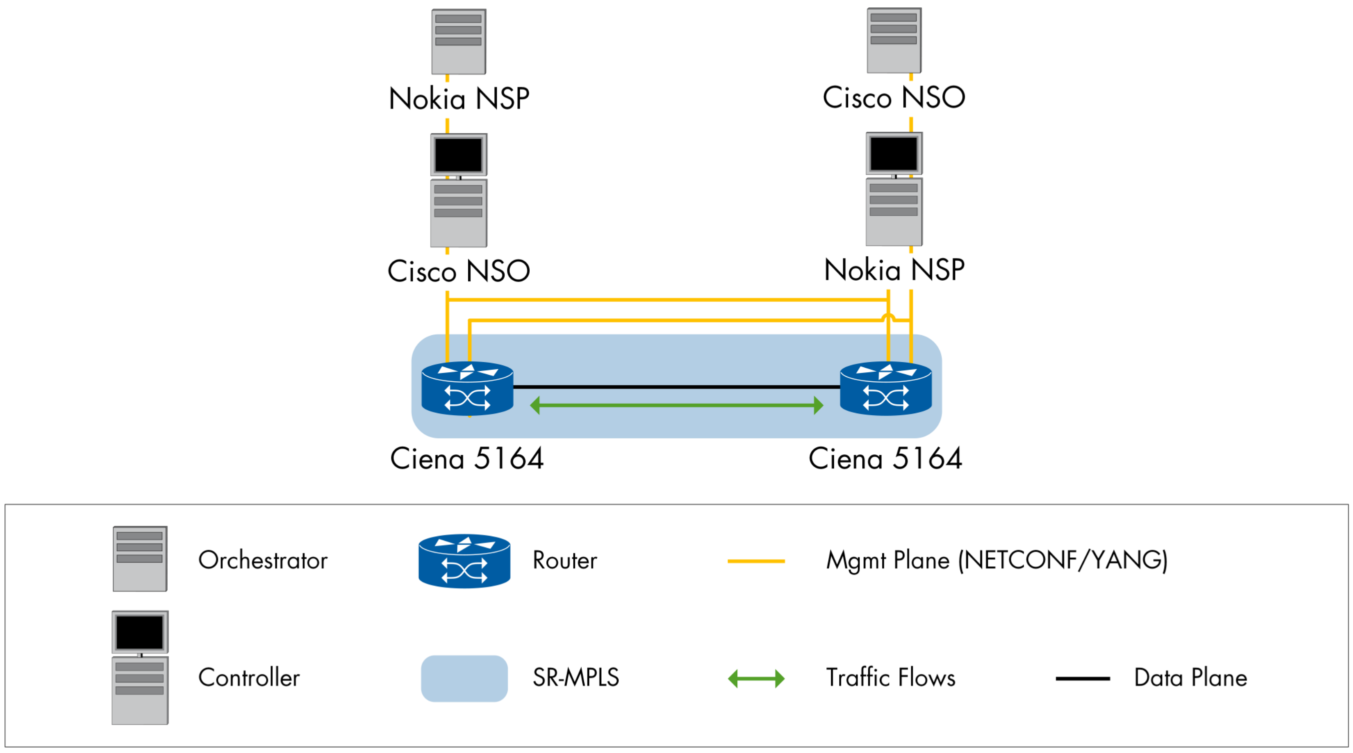

Figure 4: EVPN—Test Setup 1

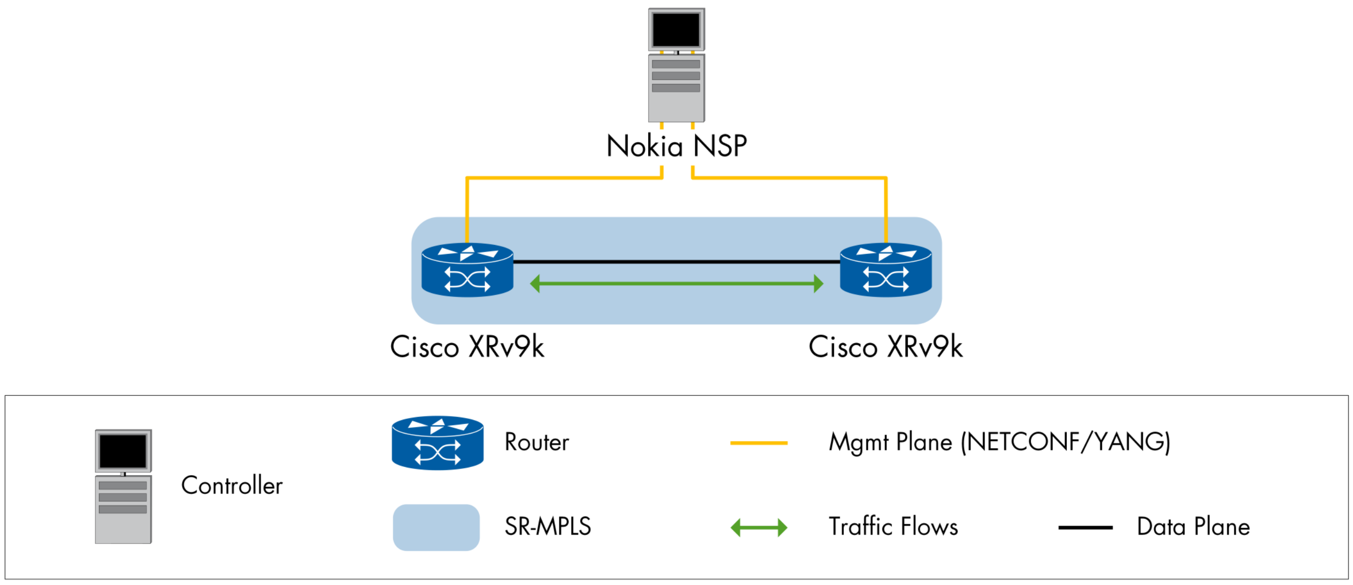

Figure 5: EVPN—Test Setup 2

YANG Models

We covered the following YANG models:

Vendor	Model and Path
Cisco	**NSO > Ciena 5164** Device model: /classifiers/classifier in ciena-mef-classifier.yang /fds/fd in ciena-mef-fd.yang /fps/fp in ciena-mef-fp.yang /evpn/ethernet-segments/ethernet-segment in ciena-evpn.yang /evpn/evpn-instances/evpn-instance in ciena-evpn.yang
Nokia	**NSP > Cisco NSO** Service model: l2vpn-svc (rfc8466) XMLNS: urn:ietf:params:xml:ns:yang:ietf-l2vpn-svc Paths: /l2vpn-svc/vpn-services/vpn-service /l2vpn-svc/sites/site **NSP > Ciena 5164** Device model: classifiers | xmlns="urn:ciena:params:xml:ns:yang:ciena-pn::ciena-mef-classifier" fds | xmlns="urn:ciena:params:xml:ns:yang:ciena-pn:ciena-mef-fd" fps | xmlns="urn:ciena:params:xml:ns:yang:ciena-pn:ciena-mef-fp" evpn | xmlns="http://ciena.com/ns/yang/ciena-evpn" Paths: /classifiers/classifier /fds/fd /fps/fp /evpn/ethernet-segments/ethernet-segment /evpn/evpn-instances/evpn-instance **NSP > Cisco XRv9k** Device model: interface-configurations | xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-ifmgr-cfg" l2vpn | xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-l2vpn-cfg" Paths: /interface-configurations/interface-configuration /interface-configurations/l2-transport /l2vpn/database/xconnect-groups/p2p-xconnects/p2p-xconnect

Bidirectional Forwarding Detection

This test verifies the Bidirectional Forwarding Detection (BFD) configuration via NETCONF. We verified the configuration creation and deletion of BFD via the controller and service protection using BFD as the failure detection on the network device. The protection mechanism was based on L2VPN/L3VPN services previously created using NETCONF. We sent traffic for all VPN services to ensure that the traffic was forwarded over the primary path without any loss under a normal network condition. We observed that the BFD configuration did not impact the traffic of existing services. While traffic was running, we introduced a failure in the Layer 2 segment between both PE devices by removing the interface configuration from the segment (emulated by a Layer 2 device). This failure emulation ensured that none of the port failures should occur locally to avoid local switchover so that the DUT detected the failure after three times of lost BFD messages.

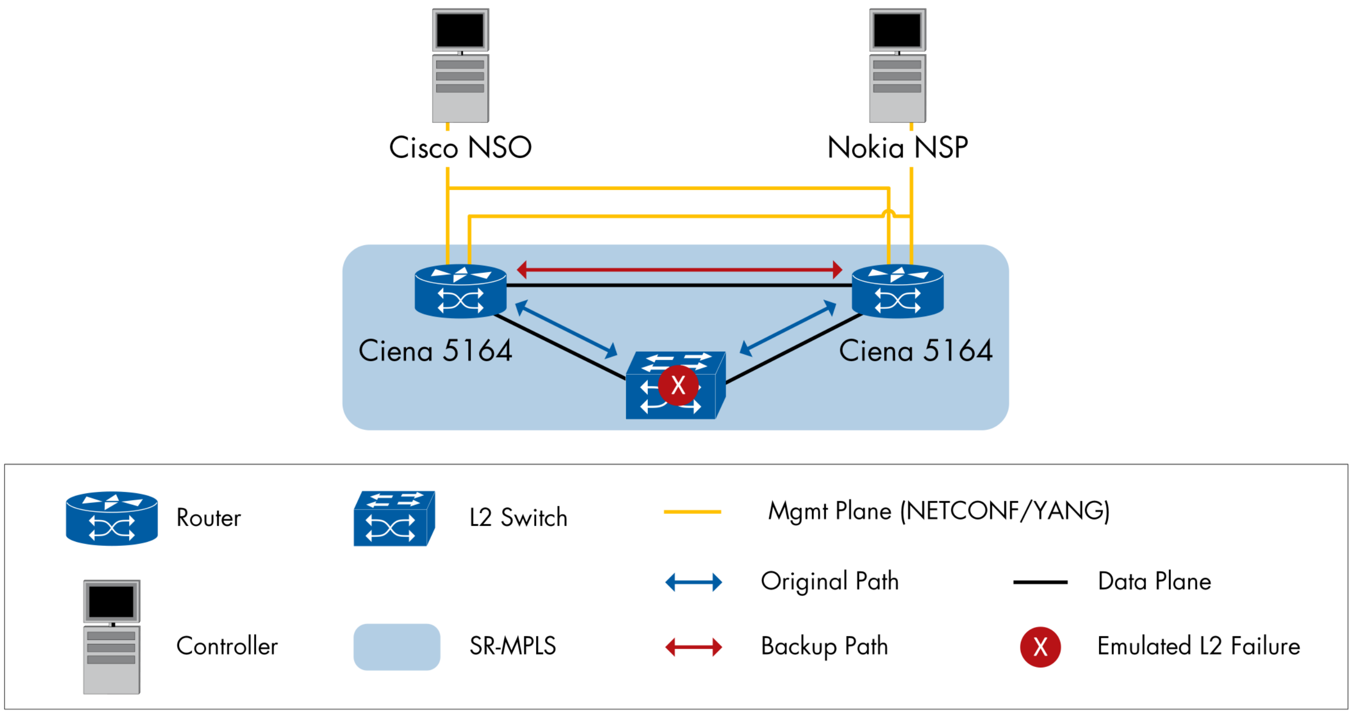

Figure 6: Bidirectional Forwarding Detection—Test Setup 1

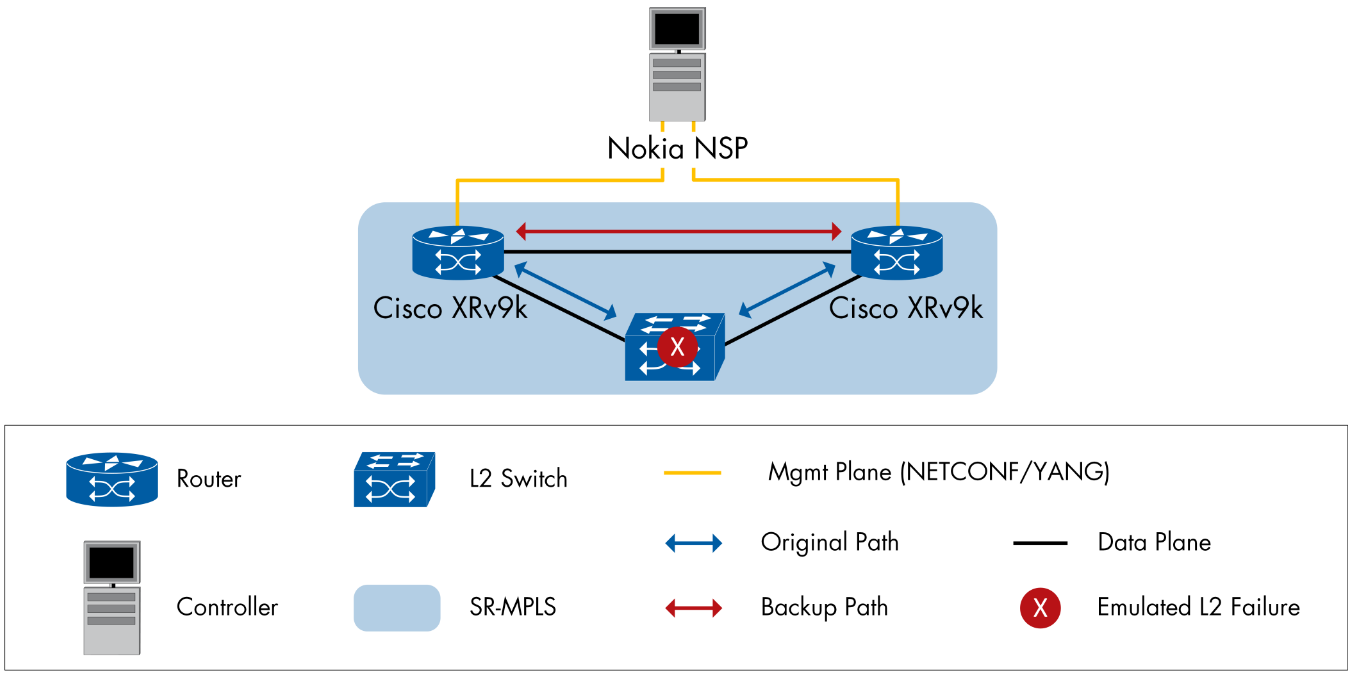

Figure 7: Bidirectional Forwarding Detection—Test Setup 2

We measured the lost packets and calculated the switch over time that the traffic had been taken to switch to the backup path. We expected that the service interruption should be between 30-40ms (based on 10ms interval) or 600-700ms (based on 200ms interval). We removed the BFD configuration and expected no impact on the traffic of existing services. We repeated this above without any BFD to reach a switchover by IGP recalculation. We expected the service interruption to be around one minute. All results were successful with BFD protection.

The following systems successfully participated in the test: Cisco NSO as controller, Nokia NSP acted as controller and Ciena 5164 as PE router. Nokia NSP acted as controller and Cisco XRv9k as virtual PE router.

YANG Models

We covered the following YANG models:

Vendor	Model and Path
Cisco	**NSO > Ciena 5164** /bfd/ip-sh/sessions/session in ciena-bfd-ip-sh.yang /isis/instance/master in ciena-isis.yang
Nokia	**NSP > Cisco XRv9k** router | xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-um-router-isis-cfg" Paths: /router/isis/processes/process/interface-configurations/interface-configuration/interfaces/interface/bfd

MPLS

This test verifies MPLS configuration based on Open-Config YANG models. The MPLS transport was part of the L2VPN service provision test and included in the service configuration tested.

The following systems successfully participated in the test: Nokia NSP acted as controller, Cisco XRv9k as virtual PE router.

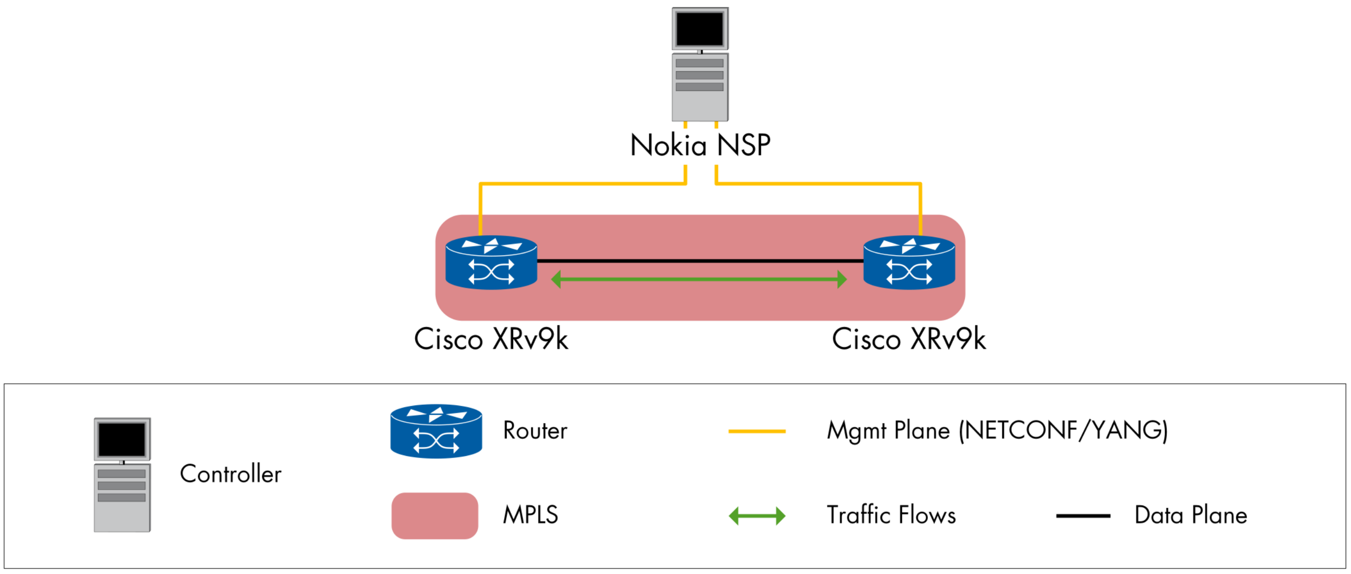

Figure 8: MPLS—Test Setup

YANG Models

We covered the following YANG models:

Vendor	Model and Path
Nokia	**NSP > Cisco XRv9k** mpls-ldp | xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-mpls-ldp-cfg" Paths: /mpls-ldp/default-vrf/global /mpls-ldp/default-vrf/interfaces/interface

MPLS Segment Routing

This test verifies MPLS Segment Routing (SR-MPLS) based on OpenConfig YANG models. The SR-MPLS transport was part of the L2/L3VPN service provision test and included in the service configuration tested.

The following systems successfully participated in the test: Cisco NSO as controller, Nokia NSP acted as controller, and Ciena 5164 as PE router. Nokia NSP acted as controller, Cisco XRv9k as virtual PE router.

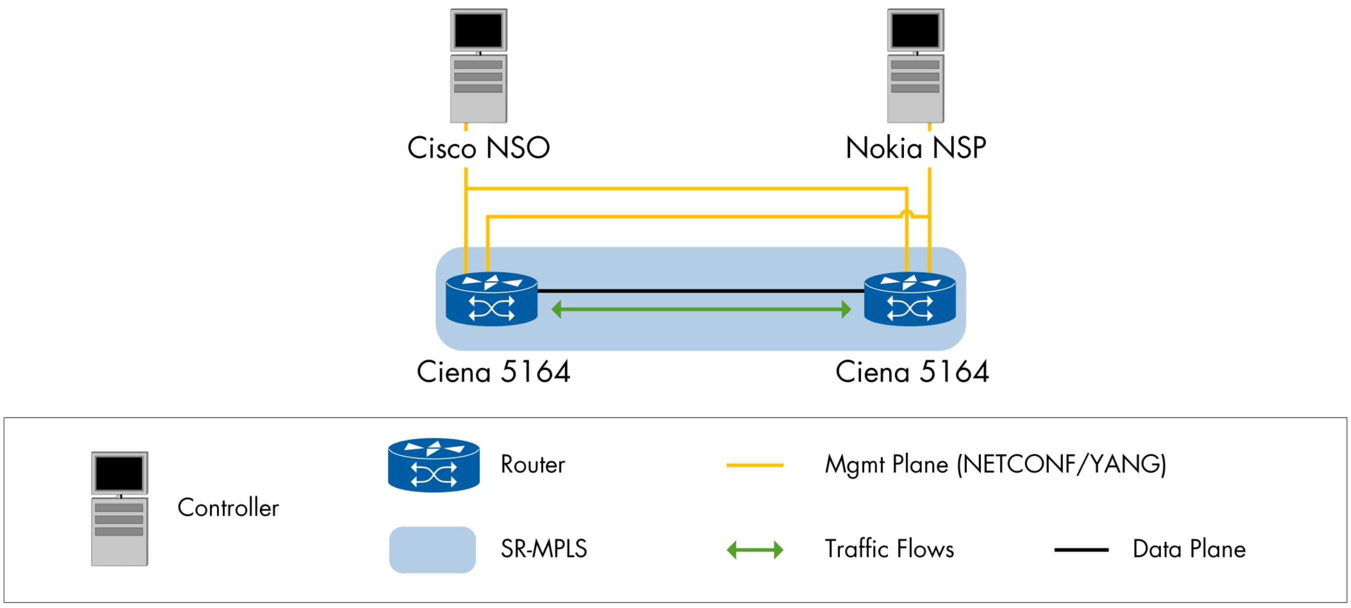

Figure 9: MPLS Segment Routing—Test Setup 1

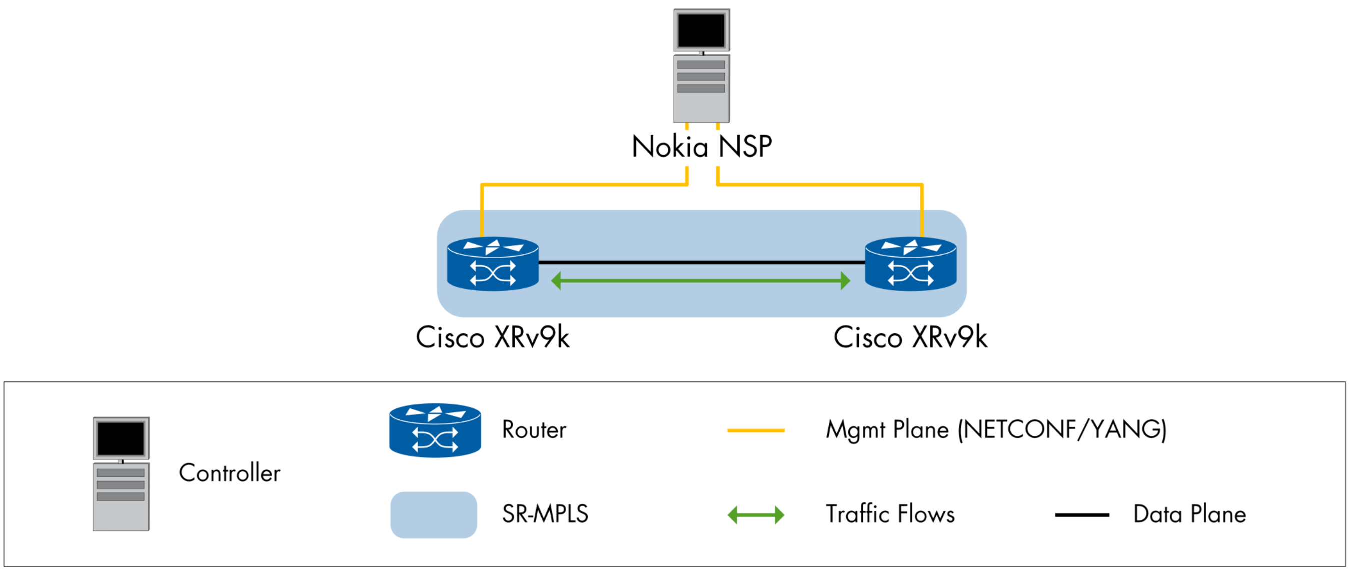

Figure 10: MPLS Segment Routing—Test Setup 2

YANG Models

We covered the following YANG models:

Vendor	Model and Path
Cisco	**NSO > Ciena 5164** /classifiers/classifier in ciena-mef-classifier.yang /fds/fd in ciena-mef-fd.yang /fps/fp in ciena-mef-fp.yang /interfaces/interface in openconfig-interfaces.yang /bgp/instance in ciena-bgp.yang /isis/instance/master in ciena-isis.yang /segment-routing in ciena-sr.yang /mpls/interfaces/interface in ciena-mpls.yang
Nokia	**NSP > Cisco XRv9k** accounting | xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-accounting-cfg" mpls | xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-um-mpls-te-cfg router | xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-um-router-isis-cfg" Paths: /accounting/interfaces/segment-routing /mpls/traffic-eng /router/isis/processes/process/address-families/address-family/mpls/traffic-eng /router/isis/processes/process/address-families/address-family/segment-routing /router/isis/processes/process/interfaces/interface/address-families/address-family/prefix-sid

Retrieve Interface Frame Sizes Distribution

Packet counters are a great indicator for link/service status monitoring. This test verifies the read capabilities of the packet counter with OpenConfig YANG models. We observed OpenConfig YANG extensions. The solution under test carried the standard OpenConfig paths and added more features since the current public definitions from OpenConfig are abstract and contained no further details. However, the current OpenConfig definition included the rate counter definition without any detailed information such as packet sizes.

We observed successful counters per RFC2544 packet sizes, inclusive 64, 128, 256, 512, 1024, and 1518 Bytes incoming packets, as well as 1518 Bytes outgoing packets through the extensions.

Furthermore, we observed a basic counter without any packet sizes for the protocol-compliant participant since the vendor-proprietary YANG model was outside the scope of the test.

The following systems successfully participated in the test: Cisco NSO as controller, Nokia NSP acted as controller and Ciena 5164 as PE router (OpenConfig extension). Nokia NSP acted as controller and Cisco XRv9k as virtual PE router.

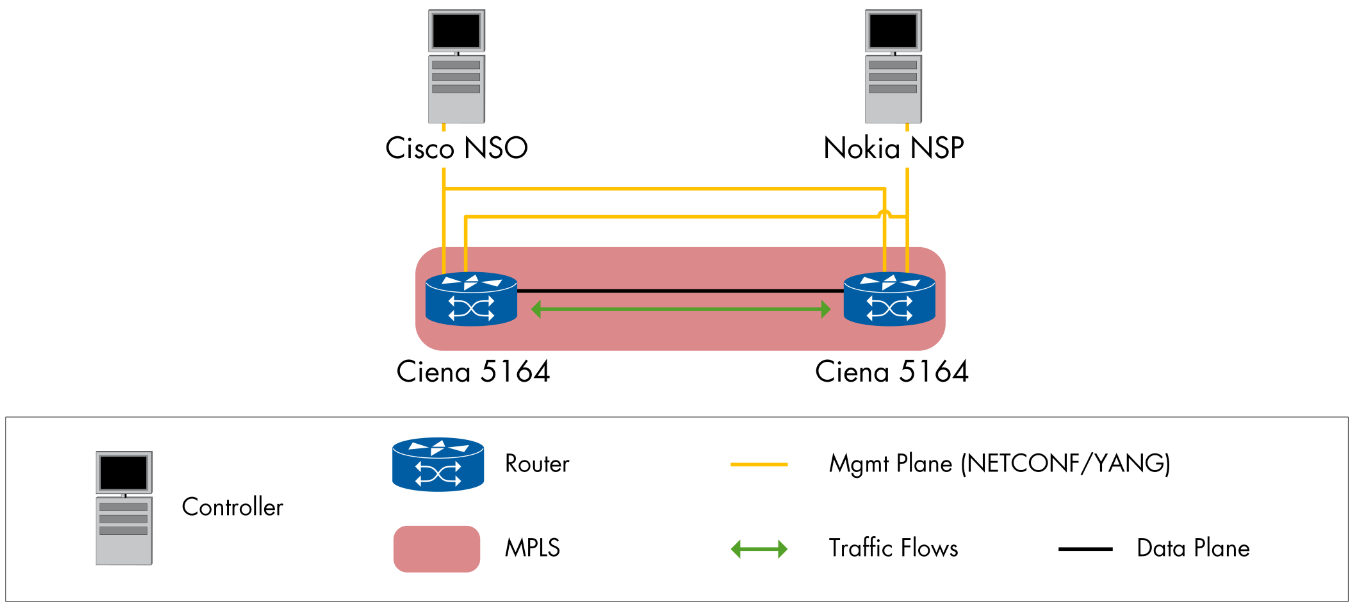

Figure 11: Retrieve Interface Frame Sizes Distribution—Test Setup 1

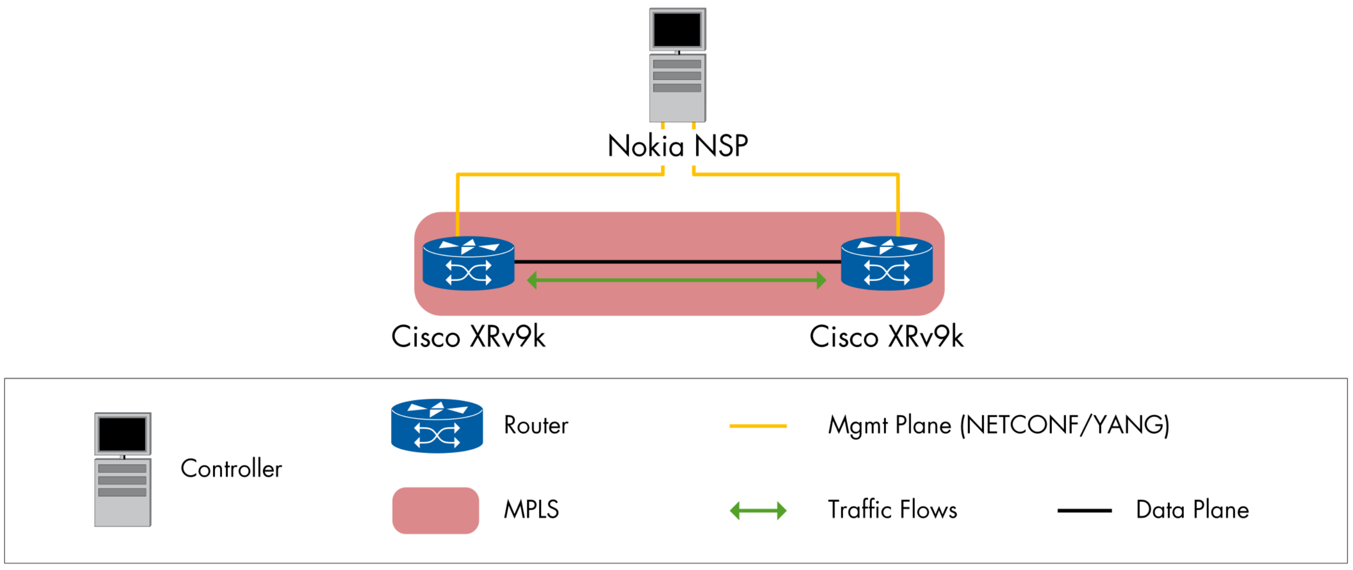

Figure 12: Retrieve Interface Frame Sizes Distribution—Test Setup 2

YANG Models

We covered the following YANG models:

Vendor	Model and Path
Cisco	**NSO > Ciena 5164** /interfaces/interface in openconfig-interfaces.yang /interfaces/interface/state/counters in ciena-openconfig-if-ethernet-port
Nokia	**NSP > Cisco XRv9k** ethernet-interface | xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-drivers-media-eth-oper " Paths: /ethernet-interface/statistics/statistic Note: Cisco IOS XR does not populate data for regular interfaces on virtual routers (XRv). Only the mgmt interface is accessible. **NSP > Ciena 5164** interfaces | xmlns="http://openconfig.net/yang/interfaces" Paths: /interfaces/interface/state/counters Note: Packet size distribution counters are not part of OpenConfig specification. Attributes have been added by Ciena using augmentation.

Precision Time Protocol Configuration

This test verifies the PTP configuration via the Open-Config YANG model. Before starting the test, none of the clock statistics were shown via CLI on the DUT. We configured the PTP profile based on ITU-T G.8275.1 for unicast through the controller towards the network device. We used the device internal clock from one of the PE as a time source and observed the synchronization status, which changed from free-running to locked on another PE. PTP counters appeared and showed PTP packets exchanged between both PEs.

The following systems successfully participated in the test: Cisco NSO as controller, Nokia NSP acted as controller, and Ciena 5164 as PE router.

Initially, we planned to delete the PTP configuration from the DUT. Since there was no participation from the DUT, no results were obtained in this step.

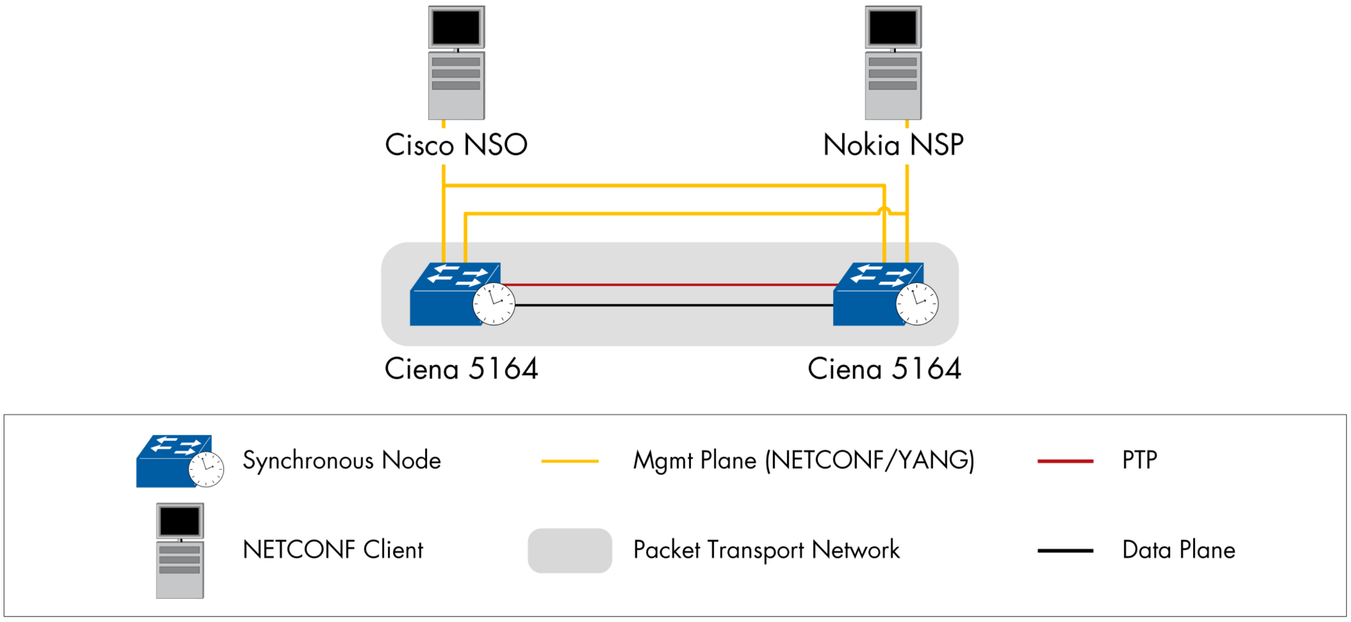

Figure 13: PTP—Test Setup

YANG Models

We covered the following YANG models:

Vendor	Model and Path
Cisco	**NSO > Ciena 5164** /sync in ciena-sync.yang
Nokia	**NSP > Ciena 5164** sync | xmlns="http://www.ciena.com/ns/yang/ciena-sync" Paths: /sync/global /sync/ptp-global /sync/input-references/synce-input-reference /sync/input-references/ptp-input-reference /sync/output-references/synce-output-reference /sync/output-references/ptp-output-reference /sync/protection-groups/frequency-protection-group /sync/protection-groups/phase-protection-group

Management of Interfaces

This was the most basic element included in the configuration provisioning test. We sent traffic for the created IP interface and verified the configuration change (MTU size or IP address) and the deletion through the controller. All tests showed successful results.

Cisco NSO, Nokia NSP as controller, Ciena 5164 as PE router, Cisco XRv9k as virtual PE router, and IP Infusion OcNOS as virtual PE router.

Conclusion

The test results showed mature configuration automation with NETCONF. The end-to-end enterprise VPN provisioning with protection showed successful automation results using a service-oriented approach. In addition, various configuration use cases included the transport network such as MPLS and SR-MPLS for the L2VPN and L3VPN services, as well as BFD for failure detection, display packet counters, and PTP for clock synchronization. Although OpenConfig YANG models are in an abstract stage, vendor-specific field ready solutions already exist through various extensions. We determined the maturity of NETCONF solutions by eliminating device integration problems and providing a service configuration solution utilizing automatically integrated devices.

EANTC AG
Salzufer 14, 10587 Berlin, Germany
info@eantc.de, https://www.eantc.de/
[v1.0 20210804]