EANTC

EUROPEAN ADVANCED NETWORKING TEST CENTER

Huawei Data Center Autonomous Network L4 Verification

EANTC Independent Test Report

June 2025

HUAWEI

Contents

Introduction

Enterprises worldwide are starting transformation journeys for their data center networks. Multiple growth factors align nowadays, leading to the substantial expansion of existing data centers and the establishment of new data center sites and clusters. To name a few reasons: New application workloads, migration to new containerized platforms, private cloud hosting of applications under data sovereignty rules (in-sourcing), and, of course, the avalanche of AI/ML trials and deployments.

In all these cases, the network management of extended data center infrastructures will likely require new operational models. It is no longer feasible to grow the IT teams to cover the manual operations of new and expanded data center deployments. Instead, the TM Forum and ETSI models of Autonomous Networks planning, operations, and monitoring promise to reduce manual efforts drastically.

Enterprises starting the transformation journey need to know whether their chosen vendor solution will support the Autonomous Networks Level 4 features they require. For the first time, EANTC has conducted an independent assessment test of a live AN L4 vendor solution for the data center network area. In contrast with all other assessments, such as questionnaires or paper audits, our test investigates the real software features. We verify how far AN L4 have been implemented by checking the concrete provisioning, application rollout, and monitoring actions.

EANTC has been commissioned by Huawei to conduct an independent assessment of Huawei's Data Center Autonomous Driving Network Solution. We analyzed the solution in detail during a test session in Nanjing, China, in May 2025.

Test Environment

Data center networks have multiple architectures, each with advantages and drawbacks. One of the most common designs nowadays is the leaf-spine architecture. In this design, each leaf switch has uplinks to each spine, providing redundancy and load-balancing, while the downlinks are connected to the servers. Additional leaf switches called border leaf switches are connected to the spine switches and on the other side to the firewalls and the data center provider edge router. Those border leaf switches provide connectivity to the outside world, such as the internet or WAN, and also handle traffic between different security zones, or any flows that must pass through the firewall where security policies are enforced.

Test Highlights

- → Intent-Based POD Design & Zero-Touch Provisioning Application and infrastructure deployment via fully automated, intent-driven workflows.
- → Pre- and Post-Provisioning Validation Deployment simulation and risk evaluation to ensure stability before production rollout.
- → Intelligent Fault Detection & Guided Troubleshooting Real-time detection of service-impacting issues with clear diagnostics and structured rectification steps.
- → End-to-End Visibility & SLA Monitoring Comprehensive monitoring of performance metrics and application health.
- → Automated Backups & Instant Rollback Automatic configuration snapshots, visual change tracking, and immediate rollback capabilities.

The described design is called a Point of Delivery (POD), which represents a building block within a data center. In smaller setups, a single POD might represent the entire data center. However, a large data center could consist of multiple PODs, resulting in what is known as a multi-pod design.. Those PODs are typically connected using a routed inter-POD network (IPN).

In our test, we built a POD infrastructure that includes four server-leaf switches, two border-leaf switches, two spine switches, and two firewalls. Each pair of leaf switches was configured with Multi-Chassis Link Aggregation (MLAG), which allows the servers to connect to both leaf switches as if they were a single device. This adds a layer of resiliency and helps with load balancing.

The network devices in a POD have L3 connectivity to each other, which is established using routed links between the devices and an interior gateway protocol (IGP). In our test bed, OSPF was used as the IGP. BGP is established on top of the underlay and provides the control plane for EVPN services. In data center environments, EVPN services are typically built over VXLAN, which extends the Layer 2 domain across different leaf switches and servers.

In addition to the POD infrastructure, we deployed the management system, which is the main focus of our tests. The management system consists of two main components: iMaster NCE-Fabric and iMaster NCE-FabricInsight. The NCE-Fabric is a virtual component

installed on a server and specializes in device management and configuration.

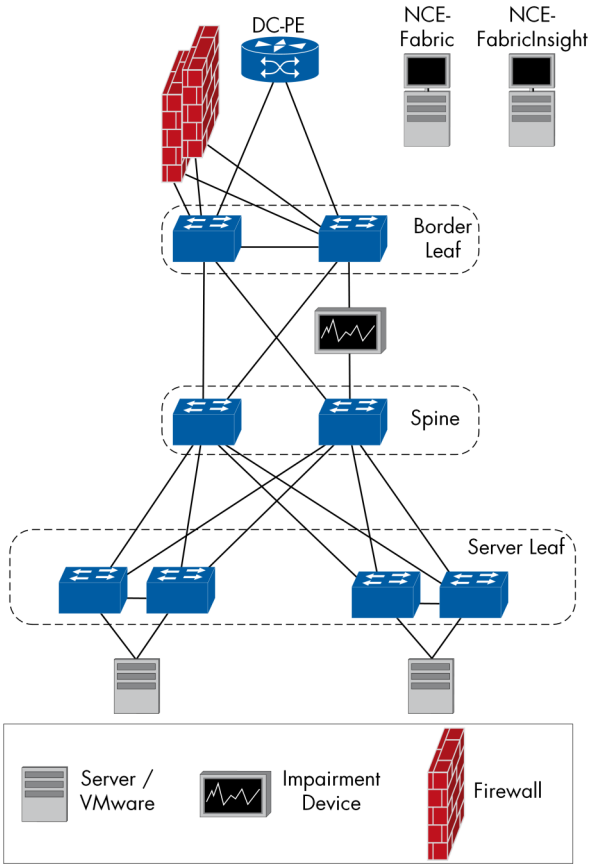
The NCE-FabricInsight is a standalone component that monitors the network, collects telemetry data, and assists with troubleshooting tasks. It consists of three main modules:

- A collector, which gathers telemetry data using protocols like SNMP and gNMI.
- A probe server that receives mirrored traffic from data center devices. In our setup, ERSPAN is used on the leaf switches to mirror selected traffic, typically TCP packet headers, and send it over a GRE tunnel to the probe server. In addition, direct port mirroring is used on the border leaf switches, which send mirrored traffic to the probe server without tunneling.
- Analyzers process the data from both the collector and the probe server. They interpret and present the information to the user in a clear, readable format.

The number of probe servers is not limited to one. It can be scaled out as needed to host all the mirrored traffic.

In addition to the introduced POD design, we installed an impairment device on the link between a border leaf switch and a spine switch. This device is not part of a data center architecture. It was added for testing purposes to introduce packet loss into the path and observe the system's behavior.

The following figure illustrates the physical structure of the testbed.



Physical structure of the testbed

The following table shows the devices used in the test setup, including their software version.

Name	Vendor	Device Type	Software Version	Number
Spine	Huawei	CE9855-32DQ	V300R024C00SPC500	2
BorderLeaf	Huawei	CE6885-48YS8CQ	V300R024C00SPC500	2
ServerLeaf	Huawei	CE6881-48S6CQ	V200R024C00SPC500	2
	Huawei	CE6863-48S6CQ	V200R024C00SPC500	2
Firewall	Huawei	USG6655E	V600R007C20SPC500	2
Impairment Equipment	XINERTEL	XINERTEL	3.0.0.200678	1
iMaster NCE-Fabric	Huawei	Virtual	V100R024C10SPC100	1 Node
iMaster NCE-FabricIsight	Huawei	Virtual	V100R024C00SPC101	5 Nodes (3 Analyzer +1 Collector +1 Probe Server)

Table 1: Devices used in the test setup

Evaluation Matrix

DC POD planning and provisioning	**POD Requirement Analysis**	**Network design generation**	**Network simulation & decision-making**	**Configuration/ provisioning**	**Verification**	**SUM/ Count**
	L4	L4	L4	L4	L4	L4
Application Rollout and Provisioning	**Application launch intent management**	**The solution generated**		**The solution Implementation**	**Verification**	**SUM/ Count**
	L4	L4		L4	L4	L4
Monitoring and Troubleshooting	**Scenario-based monitoring**	**Fault and risk detection**	**Fault diagnosis, solution generation, and decision -making**	**Solution implementation**	**Implementation verification**	**SUM/ Count**
	L4	L4	L3	L3	L4	L3.6
Overall Evaluation	Calculated as the Average of the three scenarios: **(4 + 4 + 3.6) / 3 ≈**					**3.9**

Table 2: Evaluation Matrix

1. DC POD Planning, Construction, and Deployment

Deploying a DC POD from scratch is typically a complex procedure that involves many steps and parameters to consider. It starts with the design phase, where the physical structure, the number of devices to deploy, and the underlay network parameters like IP address spaces, VRFs, and VLAN ranges are determined. After the design phase, a review is conducted to ensure the design is free of issues, such as IP overlapping or missed configurations. The implementation phase follows, during which the design is applied to the devices. This phase is often demanding and requires careful work. After implementation, a verification phase is necessary to confirm the correct implementation and functionality of the design.

Automating the DC POD deployment is a valuable feature that saves time and reduces design errors. The Document ETSI GR ENI 049 specifies the system's requirements and establishes an evaluation scale for the automation grade.

In the following tests, we designed and deployed a DC POD using Huawei iMaster NCE-Fabric and aimed for an automation level 4, which provides a user-friendly, intent-based interface and design verification.

1.1. POD Requirement Analysis and Design Generation

The first step in deploying a DC POD is to define the physical structure and create a parameter plan for the underlay network. In this test, we delegated this task to the iMaster NCE-Fabric. The NCE-Fabric gathered the user's intent, including the number of spine switches, server leaf nodes, border leaf nodes, and the security service as an on/off switch. Following the TM Forum and ETSI ENI standards, this "intent" stage does not involve communication between the NCE-Fabric and the network devices, so at this point, all routers and switches are still turned off.

The intent we defined included four server leaf nodes, two spine switches, two border leaf nodes, and enabled the security service switch. The NCE-Fabric processed this intent and recommended a physical topology with all devices connected properly according to the spine-leaf architecture, as depicted in Figure 1.

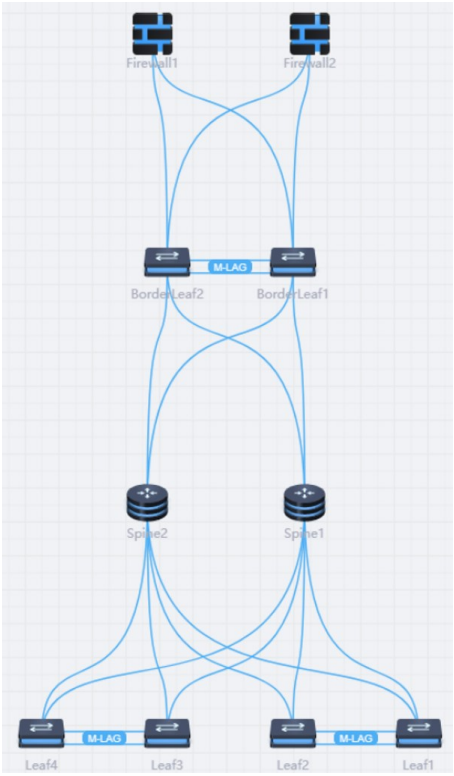


Figure 1: Proposed POD Topology by NCE-Fabric

In addition to the physical design, the NCE-Fabric also suggested hostnames, a management IP address pool, VTEP IP addresses, VTEP MAC addresses, BGP router IDs, link IP addresses for the connections between the network nodes, and recommended using OSPF and BGP for routing. All proposed settings were presented for user review.

Network Topology Data

Name	Role	Management IPv4 Address	VTEP IPv4 Address
Leaf3	Leaf	120.17.19.10/24	10.5.6.5/32
Leaf1	Leaf	120.17.19.11/24	10.5.6.4/32
BorderLeaf2	BorderLeaf	120.17.19.13/24	10.5.6.1/32
BorderLeaf1	BorderLeaf	120.17.19.12/24	10.5.6.1/32
Firewall1	Firewall		--
Firewall2	Firewall		--

VTEP MAC Address	VTEP IPv6 Address	Bgp Router ID	Device Location
0000-5E06-0105	NA	10.6.6.1/32	Local
0000-5E06-0104	NA	10.6.6.4/32	Local
0000-5E06-0106	NA	10.6.6.7/32	Local
0000-5E06-0106	NA	10.6.6.6/32	Local
--	NA		Local
--	NA		Local

Figure 2: Recommended Device Parameters by NCE-Fabric

The suggested parameters, including the intent itself, were editable and could be adjusted by the user. This allowed for customization to fit specific IP schemes or other user-specific requirements. We tested this by modifying the intent after generating the initial network plan, specifically changing the number of server leaf nodes from four to six.

The NCE-Fabric accepted the updated intent and adjusted the topology accordingly. It then rechecked whether the existing parameters were still valid for the new design. As a result, the user needed to update some parameters. For example, the management IP pool did not contain enough addresses to support the two additional nodes.

Since this was still the first step in defining the POD design, the simplest solution was to reenter the updated intent from scratch and generate a new parameter set and network design. In contrast, editing individual parameters such as IP addresses or VTEP settings worked without issues, as long as the new values were compatible with the design.

After the user reviewed the generated parameters, the NCE-Fabric proceeded to create configuration files for each network device in the POD. These files are CLI scripts defined to match each device's specific role and configuration. The scripts include all required configurations, such as interface configurations, IP addresses, routing protocols, and other parameters previously defined during the planning phase. Figure 3 shows a snippet from one of the configuration scripts generated by the NCE-Fabric.

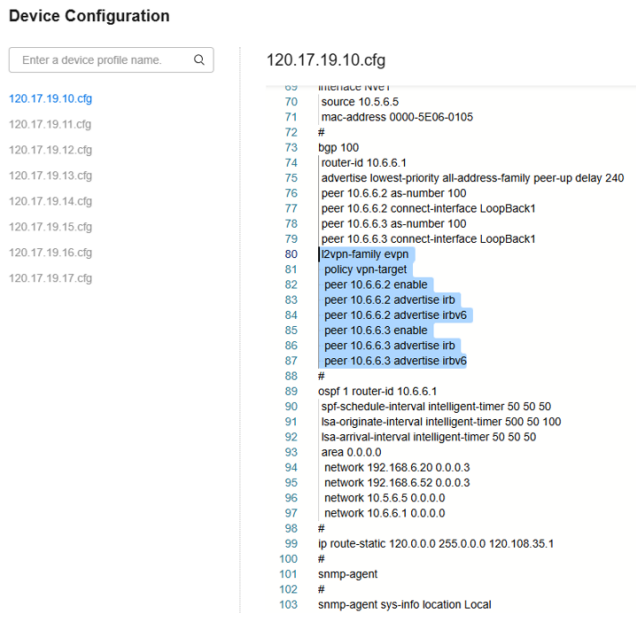


Figure 3: Configuration Script Generated by NCE-Fabric for a Network Node

The results achieved in this test confirm that the NCE-Fabric meets the Level 4 criteria for the capabilities POD requirement analysis and automated network design generation, according to the requirements for DC POD planning and provisioning defined in ETSI GR ENI 049.

1.2. Network Simulation and Decision-making

At this point, the NCE-Fabric has already generated the POD design and configuration scheme and could deliver the configurations directly to the network devices. However, performing a verification step before deployment is highly beneficial, as it helps identify and resolve potential issues before bringing the POD online.

This verification step simulates the network based on the generated configuration scripts. The NCE-Fabric simulates the entire network, including all devices, and produces an evaluation report. This report analyzes the network from several perspectives, including routing reliability, overall connectivity, routing black holes, routing loops, configuration verification, and routing conflicts. In our test, the simulation report did not identify any issues such as routing black holes, loops, or conflicts. All routes were reported as reachable, and connectivity between devices was confirmed to be normal.

To verify whether the simulation is reliable in detecting configuration issues, we intentionally modified the configuration scripts to introduce multiple problems. We then reran the simulation to evaluate if the NCE-Fabric could identify the introduced issues.

Blackhole

In this step, we added a static route to the configuration script of one node. The static route `<IP route-static 1.1.1.0 255.255.255.0 NULL 0>` is classified as a black hole route because it directs traffic to the NULL 0 interface. After rerunning the simulation, the evaluation report highlighted these issues in the routing black holes section, as shown in the figure 4 below.

Subnet Mismatch

We modified multiple IP addresses in a configuration script for several links to simulate a scenario where the physical interfaces at both ends were assigned to different subnets. We tested two variations: first, by changing the subnet mask while keeping the IP addresses the same; and second, by changing the IP addresses while keeping the subnet mask unchanged. We then reran the simulation. The NCE-Fabric detected the issue in both scenarios and described the problem in the simulation report under the Configuration Verification tab. Figure 5 highlights the issue as it appears in the report.

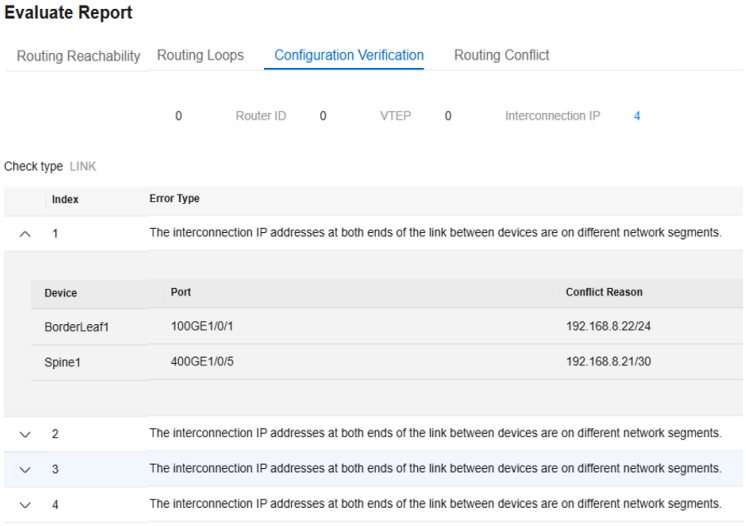


Figure 5: Simulation Report – Inconsistent Interface IP Addressing

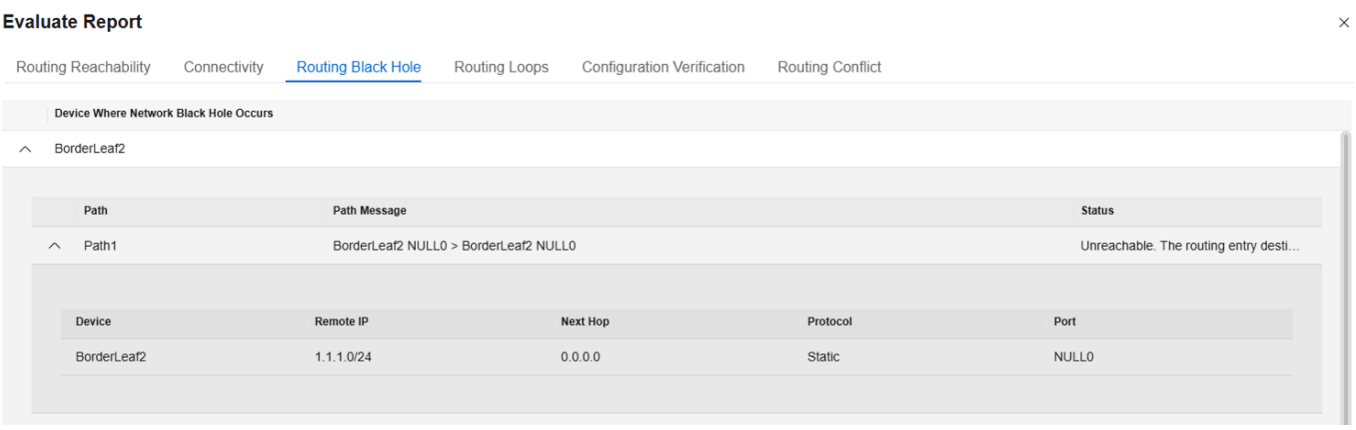


Figure 4: Simulation Report - Routing Black Hole

Routing Loop

Routing loops are complex network problems that require wide analysis and elimination and can cause a network breakdown. In this test, we intentionally modified the generated configuration scheme to create a routing loop in the network. This was done by adding two static routes to two directly connected devices, a spine and a border leaf. Each static route pointed to the other device as the next hop for the same destination IP address. After running the simulation, the NCE-Fabric detected and reported the issue under the Routing Loops section. The report included in the figure 6 details about the specific routes that caused the loop.

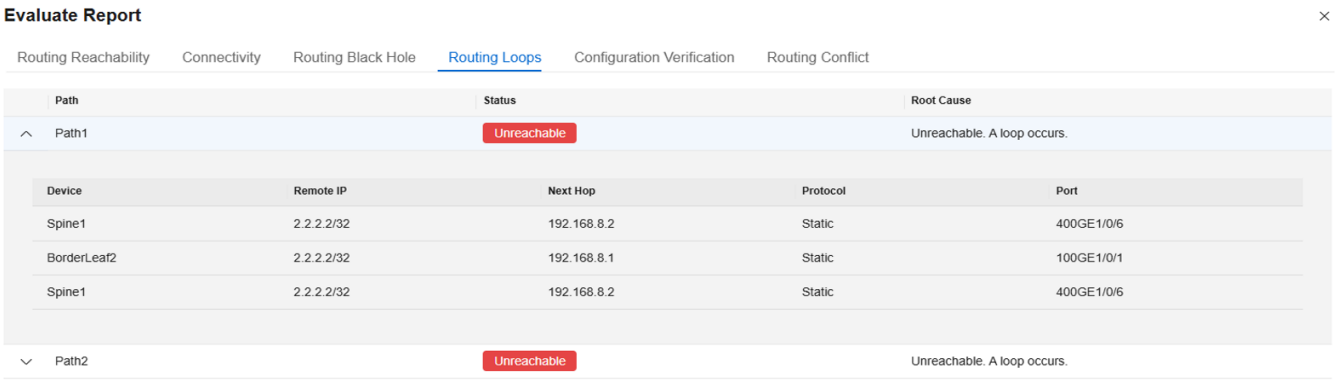


Figure 6: Simulation Report – Routing Loop

Interface Down

We simulated an interface failure scenario to further evaluate the NCE-Fabric's ability to detect issues before deploying configurations to the devices. This was done by adding an interface-shutdown command to the Leaf 4 configuration script. After running the simulation, the NCE-Fabric identified the issue and reported all affected paths under the connectivity tab as unreachable, as shown in Figure 7 below.

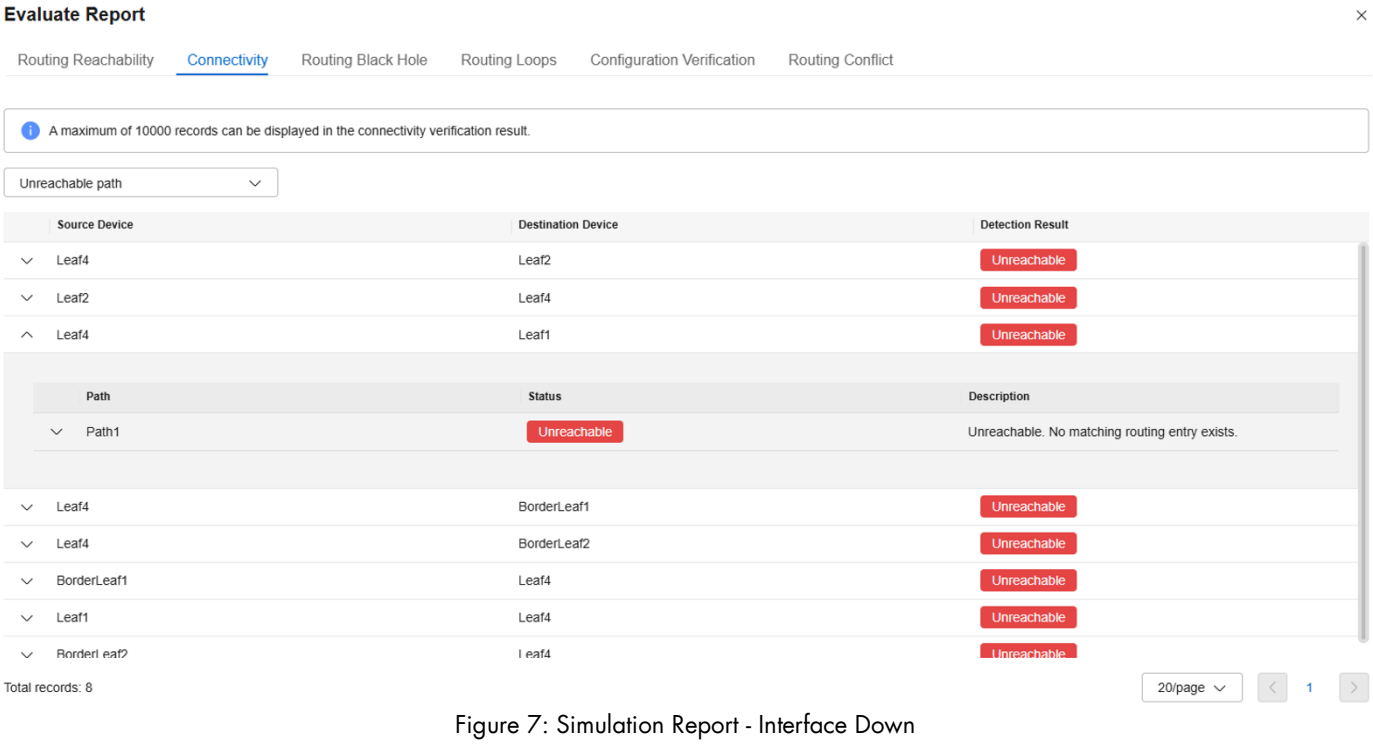


Figure 7: Simulation Report - Interface Down

IP Conflict

To produce a routing conflict and evaluate the NCE-Fabric's ability to detect it, we modified the loopback address of one device to match that of another. The NCE-Fabric successfully detected this conflict and reported it in the simulation report under the Routing Conflict section. The report included details such as the conflicting IP address and the devices where the conflict was detected. The figure 8 shows the conflicting IP in the simulation report.

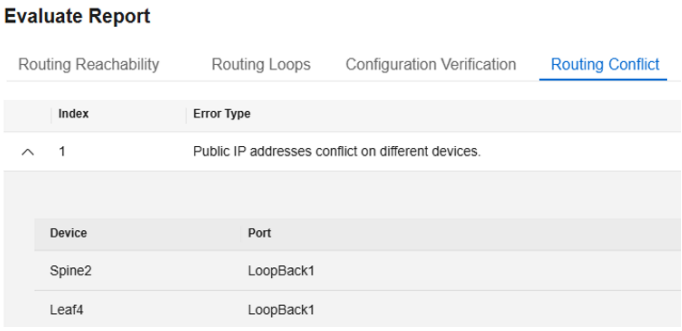


Figure 8: Simulation Report - IP Conflict

This test confirms that the NCE-Fabric meets the Level 4 network simulation and decision-making capability requirements, as defined in ETSI GR ENI 049.

1.3. Network Provisioning Execution

After confirming that the configuration scheme was free of mismatched settings and inconsistent parameters, we proceeded to deploy it to the physical network devices. All devices were installed and connected to the management network according to the topology. Additionally, each device was connected to a console server to allow log monitoring during the zero-touch provisioning process.

Before starting provisioning, we performed a factory reset on all devices using the console connection. This ensured a clean starting state.

The zero-touch provisioning process begins with the spine switches, which must be recognized by the NCE-Fabric using their Electronic Serial Numbers (ESNs). We entered the ESNs of both spine switches into the NCE-Fabric. Once registered, we selected the spine switches for onboarding and restarted them.

During the boot process, the spine switches automatically started zero-touch provisioning. They first requested an IP address from the DHCP server integrated into the NCE-Fabric. Along with the IP address, the DHCP server provided the location of a Python script used for onboarding. The spine switches downloaded this script via Secure File Transfer Protocol (SFTP) and executed it. The script connected the devices to the NCE-Fabric using predefined credentials and retrieved their configuration files.

Onboarding devices in the NCE-Fabric is a straightforward process. As shown in Figure 9, the user selects the required devices from the left panel, moves them to the right panel, and confirms the deployment.

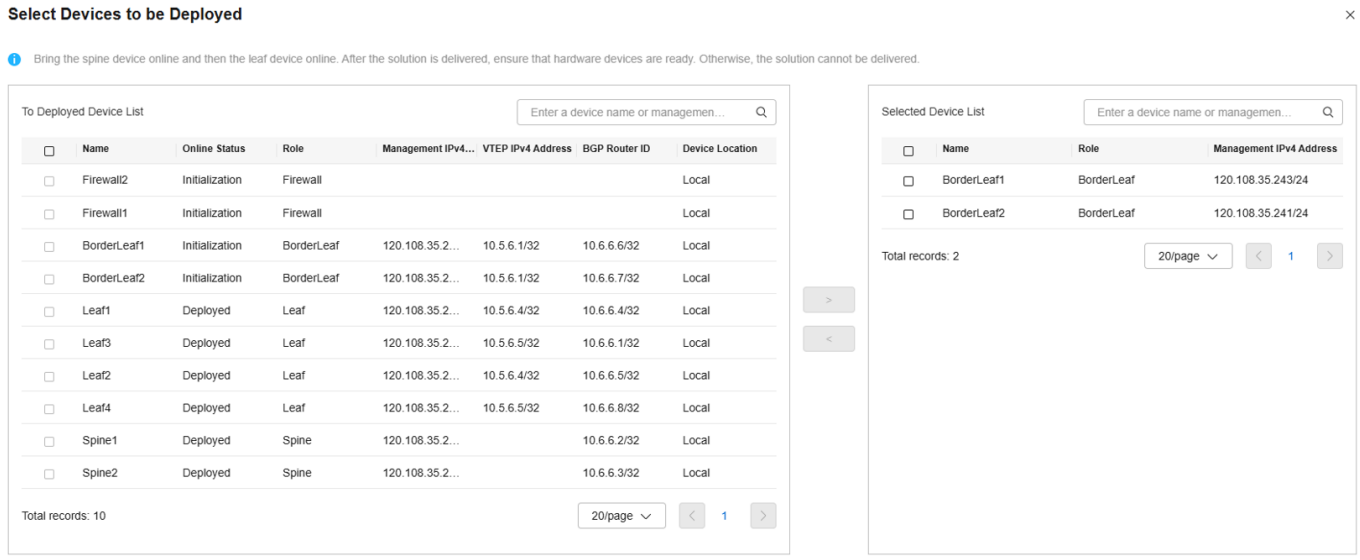


Figure 9: Device Selection for Onboarding

After successfully onboarding the spine switches, we continued with the leaf and border leaf switches. These devices follow the same provisioning steps as the spines, but with a key difference: only the spines require ESNs to be manually entered. Leaf and border leaf switches are identified automatically by the NCE-Fabric based on their physical connection to the spine nodes. LLDP is used to verify each device's identity and position during this process. For example, a device connected to spine interface 1 is recognized as leaf 1, and the corresponding configuration is assigned accordingly.

We onboarded the leaf switches first, followed by the border leaf switches. Although these two steps cannot be performed simultaneously, their order is flexible—either group can be onboarded first. The only strict requirement is that all spine switches be onboarded before any other devices.

After completing the onboarding process, the NCE-Fabric displayed a deployment summary confirming the successful provisioning of all eight nodes, as shown in Figure 10.

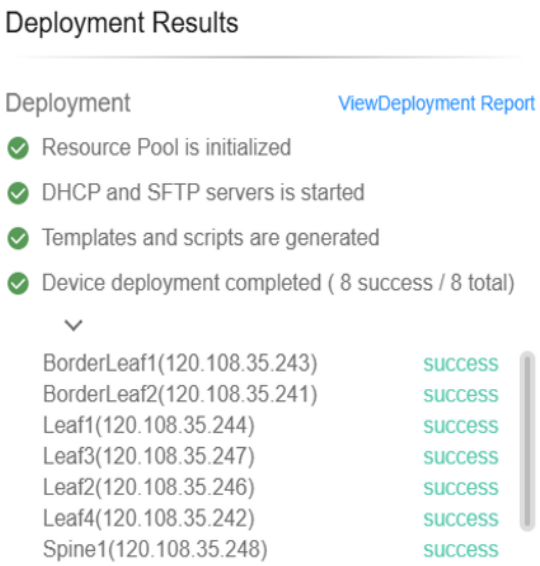


Figure 10: Deployment Results Summary

We ran multiple connectivity checks using ICMP between the devices' loopback addresses to verify that the devices and the underly network were deployed correctly. All checks were successful, confirming that the onboarding process completed as expected.

ETSI GR ENI 049 states that Level 4 configuration and provisioning capabilities require "automatic device configuration with zero-touch deployment, and plug-and-play." In our testing, the NCE-Fabric met this requirement by successfully executing zero-touch onboarding and automatically applying device-specific configurations without manual intervention.

1.4. Network Provisioning Verification

In data center networks, configuration inconsistencies and abnormal traffic patterns can lead to service disruptions if not identified early. Continuous monitoring is essential to reduce this risk. NCE-FabricInsight, a dedicated component separate from NCE-Fabric, was developed specifically for this purpose.

NCE-Fabric is responsible for planning, provisioning, and configuring network systems, while NCE-FabricInsight is focused on monitoring and conducting operational risk analysis. It evaluates the network through device configurations and real-time telemetry data, analyzing risk across five dimensions: reliability, consistency, performance, capacity, and stability. This approach enables the early detection of potential issues and offers continuous insight into the overall health of the data center network.

Initially, NCE-FabricInsight reported zero risks, as shown in Figure 11. This was expected since the network was operating exactly as provisioned by NCE -Fabric, and no traffic had yet been generated into the network.

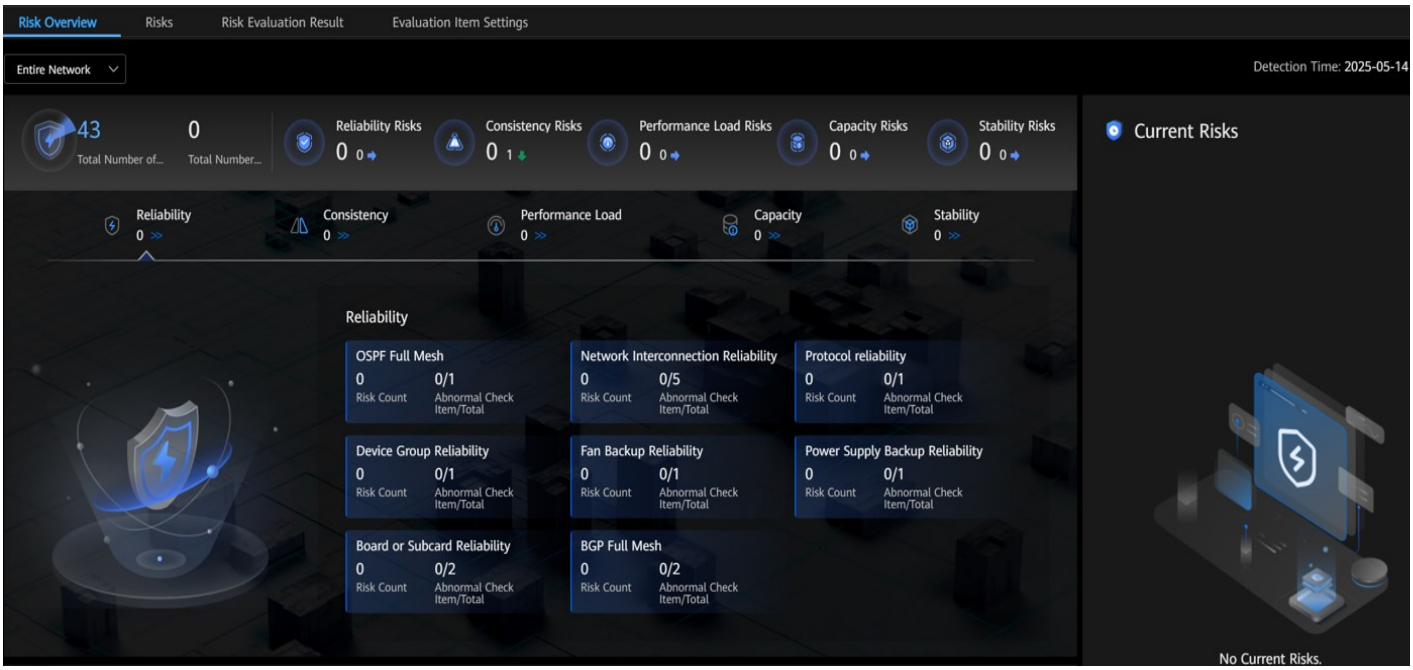


Figure 11: Initial Zero-Risk Status

To test the NCE-FabricInsight's ability to detect configuration-related risks, we intentionally introduced a mismatch in the VXLAN Tunnel Ingress Replication List between two leaf switches. On Leaf1, we added an extra VNI entry (vni 100 head-end peer-list protocol bgp) that was not present on Leaf2. This resulted in an inconsistency between the two devices' Nve1 interface configurations.

NCE-FabricInsight successfully identified the issue as a VXLAN tunnel ingress replication list inconsistency. It highlighted the mismatch between Leaf1 and Leaf2, showing the exact configuration difference, and flagged it as a risk under the configuration check.

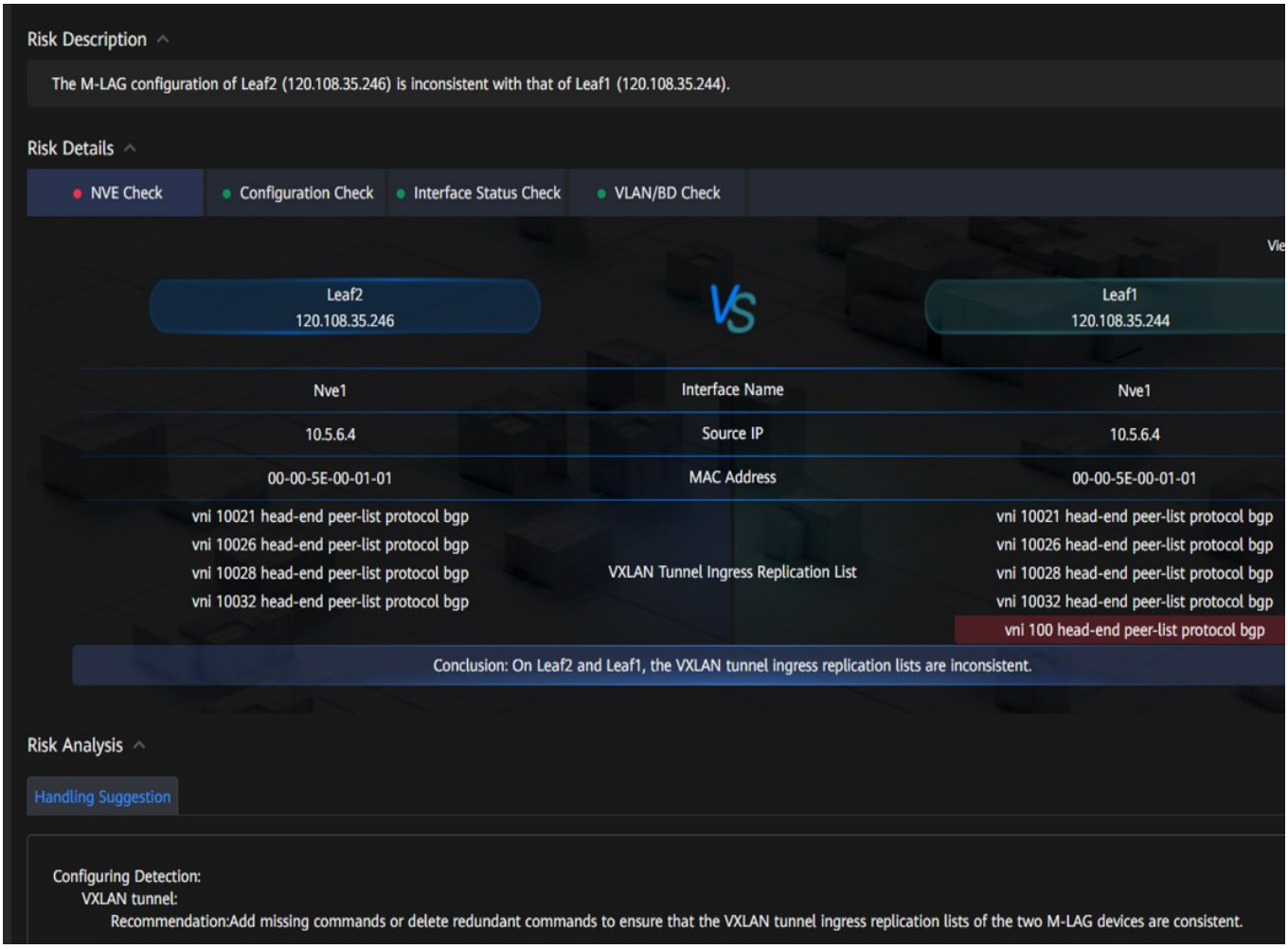


Figure 12: M-LAG inconsistency Risk

NCE-FabricInsight compared both devices configurations side-by-side, showing their IP addresses, interface names, MAC addresses, and the complete VNI configuration under interface Nve1. It clearly identified the VXLAN tunnel ingress replication list as inconsistent and highlighted the specific configuration difference. In the Risk Analysis section, NCE-FabricInsight recommended a resolution to the issue. It advised adding missing commands or removing redundant entries to ensure consistent VXLAN configurations between the two M-LAG peers. After applying the recommended change, the inconsistency was resolved successfully.

The network verification and risk evaluation features provided by NCE-FabricInsight meet the Level 4 requirements for verification capability, as defined in the ETSI GR ENI 049 document.

2. Application Rollout and Provisioning

Introducing new applications to a network requires several configuration steps, such as defining the security zones in which the application's components will be deployed, establishing communication restrictions between security zones, managing IP addressing, configuring physical ports, and ensuring external network connectivity.

Like the previous approach for designing and provisioning the network and devices, the application rollout in the network involves similar steps, including intent management, network scheme generation, simulation, and, ultimately, the implementation and verification of the scheme. In the upcoming tests, we will go through all those steps in detail.

2.1. Application Launch Intent Processing

Modern data centers require scalable, agile, and reliable infrastructure to meet the demands of dynamic workloads and increasing service expectations. Minimizing manual operations, ensuring configuration consistency, and accelerating service rollout are critical for operational efficiency. In this context, the test evaluated NCE-Fabric's ability to provision intent-based applications, integrate with virtual machine manager (VMM), and recommend deployment solutions.

We began the test by integrating a VMware vCenter environment as the VMM. As stated by Huawei, NCE-Fabric supports interconnection with platforms such as VMware vCenter and Microsoft System Center, while Nutanix integration is under development. In our case, we used a pre-configured VMware vCenter installation and completed the setup by entering the vCenter URL and credentials in the NCE-Fabric interface.

We then created a new tenant named "EANTC" from scratch, and linked it to the previously provisioned fabric and the VMM instance (vcenter141).

Then we started the applications provisioning using NCE-Fabric's intent-driven approach. For each application, we entered a new intent. Since no templates were available, we followed a customized modeling approach. For the first application (App1), we defined a single security zone connected to the fabric, entered IPv4 and IPv6 address ranges, and then defined a service, which we mapped to the appropriate leaf switches. We did not use VMM integration for App1, so we manually configured the switch ports and assigned VLAN 3001 on leaf1, leaf2, leaf3, and leaf4. The service was associated with a distributed switch (DSwitch) managed by the VMM.

For the second application (App2), we used VMM integration, eliminating the need for manual VLAN and port mapping. NCE-Fabric automatically created a port group in the vCenter cluster. However, NCE-Fabric did not manage virtual machines and their IPs. We still had to bind the VM's network adapters to the correct port group and configure IP addresses manually.

For the third application (App3), we defined two security zones, three services, and their communication relationships. We specified which services were associated with which zones and which zones were allowed to communicate with each other, as shown in Figure 13.

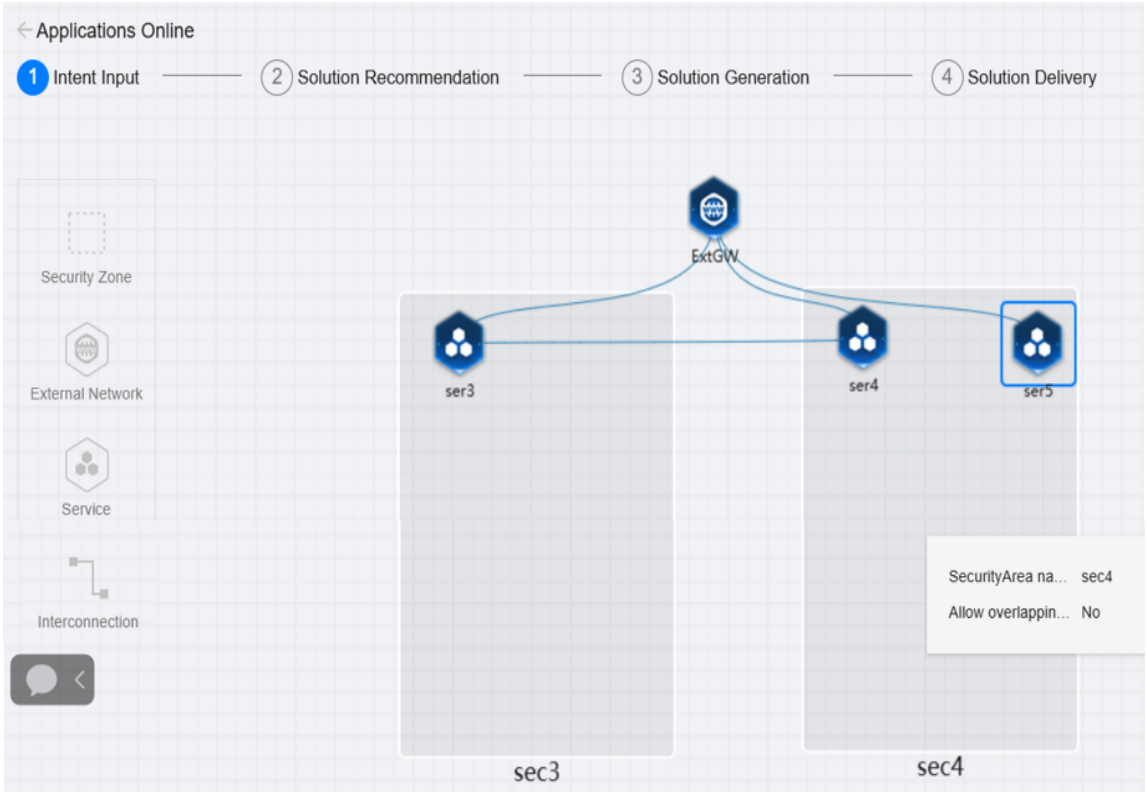


Figure 13: App3 Intent Definition

Once all three applications had been defined, we moved on to configuring inter-application communication. We provided NCE-Fabric with an intent specifying that App1 (Service 1) and App2 (Service 2) were allowed to communicate across their respective security zones. This inter-application communication intent enabled NCE-Fabric to generate deployment options for the required connectivity while preserving isolation for the other services.

This test confirmed that NCE-Fabric meets the Level 4 requirements defined in the ETSI GR ENI 049 document for the Application Launch Intent Management capability.

2.2. Application Scheme Generation

In the previous test, NCE-Fabric gathered all the application intents along with the defined interconnection requirements. Based on this input, it suggested several deployment options for each application and for the inter-application connectivity. The user could then choose the suitable option depending on user-specific design goals or operational constraints.

For App3, which consisted of two security zones and three services, NCE-Fabric proposed two deployment solutions. In Solution 1, the system grouped services into two Virtual Private Clouds (VPCs), each representing one security zone. A shared logical gateway connected both zones, enabling inter-zone communication. This solution was optimized for large-scale environments, balancing resource efficiency with centralized policy control.

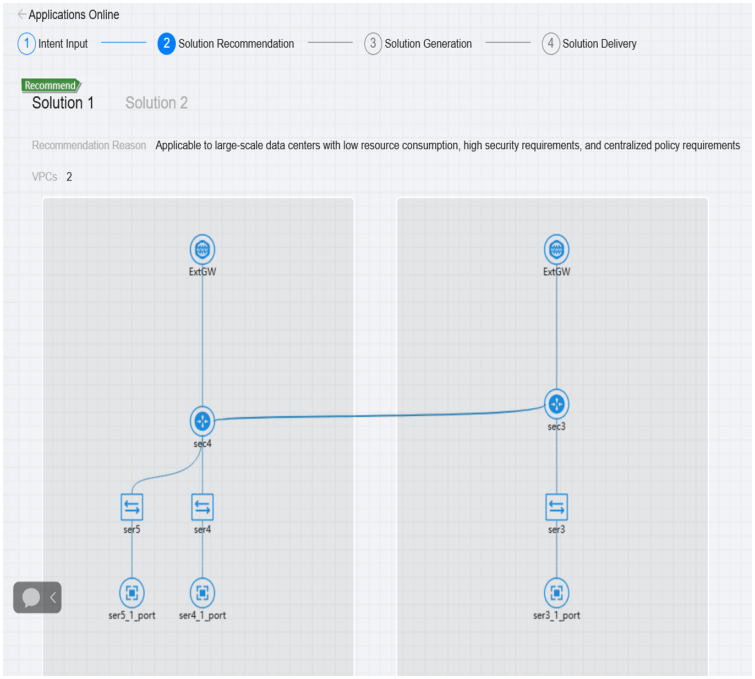


Figure 14: App3 - Deployment solution 1

In Solution 2, NCE-Fabric deployed three separate VPCs, assigning each service to its own isolated security zone. Communication between VPCs was configured based on the defined application intent. For instance, where the intent permitted communication between services in different zones, NCE-Fabric established routing paths accordingly. This solution provided strict isolation by default and selectively enabled inter-zone communication through explicitly defined interconnection logic, trading increased resource usage for greater segmentation and control.

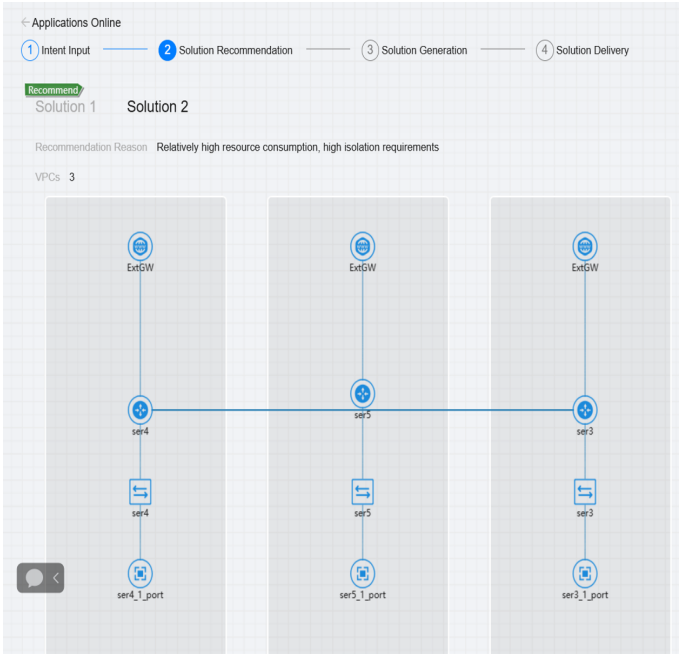


Figure 15: App3 - Deployment solution 2

For App1 and App2, NCE-Fabric provided a single deployment option in each case, as both applications were simple in structure, consisting of a single service within a single security zone.

After selecting the preferred deployment solution, we moved to the solution generation step. In this phase, the user specifies the server-facing interfaces on the leaf switches where the application will be deployed and assigns a VLAN tag to classify the traffic. For inter-application connectivity, the user also defines which subnets are permitted to communicate between the security zones.

2.3. Application Scheme Simulation and Decision-Making

The applications' configurations have been created in NCE-Fabric but have not yet been delivered to the network devices. At this point, running a simulation is helpful and provides the user with insight into the expected changes and the behavior of the applications once deployed. NCE-Fabric supports application

simulation and analyzes the results from multiple perspectives.

We ran the simulation for App3's configurations. NCE-Fabric provided a detailed simulation report that includes the logical services and entities that will be created in the network, along with descriptions indicating which logical instances will be used to deploy each service.

The simulation report confirmed a successful creation of all required logical components, including logic networks, logic routers, switches, and external connectivity elements. Each resource was marked with a "Success" execution status, indicating that the planned configuration passed the validation, as shown in Figure 16.

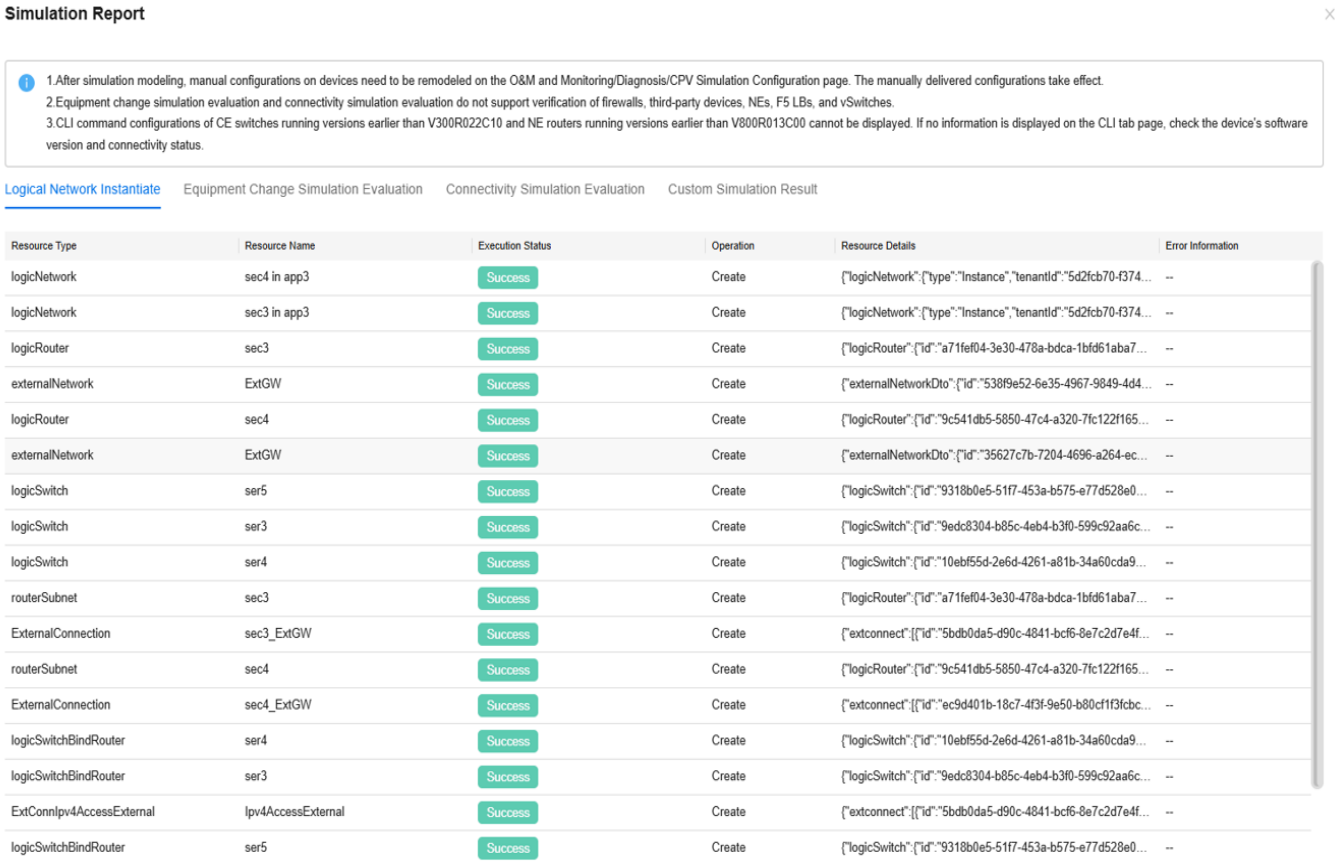


Figure 16: Simulation Report - Logical Network Instantiate

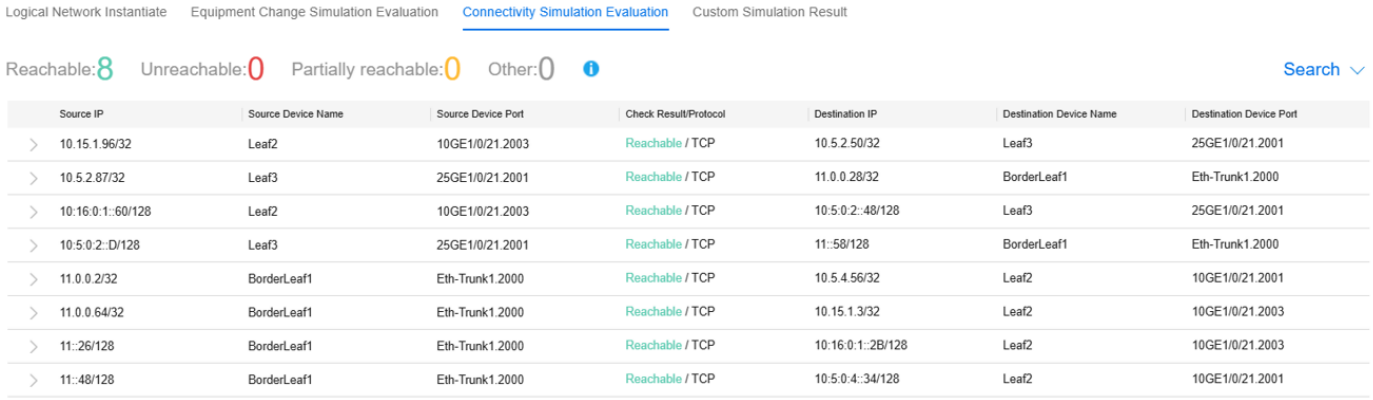
The simulation report also included detailed information about device configuration changes and resource consumption resulting from the application deployment. This data helps verify whether each device in the fabric can support the required configuration without exceeding resource limits.

The report showed parameters such as the number of VRFs, static routes, Layer 2 sub-interfaces, and VNI/BD/EVPN entries per device. As shown in Figure 17, both current consumption and expected usage after deployment were listed.

Figure 17: Simulation Report - Equipment Change Simulation Evaluation

The device entries shown in the previous figure can also be expanded to display the exact configuration changes that will be applied. This information is available in CLI and NETCONF formats, allowing for a granular verification level.

Under the Connectivity Simulation Evaluation tab, NCE-Fabric checked the reachability between IP addresses assigned to the subnets defined in the application services. The simulation listed eight source-destination pairs associated with specific devices and interfaces. All entries were marked as "Reachable," indicating that the simulated configuration is supposed to work as expected.



Source IP	Source Device Name	Source Device Port	Check Result/Protocol	Destination IP	Destination Device Name	Destination Device Port
10.15.1.96/32	Leaf2	10GE1/0/21.2003	Reachable / TCP	10.5.2.50/32	Leaf3	25GE1/0/21.2001
10.5.2.87/32	Leaf3	25GE1/0/21.2001	Reachable / TCP	11.0.0.28/32	BorderLeaf1	Eth-Trunk1.2000
10:16:0:1::60/128	Leaf2	10GE1/0/21.2003	Reachable / TCP	10:5:0:2::48/128	Leaf3	25GE1/0/21.2001
10:5:0:2::D/128	Leaf3	25GE1/0/21.2001	Reachable / TCP	11::58/128	BorderLeaf1	Eth-Trunk1.2000
11.0.0.2/32	BorderLeaf1	Eth-Trunk1.2000	Reachable / TCP	10.5.4.56/32	Leaf2	10GE1/0/21.2001
11.0.0.64/32	BorderLeaf1	Eth-Trunk1.2000	Reachable / TCP	10.15.1.3/32	Leaf2	10GE1/0/21.2003
11::26/128	BorderLeaf1	Eth-Trunk1.2000	Reachable / TCP	10:16:0:1::2B/128	Leaf2	10GE1/0/21.2003
11::48/128	BorderLeaf1	Eth-Trunk1.2000	Reachable / TCP	10:5:0:4::34/128	Leaf2	10GE1/0/21.2001

Figure 18: Simulation Report - Connectivity Simulation Evaluation

The entries in the previous figure can be expanded to display all possible paths between each pair of IP addresses. These paths typically differ in the intermediate hops.

The Connectivity Simulation Evaluation can also be customized. The user may specify any two IP addresses, and NCE-Fabric will simulate and validate the connectivity between them and generate a report similar to the one shown previously in Figure 18.

We then ran simulations for App1 and App2, including their inter-application connectivity. The simulation output followed the same structure format as the App3 results, while reflecting the specific design of App1 and App2. NCE-Fabric provided a detailed view of the logical components to be instantiated, the configuration changes on the relevant devices, and the expected resource consumption. In addition, the connectivity simulation confirmed that the defined communication path between App1 and App2 was reachable.

To evaluate the simulation's ability to detect potential issues in the services to be deployed, we emulated interface failure scenarios on the links between Leaf2 and both spine switches. This was done in two ways: first, by shutting down the physical interfaces on the device using CLI, and second, by removing the corresponding links within the NCE-Fabric user interface. In both cases, NCE-Fabric detected the disruption and marked the paths between the service endpoints on Leaf2 and those on Leaf3 and Leaf4 as unreachable, as shown in Figure 19.

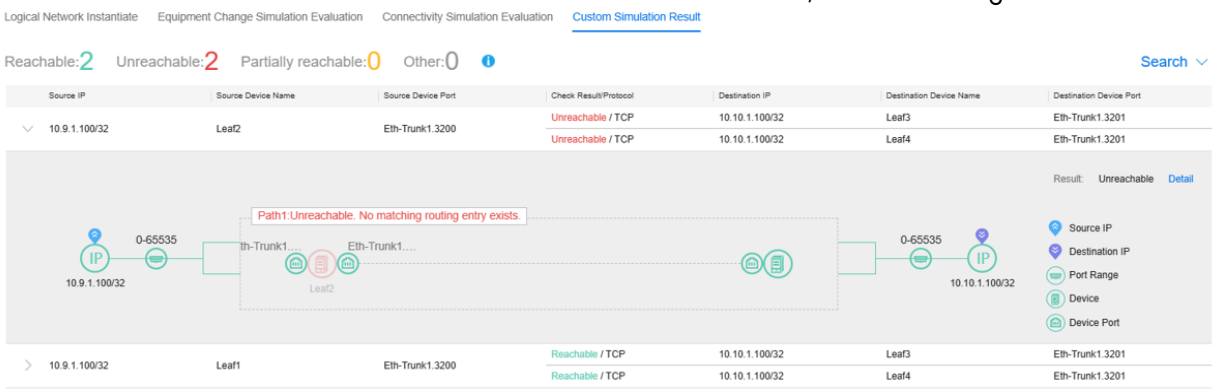


Figure 19: Simulation Report - Connectivity Results with Link Failure

The results of the previous and current tests demonstrated that NCE-Fabric supports the recommendation of network configurations for application provisioning, combined with detailed simulation and failure detection. This aligns with the Level 4 requirements for the Solution Generation capability, as defined in ETSI GR ENI 049.

2.4. Application Scheme Implementation

The simulation in the previous test confirmed that the configurations could be reliably deployed to the network devices. Based on this outcome, we delivered the configuration scheme to the devices. Deployment began with App3, which was successfully applied, as shown by the solution delivery result in Figure 20.

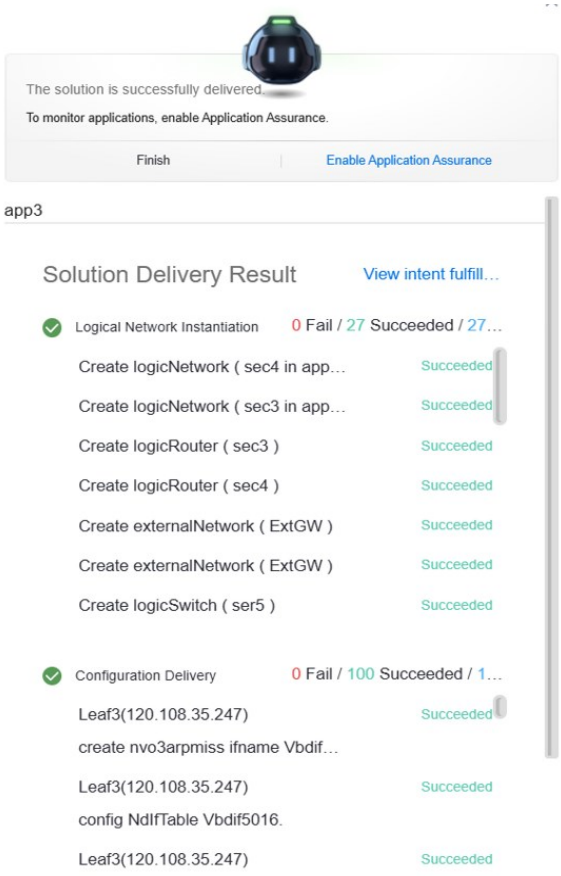


Figure 20: App3 Deployment Report

The deployment of App1 and App2 and the inter-application communication between them were also deployed successfully.

To verify the correct deployment of the applications and inter-application communication, we deployed virtual machines connected to the different leaf nodes where the services were deployed, and used ICMP to verify the connectivity. All pings were successful, indicating the proper deployment of the services.

To extend the test further, we defined an additional intent for the application (App6), which includes a firewall connected to the border leaf. This application consists of a single security zone (sec7). NCE-Fabric managed this intent, generated a solution, and deployed it successfully, just like the other applications.

It's worth noting that the firewall had to be initially onboarded manually, as NCE-Fabric currently does not support zero-touch provisioning for firewalls due to their complexity.

Configuration Backup

Creating a configuration backup and being able to roll the network back to a previous state are valuable in cases of unexpected behavior or failed changes.

NCE-Fabric supports both periodic and manual backup creation. After deploying all applications to the network, we created two manual snapshots for our test. One snapshot was created at the tenant level, capturing the state of all previously configured applications. The second snapshot covered the entire network, providing a complete backup of the deployed configuration across the fabric.

After successfully creating the snapshots, we rolled back the configuration related to the inter-application connectivity between App1 and App2. To verify the rollback, we performed a ping test between the two applications, which failed as expected, confirming that the rollback removed the relevant connectivity settings.

We then restored the tenant configuration from the previously created snapshot. The restoration was completed successfully, and connectivity between App1 and App2 was re-established. We verified this by performing a ping test, which confirmed that communication between the applications was functioning as expected.

To test the network-wide backup, we took App3 offline, including all associated services, and then performed a restoration using the network-wide configuration snapshot. The restoration was completed successfully, as shown in Figure 21, and the connectivity for App3 was restored, as confirmed by a successful ping test.

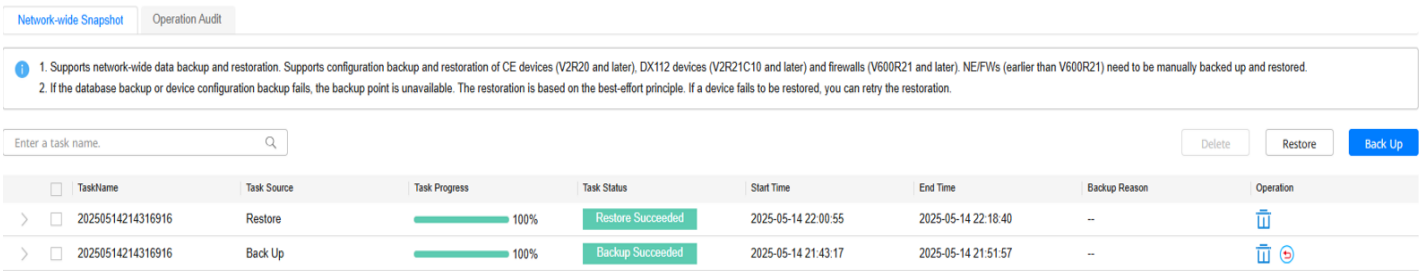


Figure 21: Backup and Restoration Log

However, it's important to note that network-wide backup restoration is intended as an emergency operation for critical situations, such as major outages or unresolved faults. During the process, NCE-Fabric also reverted its own configuration to the snapshot state, making it temporarily unreachable.

This test confirms that NCE-Fabric fulfills the Level 4 requirement for the Solution Implementation capability, as defined in the ETSI GR ENI 049 specification.

2.5. Application Service Verification

With the applications successfully deployed and running, we moved on to verifying NCE-Fabric's ability to monitor the services and evaluate risks. This test focused on NCE-FabricInsight, a standalone component specialized in monitoring and analysis.

NCE-FabricInsight provided a network-wide risk overview. The displayed risks were not limited to application-level issues but included general network risks as shown in figure 22.

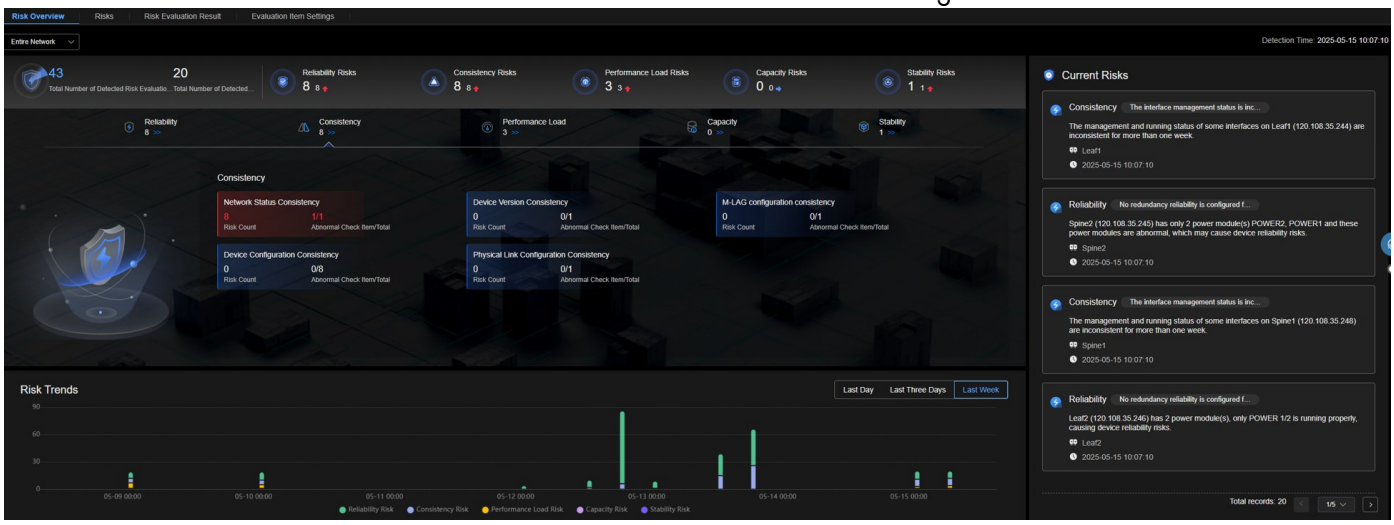


Figure 22: Risks Overview

The risks were categorized into five areas: reliability, consistency, performance load, capacity, and stability. In our test, the detected risks included redundancy issues related to power supply units and physical interfaces on certain devices that were operationally down but administratively up. An additional risk was related to load balancing. NCE-FabricInsight detected uneven traffic distribution across uplinks and suggested a general handling action, such as optimizing the hashing algorithm.

However, this risk was triggered by a reported 100% utilization on interface 100GE1/0/2 of Leaf2 over an extended period. At that point, no traffic was being generated into the fabric, so such high utilization was unexpected. Huawei later confirmed that this was a known issue in that version of NCE and that it had already been resolved in a newer version. Another risk reported by NCE-FabricInsight was about missing Layer 3 interfaces between a leaf switch and Spine1. This was due to unused connected links.

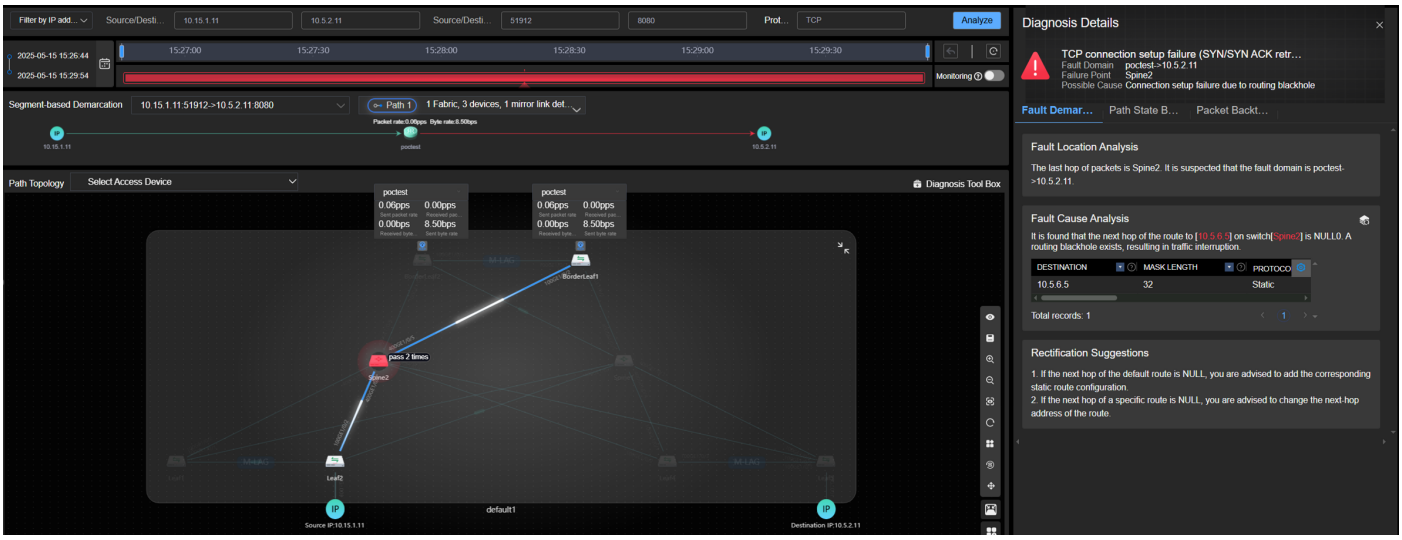


Figure 23: End-to-End Path Diagnostic View

To verify NCE-FabricInsight's ability to detect service-level issues and analyze end-to-end communication paths, we introduced a routing black hole on Spine2. This fault was intended to disrupt traffic to an endpoint connected to Leaf3. We generated iPerf traffic between virtual machines on Leaf2 and Leaf3, ensuring that the traffic would flow over the affected path. NCE-FabricInsight detected the issue and produced a detailed analysis. It identified the impacted IP addresses and highlighted the route causing the disruption. Additionally, the platform presented an end-to-end path view showing the health status of each segment, the failure domain, and the suspected root cause of the interruption.

In the application health status overview, the affected services were highlighted, showing the issue in the path between them, as shown in Figure 24 below.

Figure 24: Service-Level Health Status Overview

Selecting the icon for sec4 in app3 (Security Zone 3 in Application 3), shown in the previous figure, provided more detailed information about the health status of the security zone and the nature of the detected issue. NCE-FabricInsight marked the application state as an "Exception," indicating the presence of one issue. The health trend timeline clearly showed when the disruption began. The reported issue was related to connection setup, with a 0% connection success rate during the observed interval and a 100% server no-response rate.

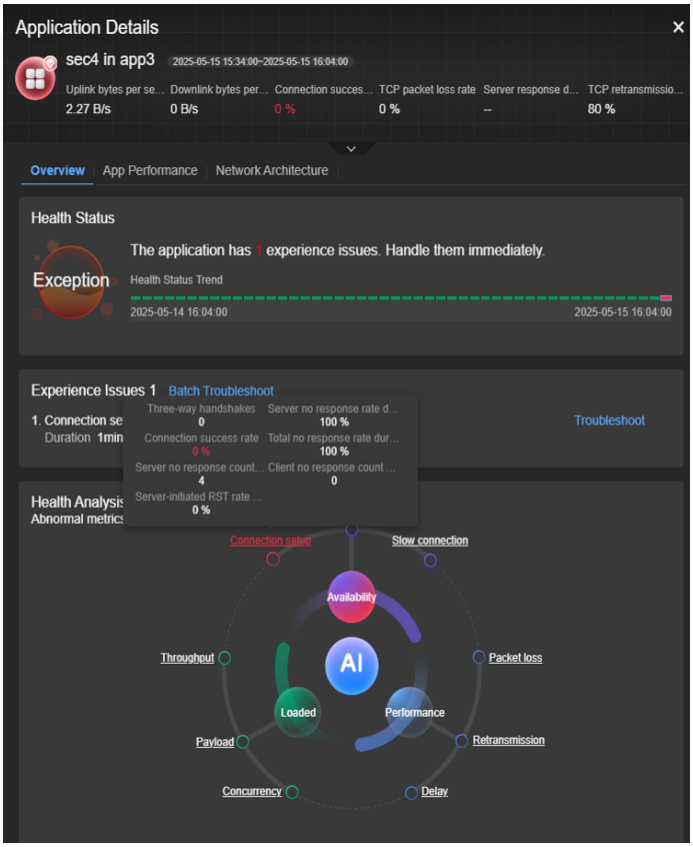


Figure 25: Application Health Status

The results of this test verified NCE-FabricInsight's ability to monitor application services, detect service-level issues, and evaluate network risks in real time. Its diagnostic capabilities and end-to-end path visibility fulfill the Level 4 requirements for the Verification capability, as defined in ETSI GR ENI 049.

3. Network Monitoring and Troubleshooting

Monitoring is essential for managing and maintaining network infrastructure. It gives visibility into device health and application behavior and plays a key role in detecting faults, troubleshooting performance issues, and responding to risks more effectively.

As networks grow in size and complexity—often spanning multiple domains and technologies—manual monitoring and troubleshooting become increasingly impractical. A good monitoring system helps by continuously checking the network's status, spotting unusual behavior, and identifying risks before they cause real problems. It should also support performance tracking, capacity management, and ensure service-level agreements (SLAs) are met.

When monitoring is combined with automation, it becomes even more powerful. Automated responses help reduce manual effort, speed up reaction times,

and maintain consistency. They also enable handling problems outside of business hours, which is critical in large, always-on environments.

The ETSI GR ENI 049 document defines procedures and capability levels for systems that enable automated monitoring, issue detection, and troubleshooting in intent-based networks. We will evaluate these capabilities in detail throughout the coming tests.

3.1. Intent Interpretation and Network Observation

Network monitoring typically involves defining multiple tasks and deploying mechanisms to measure latency, loss, and availability metrics. These mechanisms require careful planning and must be consistently executed across devices. Collected data must then be exported to a central interface, where it can be interpreted in the context of service performance. This process is often complex and resource-intensive.

In this test, we evaluated NCE-FabricInsight's ability to simplify and automate that workflow by processing high-level monitoring intents and translating them into executable monitoring tasks.

NCE-FabricInsight supports two levels of monitoring. The first is Data Plane Verification (DPV), which checks device configurations and operational state using collected configuration files and gNMI telemetry.

The second level is Assurance Service Monitoring, where traffic performance is observed in real time. This is achieved using port mirroring and Encapsulated Remote Switch Port Analyzer (ERSPAN). Port mirroring is typically configured on border leaf switches, where selected traffic is mirrored to a probe server for analysis. ERSPAN forwards the mirrored traffic—often limited to TCP headers—over GRE tunnels, allowing the probe server to reconstruct TCP flows, track sequence numbers, and detect issues such as retransmissions or connection setup failures.

Additionally, NCE-FabricInsight supports In-situ Flow Information Telemetry (IFIT) for direct measurement of network path performance. IFIT inserts telemetry metadata into live data packets, enabling hop-by-hop tracking of delay, jitter, and packet loss without requiring active test traffic.

To test the platform's intent processing, we defined a monitoring intent for a selected IP address pair in security zones 3 and 4 as shown in Figure 26 below.

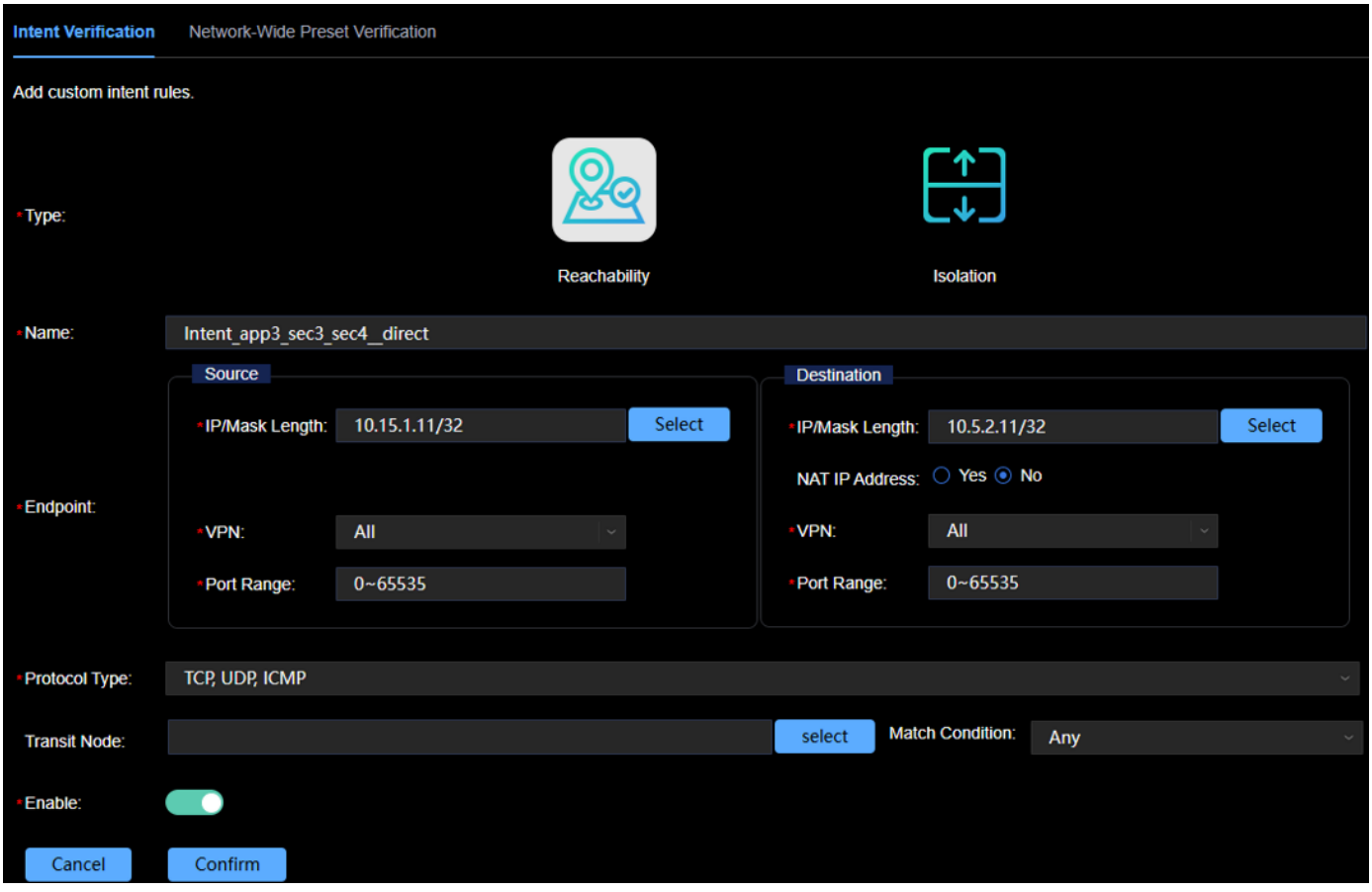


Figure 26: Monitoring Intent

NCE-Fabric processed the defined intent and analyzed all available paths between the selected endpoints. A total of 16 paths were identified, each going through different spine and border leaf node combinations. All paths were marked as reachable, indicating no connectivity issues between the defined IP addresses.

Figure 27: Reachability Status Between the Defined Endpoint

We then extended the intent by creating an assured application based on the defined IP pair. This activated SLA monitoring, allowing us to define thresholds for metrics such as connection success rate, latency, and setup time. In addition to these, NCE-FabricInsight supports detailed health monitoring across multiple categories, including Connection Setup, Congestion Control, Packet Loss, Retransmission, Delay, Throughput, Payload, and Concurrency, each with various parameters and configurable thresholds, as shown in Figure 28.

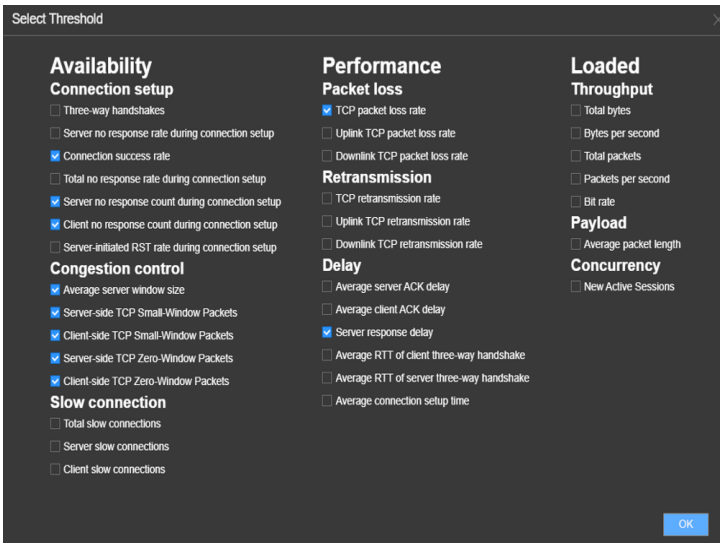


Figure 28: SLA Threshold Configuration Options

In addition to the previously defined IPv4 monitoring intent between two IP addresses in the security zones 3 and 4, we created a second intent using an IPv6 address pair for the same security zones. Both intents were processed similarly, with NCE-FabricInsight enabling path analysis and SLA monitoring for each flow. Figure 29 illustrates the exact path taken by traffic associated with the monitored IPv6 addresses, including the intermediate hops.

The traffic was forwarded from the spine to the border leaf and then back to the spine. This reflects the expected behavior for inter-security zone communication, as this type of traffic typically passes through the border leaf.

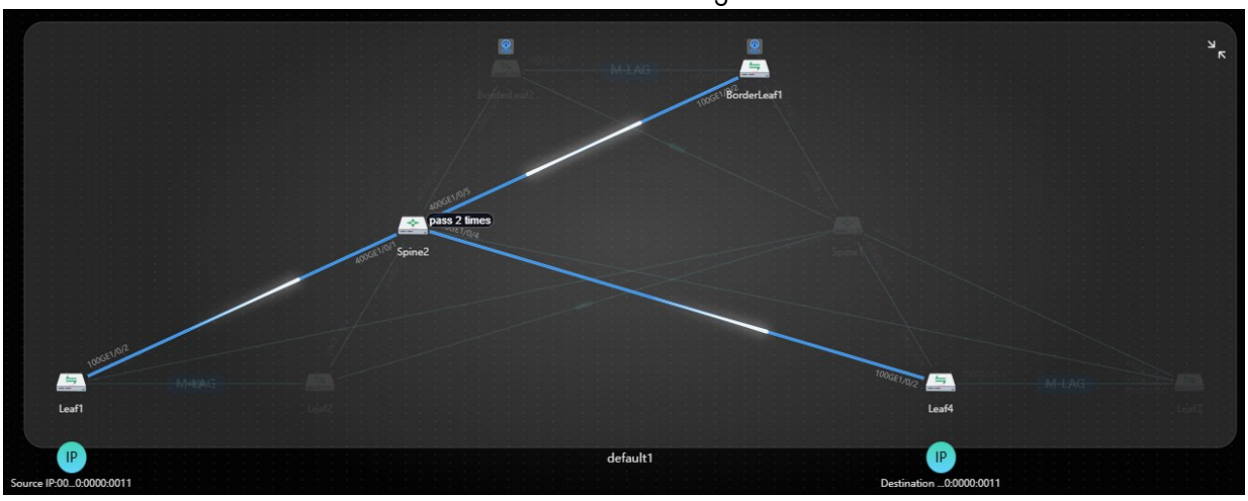


Figure 29: Traffic Path of the Monitored IPv6 Addresses Pair

It is worth noting that the association between applications and the IP addresses to be monitored must be defined manually. Automatic detection of application-level flows is only supported when a load balancer is present in the network design. Additionally, port mirroring is configured by default only for IPv4 traffic and does not include IPv6. Manual port mirroring must be configured to enable end-to-end path visibility and route condition analysis for the previously created IPv6-based intent. This mirroring still uses the existing IPv4 GRE tunnel, so no additional tunnel setup is required. However, enabling IPv6 monitoring does require additional ACL entries to cover mirrored traffic types, including VXLAN, IPv4, and IPv6.

This test verified that NCE-FabricInsight automatically translated the defined intent into concrete monitoring tasks. It also generated SLA tracking for metrics such as latency, connection setup success rate, and retransmissions and activated a range of health indicators, each with configurable thresholds. Although IPv6 monitoring required manual configuration, including port mirroring and ACL setup, this did not affect the platform's ability to perform the monitoring tasks. These observations confirm that NCE-FabricInsight supports the Scenario-Based Monitoring capability as defined in the ETSI GR ENI 049 specification.

3.2. Fault Detection and Rectification Decision-Making

In the previous tests, we verified NCE-FabricInsight's ability to define intent-based monitoring tasks and track the health status of services and traffic paths. In this test, we aim to evaluate its capability to detect network issues, suggest appropriate solutions, and, where applicable, resolve them automatically.

To achieve this, we introduced multiple issues into the network and monitored the response of NCE-FabricInsight.

Packetloss

In this scenario, we installed an impairment device between BorderLeaf1 and Spine1, using a 10G link due to the impairment device's interface limitation. The original 100G link was temporarily replaced to enable this test scenario. The telemetry data was collected using In-situ Flow Information Telemetry (IFIT). It is important to note that IFIT collects data only for L3 interfaces, so telemetry for Leaf-Spine traffic was not included in this test.

The impairment was configured as unidirectional packet loss at 10% for all traffic streams on the link. NCE-FabricInsight reported a packet loss rate of 7%, which is reasonably close given the passive nature of telemetry collection.

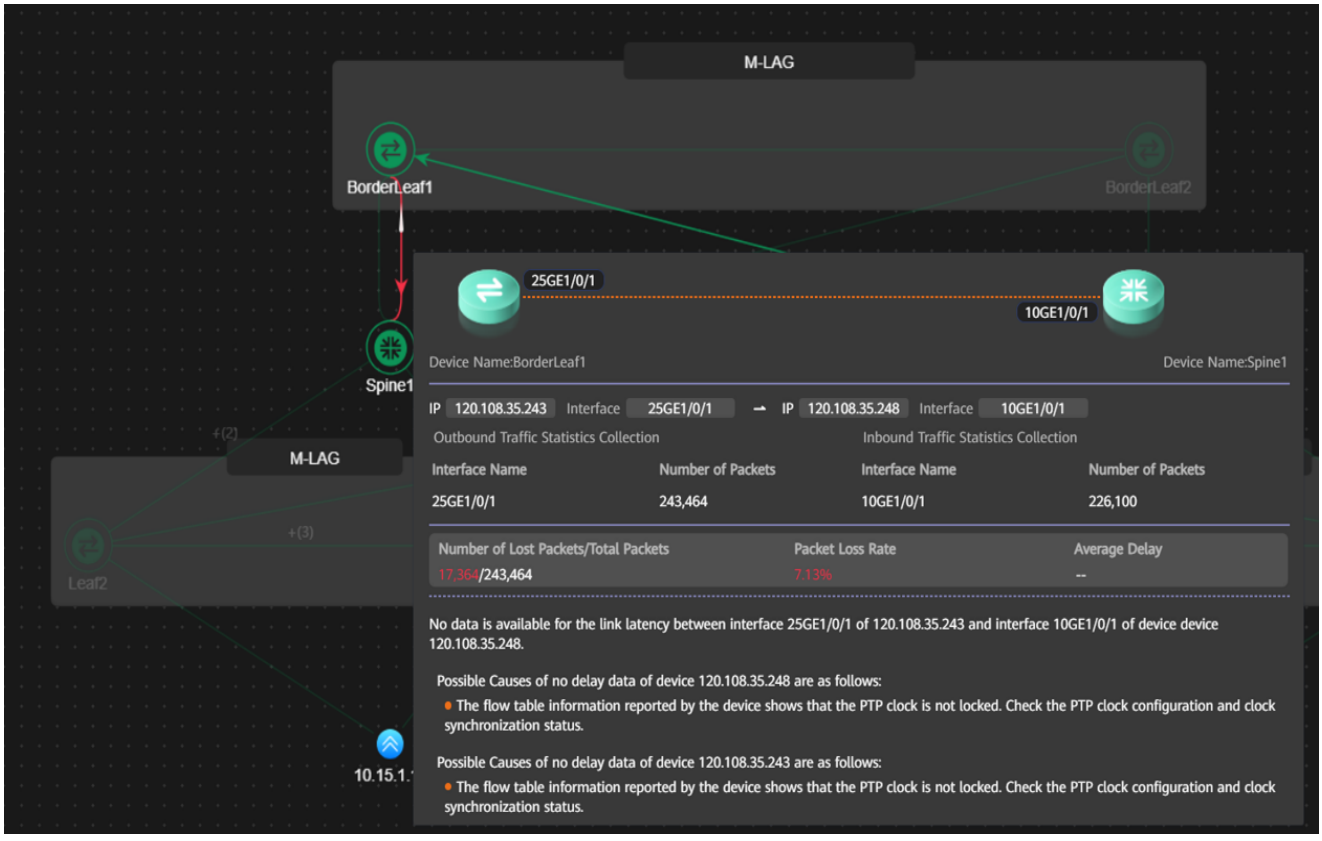


Figure 30: Link Operational Metrics - Packet Loss Rate

Latency data was not available, as Precision Time Protocol (PTP) was not enabled on the network devices. The system also did not provide a specific root cause for the packet loss, as link-based impairments are not directly linked to application flows. For comparison, a hardware issue such as an SFP with low optical power would have been reported under the system's issue detection logic.

Link Flapping

We introduced link flapping to the network by repeatedly shutting down and re-enabling interface 1/0/1 on Leaf2, which connects to Spine1. The simulation consisted of ten shutdown and undo shutdown cycles, each spaced 10 seconds apart.

NCE-FabricInsight is configured to detect flapping if ten such events occur within a three-minute window. This threshold is adjustable. During the initial run, the device timestamps were inconsistent due to incorrect time zone settings and missing NTP synchronization, which affected detection accuracy.

The test was repeated after Huawei configured NTP on the devices, using NCE-FabricInsight as the time source.

The system correctly identified the issue this time, reporting two issues, one related to the leaf switch and one related to the spine, as shown in Figure 31.

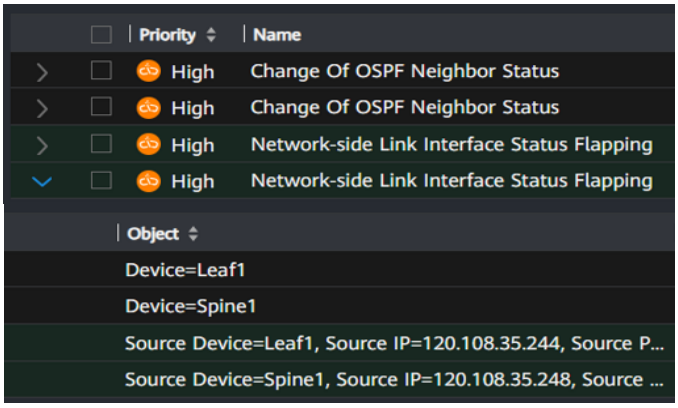


Figure 31: Link Flapping Issue

Expanding the issue shown in the previous figure provided additional details, including a repair recommendation, as illustrated in Figure 32.

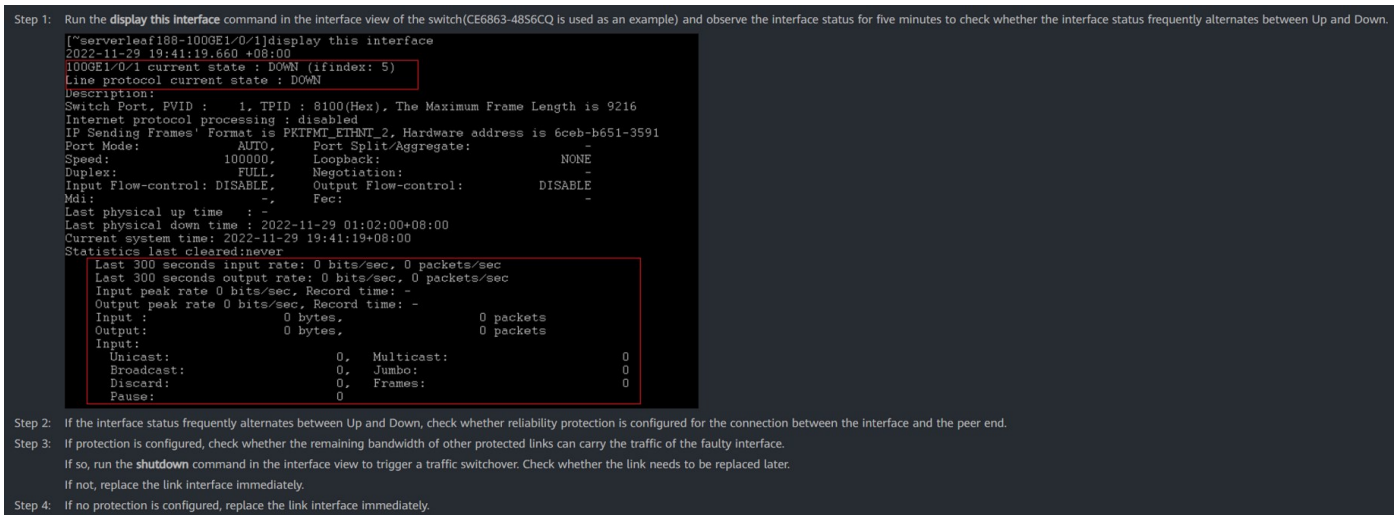


Figure 32: Link Flapping Repair Advice

The provided repair advice focused on manual steps to resolve the issue, which is helpful in environments where only NCE-FabricInsight is deployed.

NCE-FabricInsight redirected us to NCE-Fabric to automatically resolve the issue. Two options were shown: shutting down the interface on either the leaf or the spine, as shown in Figure 33.

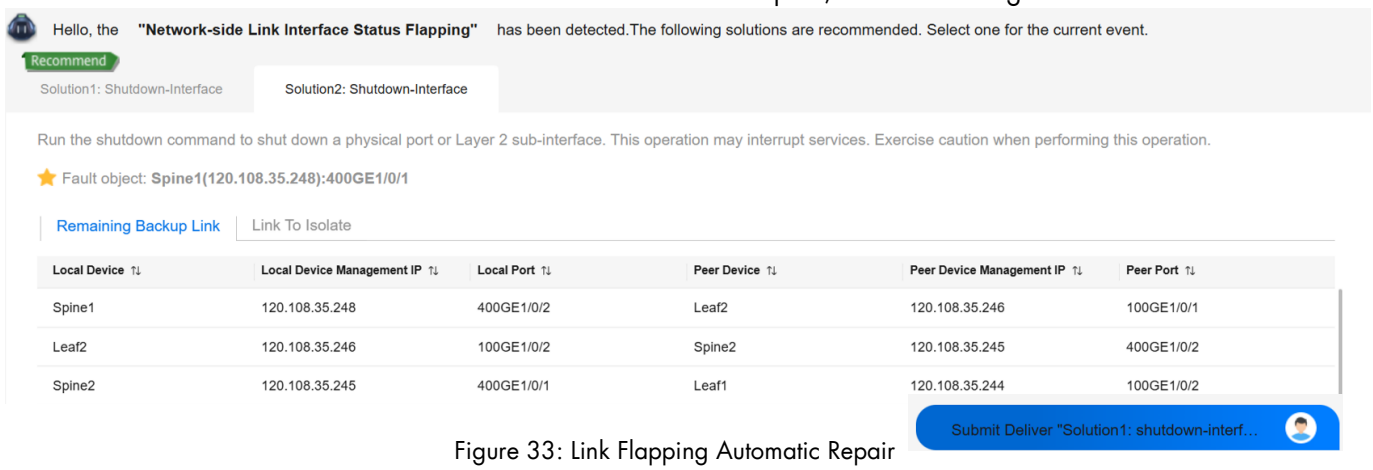


Figure 33: Link Flapping Automatic Repair

We proceeded with one of the options. The selected configuration was successfully delivered, and the system also analyzed the remaining backup paths.

After the solution was delivered, a rollback option was available in case the implemented action did not resolve the issue. In our case, we rolled back the solution configuration because the issue was artificially introduced. The rollback went without any problems.

The conversation between the user and the NCE-Fabric shown in Figure 34 is not text-based; the user does not type the messages highlighted in blue. Instead, these messages appear automatically after selecting one of the predefined options displayed beneath the white-highlighted messages.

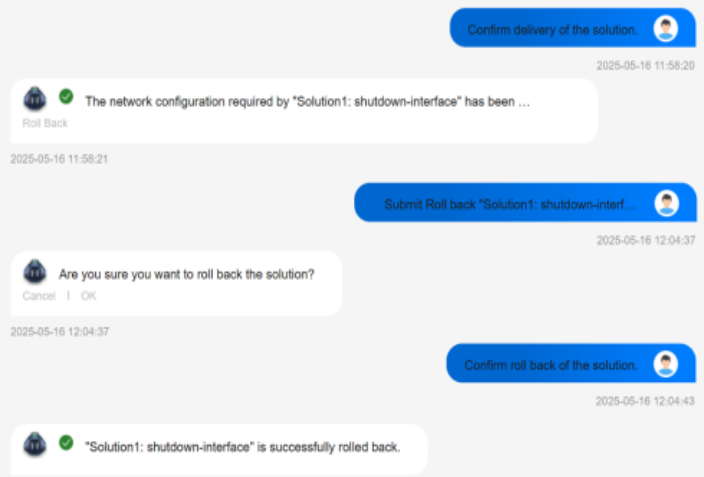


Figure 34: Rollback the Automatic Repair Change

IP Conflict

We created another issue by assigning the same IP address to two virtual machines—one on Leaf3 and one on Leaf4—both part of a Multi-Chassis Link Aggregation (MLAG) configuration. NCE-FabricInsight detected the duplicate IP issue as shown in Figure 35

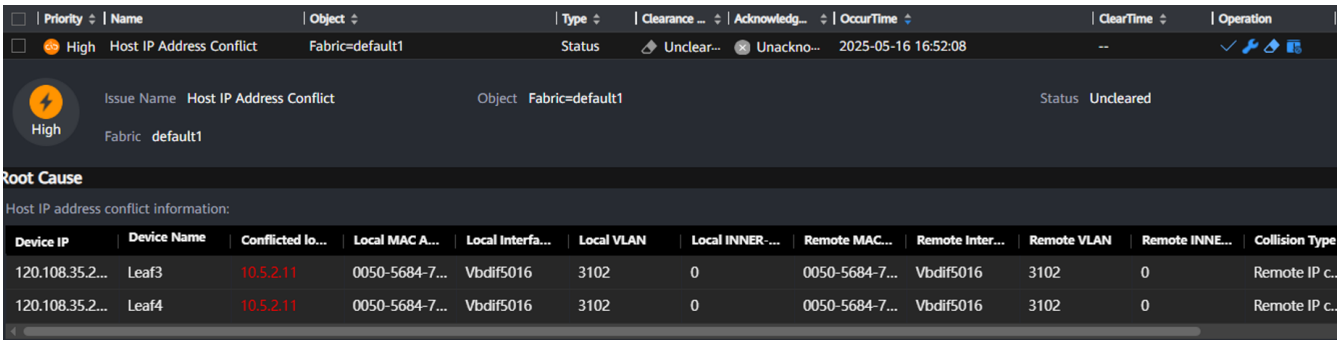


Figure 35: Duplicated IP Address Issue

The system also suggested recommended actions to resolve the issue, including multiple steps and specific CLI commands as shown in Figure 36.

However, in this case, automatic delivery of the solution was not supported, likely due to the operational risks associated with modifying active MLAG configurations.

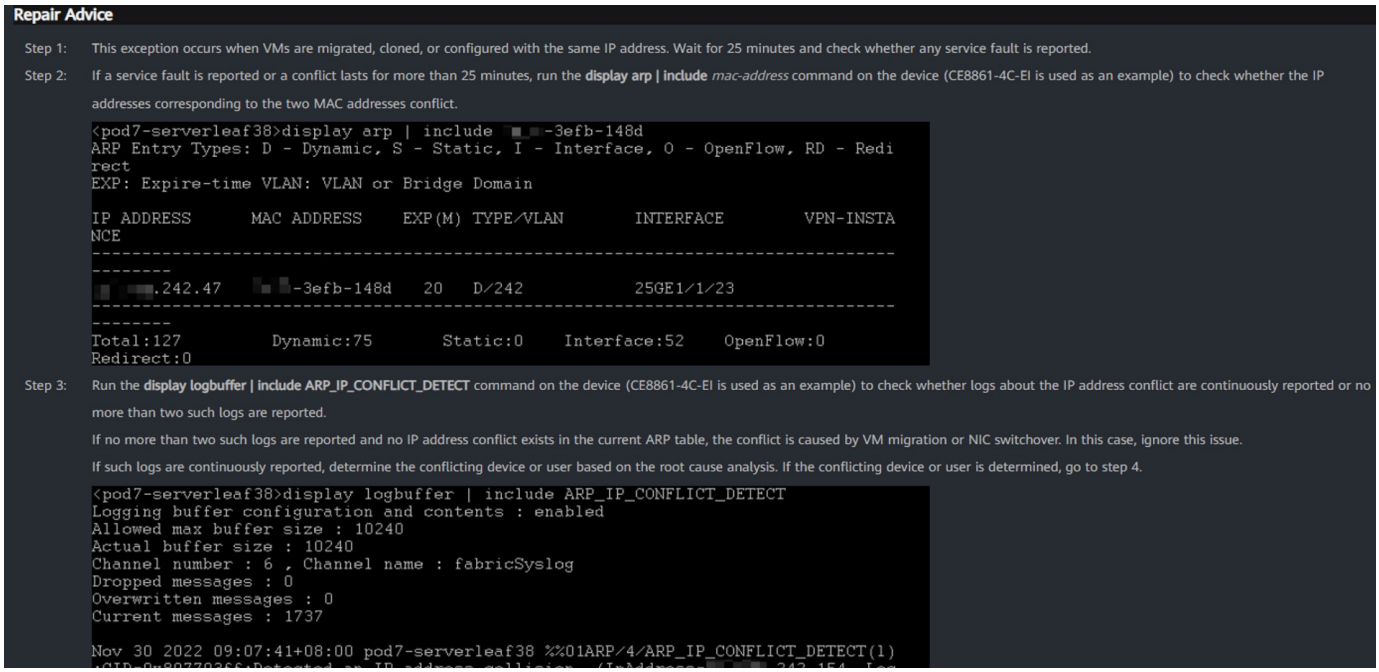


Figure 36: Duplicated IP Address Repair Advice

Routing Loop

To simulate a routing loop, we manually configured a static route on Spine2 that conflicted with an OSPF-learned route on the border leaf. This inconsistency created a routing loop in the network. NCE-FabricInsight successfully detected the loop and reported it. The diagnostic output included a basic recommendation to verify the static routing configuration. However, no automated solution delivery was available for this issue, and resolution required manual action.

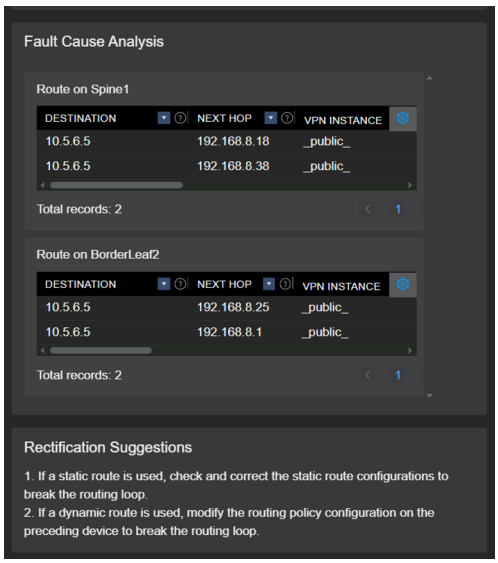


Figure 37: Routing Loop Analysis and Repair Suggestion

Security Policy Conflict

In this test, we configured a firewall rule to deny traffic to a specific address. NCE-FabricInsight detected the resulting connectivity issue and correctly flagged the traffic path as unreachable. The system identified the root cause as security policy-based filtering, as shown in the intercepted packets section. It provided detailed context, including source and destination IP addresses. The corresponding security rule "ENTAC" was highlighted in the configuration diff view, where the action was set to deny for the specified destination address.

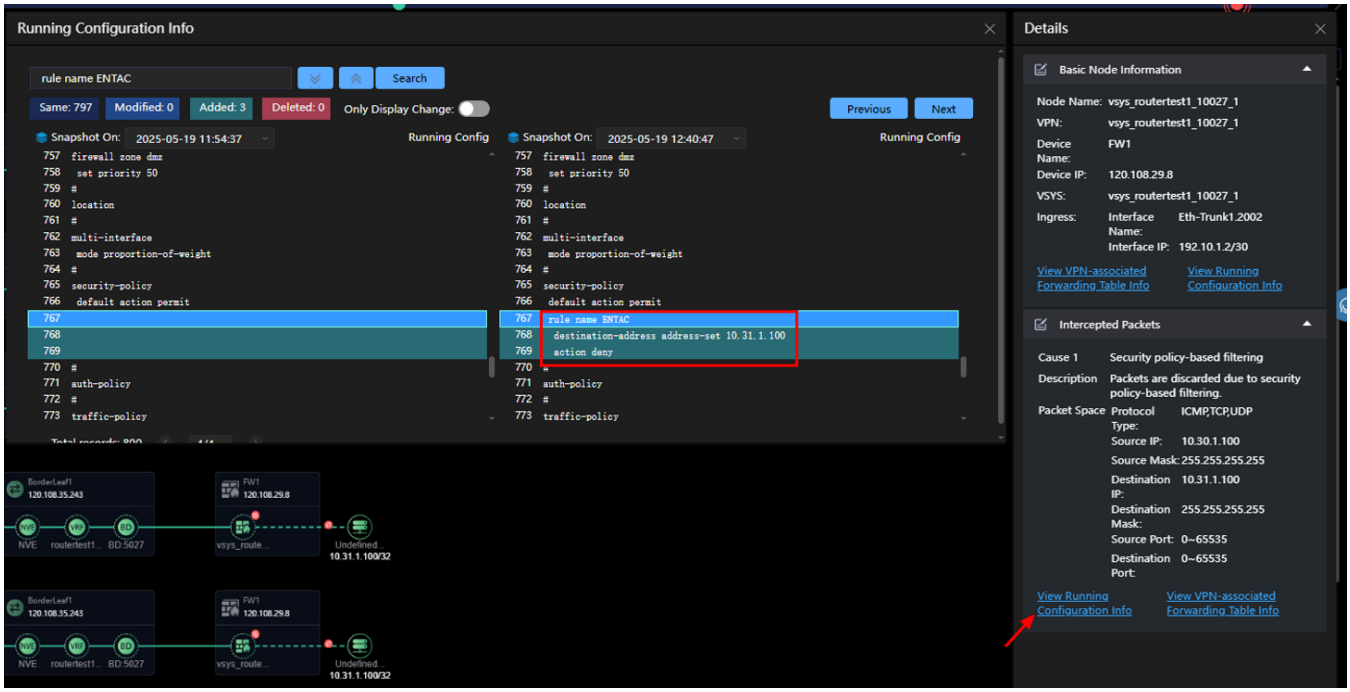


Figure 38: Security Policy Issue

Similar to the suggestion in the previous issue, NCE-FabricInsight provided repair guidance, instructing the user to check the security configurations by clicking View Running Configuration Info in the Details area to verify whether the device had rules blocking packets. While the resolution was not delivered automatically, the system offered clear and actionable guidance that helped identify the misconfigured policy.

Faulty Optical

To emulate a faulty optical module or fiber, we connected a traffic generator to a border leaf switch and configured the corresponding routing interface. The traffic generator was set to inject FEC (Forward Error Correction) errors into the transmitted frames, simulating the behavior of a degraded optical link. NCE-FabricInsight successfully detected the issue, identified the exact physical interface where the errors were observed, and provided repair suggestions, such as cleaning the optical module or replacing it if the problem persisted.

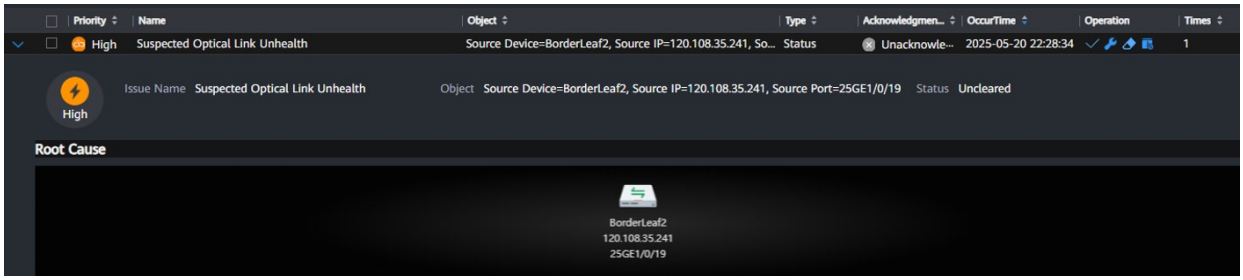


Figure 39: Faulty Optical Module Issue

NCE-FabricInsight successfully detected all the issues we introduced into the network and provided detailed diagnostic information, including rectification suggestions with CLI commands and step-by-step guidance. However, automatic implementation of these solutions was only supported in the case of link flapping. The platform required manual actions for other issues, such as IP conflicts and routing loops.

Additionally, NCE-FabricInsight does not support simulating the proposed solutions before applying them, which limits its ability to meet the requirements for full automation.

Huawei stated that the NCE-Fabric, in collaboration with the NCE-FabricInsight, already supports a variety of issues for which an automatic repair can be generated, simulated, and implemented. The following table shows a full list of those issues and the rectification solution.

Event Name	Rectification Plan
Intermittent Link Disconnection	Isolate the ports at both ends of the link.
Unidirectional Link Connectivity Fault on Network Side of a Switch	Isolate the port.
Switch Physical Port Suspended	Isolate the physical ports of the switch.
Suspected Optical Link Fault	Isolate the port.
Repeated Switch LPU Fault	Isolate LPUs.
Traffic Exception Caused by Switch Entry Inconsistency Between the Software Table and Hardware Table	Re-deliver specified entries in the software table to the hardware table. Restart the board. Restart the switch.
Traffic Exception Caused by Lost Switch Routing Hardware Tables	Re-deliver specified entries in the software table to the hardware table. Restart the board. Restart the switch.
Neighbor Relationship Flapping Due to an Incorrect Update Packet Received by Switch	Isolate the peer.
IP Address Conflict on Network Access Side	Display the conflict IP address or the port of the conflict IP address. Then isolate the specific IP address or port.
Suspicious Layer 2 Loop	Isolate the port.
Server Access Fault	Restart the port.

Table 3: List of issues and the rectification solution — Part 1

Event Name	Rectification Plan
Repeated Switch Restarts	Isolate the switch.
Two Master Switches in M-LAG	Isolate the switch.
The protocol status of the port is down	Isolate the port.
TCP SYN Flood Attack	Isolate the VM. Isolate the port.
ARP Attack	Isolate the VM. Isolate the port.
ND Attack	Isolate the VM. Isolate the port.
Flow Exception Caused by CE Switch Chip Soft Failure	Restart the switch.
OSPF Router ID Conflict	Configure a router ID for the device.
Designated Router IP Address Conflict	Reconfigure the IP address of the interface where a conflict occurs.
IP Address Conflict on Network Side	Shut down the port.
Invalid ARP Packet Received by Switch	Shut down the port and isolate the IP address.

Table 3: List of issues and the rectification solution — Part 2

However, due to time constraints, those have not been tested in the scope of this evaluation.

Based on the observations in this test, NCE-FabricInsight fulfills Level 3 requirements for the capabilities of Fault Diagnosis, Solution Generation and Decision-Making, and Solution Implementation as defined in the ETSI GR ENI 049 specification.

3.3. Post-Fault Service Verification

As observed in previous tests, NCE-FabricInsight supports both risk evaluation and continuous monitoring of network health status. Snapshots can be created manually or scheduled at regular intervals. An autosave feature is available, which automatically backs up the configuration after each commit operation. The configuration comparison tool presents differences in a clear and readable format, with an option to highlight only the modified elements, simplifying change tracking and review as shown in Figure 40.

Figure 40: Snapshot Configuration Change

For specific parameters such as the interfaces and routing protocols, NCE-FabricInsight presents configuration differences in a clear and visually accessible format, highlighting only the changes directly within the interface, as shown in Figure 41. Thus, it eliminates the need to manually inspect CLI-based configuration files.

NCE-FabricInsight demonstrated support for monitoring network health, evaluating risks, and managing configuration changes. With these abilities, NCE-FabricInsight meets the requirements for Level 4 of the Implementation Verification capability, as defined in the ETSI GR ENI 049 specification.

Figure 41: Interface Change Comparison

Executive Summary

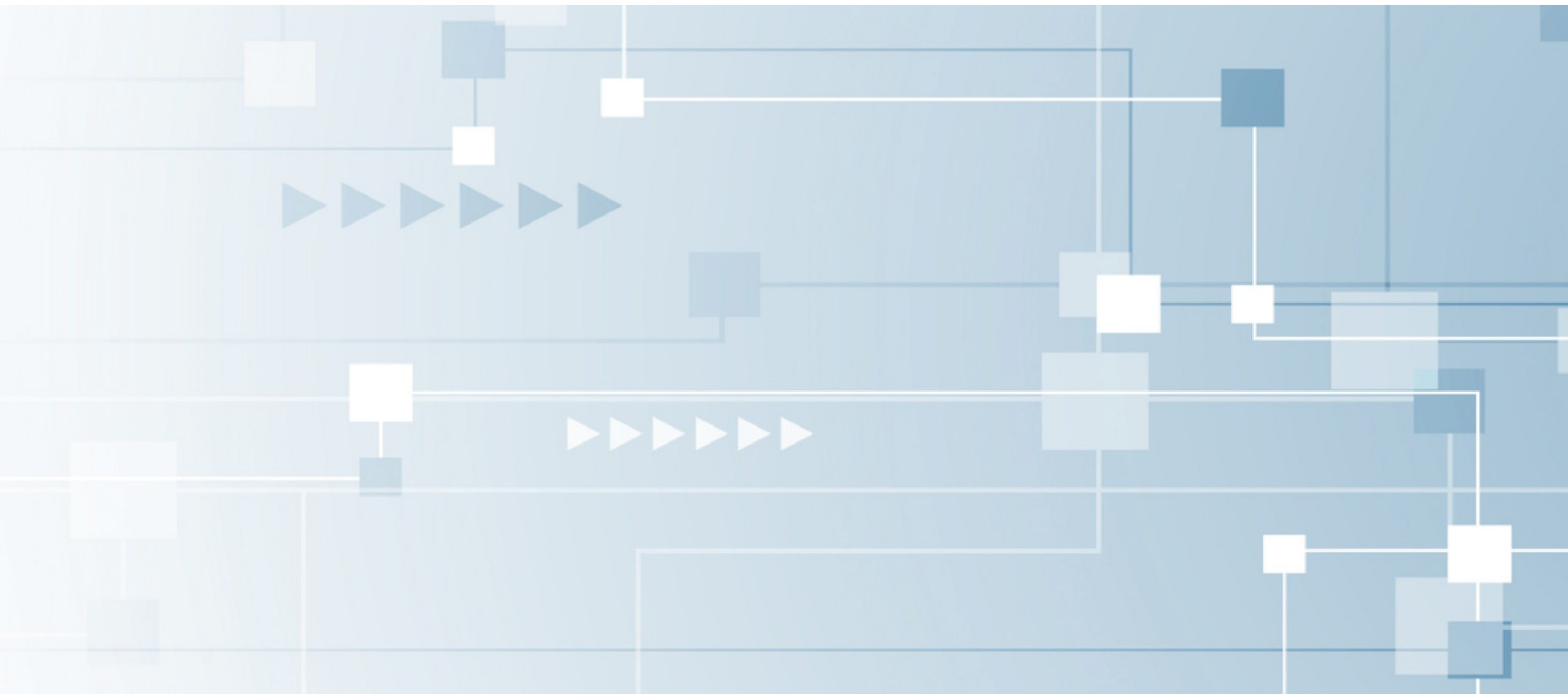
The Huawei Data Center Autonomous Driving Network Solution, implemented by NCE Fabric and NCE FabricInsight, has shown excellent support of AN L4 features as defined by the TM Forum and ETSI ENI 049. In total, the solution exhibited an average level of L3.9 (see evaluation matrix in Table 2 below). In a 2-tier data center network design with Huawei CE9855 and CE688x routers using EVPN VXLAN services, almost all provisioning and monitoring activities were performed autonomously:

- The data center pod was provisioned, including the underlay, bootstrapping from scratch with zero-touch provisioning methods. Many configuration aspects were provided correctly, including all management addressing and routing aspects. Automated clock synchronization provisioning was not supported. The overlay service provisioning worked perfectly for the typical data center EVPN use case scenarios we requested.
- The Huawei solution verified all suggested provisioning actions with a digital twin, simulating the effects of the intended configuration before acting on it. The digital twin is a crucial function to manage the risks associated with fully autonomous configuration. It worked very well in the tested scenarios.
- The application provisioning was implemented using an intent language to describe VXLAN services intent. This language permitted formulating the straightforward connectivity requests. As a side note, security intent was not supported and we did not require it, as security intent formulation for network services is way beyond the state of the art in the industry.
- For the application services, verification can be a difficult challenge because it relates to the application layer. Huawei implements an in-band service verification, which worked well during our assessment.
- For service monitoring, Huawei showed support for multiple telemetry methods, including in-band data monitoring via mirroring breakout. While this kind of monitoring naturally does not scale for all services provided, it can be used to monitor high-value services very closely, or to spot-check specific services representative of a larger group.
- We also verified alarm and log correlation across all routers. The system was able to correlate many log entries and correctly identify the root cause.
- Huawei has implemented the fault diagnosis and solution generation with an individual problem assessment that supports dozens of different problem types. It gives a good and fairly precise suggestion how to fix an issue, although it is not AI-supported and does not provide specifically, individually tailored solutions at this time.
- It was possible to automatically remediate some of the standard faults discovered by network monitoring, and to verify a correct remediation subsequently.

EANTC validated all functional aspects of AN L4, as required in the ETSI ENI 049 standard. We did not validate performance or service scale, as the test bed was sufficient but not very large. We also focused on the EVPN VXLAN architecture and did not evaluate other data center architectures. Data center interconnects and the integration into a larger end-to-end network were out of scope for our evaluation. Multi-vendor interoperability (how to manage third-party routers) was out of scope, too.

In summary, Huawei's solution has passed our tough audit with flying colors, resulting in an overall score of L3.9. The solution is very close to achieving L4.0; only one minor function needs to be amended. Huawei NCE Fabric and NCE Fabric Insight are certainly one of the industry-leading, advanced network automation solutions existing today.

At EANTC, we were particularly impressed by the 360-degree support of Autonomous Network L4 features from underlay provisioning through intent-based application rollout to monitoring and troubleshooting. The Huawei solution will help to accelerate data center installation and management substantially, while requiring less qualified staff during planning, rollout, and everyday operations.



EANTC AG
Salzufer 14, 10587 Berlin, Germany
info@eantc.de, https://www.eantc.de/
[v0.1 20250619]