Network Services Orchestrator (NSO) - MIX AND MATCH

VERSO ALTIMA

CHALLENGE

- Change network topology on the fly short period of time
- Effortlessly migrate to new equipment/OS type version/protocol
- Migrate to a new vendor
- Easily move customer services from another, from one device to another

Q: How many commands are required to switch a network link from Juniper device to Cisco device?
A: One to remove old link, one to add new link!

SOLUTION

+ + + Service Model
Once services have been modelled, NSO automatically calculates every step needed for service provisioning (add/change/delete/rollback)

+ + + Network Element Drivers
Inform NSO on how to provision specified service by using a minimal possible set of instructions

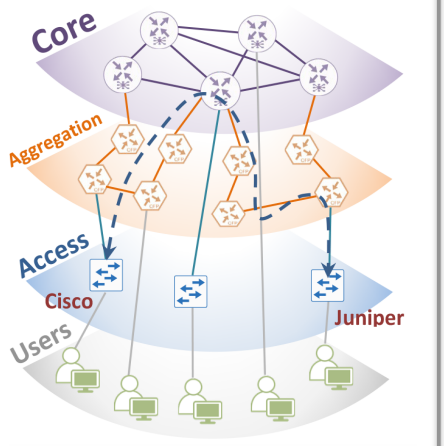
+ + + Uniform CLI
Use exactly the same commands to provision services without worrying about underlying equipment or OS

NSO translates a single service command into as many device commands are required to accomplish the task. This is why a complex action can be achieved with a single service command.

Mix and match use cases

+ Replace one vendor equipment with another vendor equipment.
+ Move access switch from one aggregation node to another.
+ Move customers from one-line card to another, or spread them over multiple different line cards that are on the same or different devices. Quickly mitigate effects of failed cards or modules.
+ Replace a card in chassis with a different model that has different interface naming convention.
+ Change pseudo-wire signaling from EVPN to LDP.
+ Change critical service parameters on the fly: BGP AS Number, routing protocol, IP addressing, VLANs and much more.

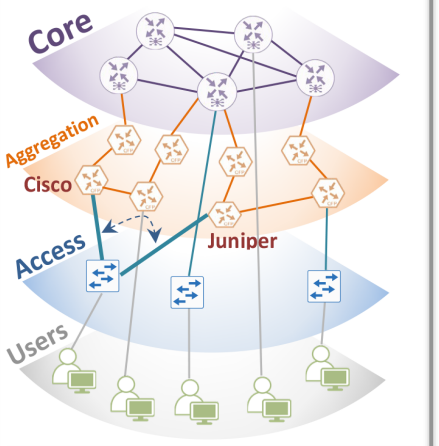
Change pseudo-wire signaling protocol




```
vpn l2vpn point-to-point TEST-P2P
signaling-type ldp
```

Yes, it takes only one command

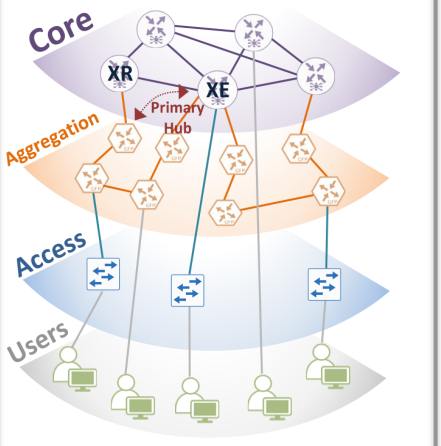
Move access switch from one aggregation node to another




```
vpn l2vpn node-mapping sw1
  no connected-endpoint ios0
  connected-endpoint jnpr1 ge-1/0/0
```

One command triggers all required actions on all affected devices

Move aggregation hub from one core node to another




```
vpn l2vpn node-mapping ios0
 no primary-hub xr1
 primary-hub xe1
```

One service command to effortlessly accomplish the task

NSO USE CASES – KEEP CONFIGURATIONS CLEAN

VERSO ALTIMA

CHALLENGE

- Keep device configurations minimal, clean and up to date
- Make sure that services are always correctly provisioned and deleted
- Keep the number of provisioning mistakes at a minimum
- Quickly revert configuration changes with a single command

SOLUTION

+ + + Model, Model, Model
Make the model as complex as necessary to keep service provisioning as easy as possible.

+ + + Atomic Provisioning
Never have half-deployed services. Automatically perform all or nothing provisioning. Make rollback trivial.

+ + + Programmable Intervention
Dynamically update configurations based on service changes.

CASE 1 – DELETE AN ENTIRE SERVICE WITH A SINGLE COMMAND

Using device CLI

```
ethernet cfm
 domain TEST-EVC level 4
  no service TEST-MP bridge group TEST-EVPN
  bridge-domain MP_TEST-MP_1
!
evpn
 no evi 1 ?
 !
 no virtual neighbor 10.10.100.10 pw-id 1 ?
!
l2vpn
 no pw-class EVC-PW ?
 !
 bridge group TEST-EVPN
  no bridge-domain MP_TEST-MP_1 ?
  !
   no interface GigabitEthernet0/0/0/1.10 ?
   !
   neighbor 10.10.100.10 pw-id 1 ?
```

? Operator needs to know the details for the device, OS and vendor

? What values should be used? Are they unique within the service, across services?

? Will some added/removed commands affect other services

? How do I know there are no leftover configuration items?

? What is the minimal set of commands I need to create?

Using NSO

```
# no service vpn l2vpn multipoint TEST-MP
```

Yes, a single NSO command deletes the entire service on all nodes and ensures there is no leftover configuration on any device

+ All-or-nothing config removal
+ Zero leftover service configuration on devices
+ Sends only minimal set of commands and always in the right order
+ Automatically checks dependencies, keeps track of configuration items used by other services
+ Removes only what needs to be removed, and only if no other service needs the item

CASE 2 – AUTOMATIC DESCRIPTION UPDATE

+ Have service description automatically reflect change in service parameters
+ Get exact time when the change is made, without any delay
+ Ensure that all descriptions are always 100% consistent
+ Start relying on interface descriptions for process automation, such as NMS activation
+ Automatically update back office systems with changed information
+ Great for automated NMS – every time service changes, NMS automatically updates

Service at 50 MBPS

```
vpn l2vpn multipoint TEST-MP
 endpoint ios0
  interface GigabitEthernet0/10
   vlan 100
    police rate 50 mbps
interface GigabitEthernet0/10
 service instance 100 ethernet TEST-MP
description "TEST-MP::50MBS::L2VPN-MP::VPN1::NSO"
```

The Same Service at 100 MBPS

```
vpn l2vpn multipoint TEST-MP
 endpoint ios0
  interface GigabitEthernet0/10
   vlan 100
    police rate 100 mbps
interface GigabitEthernet0/10
 service instance 100 ethernet TEST-MP
 description "TEST-MP::100MBPS::L2VPN-MP::VPN1::NSO"
```

NSO USE CASES – TEMPORARY CONFIGURATIONS

VERSO ALTIMA

CHALLENGE

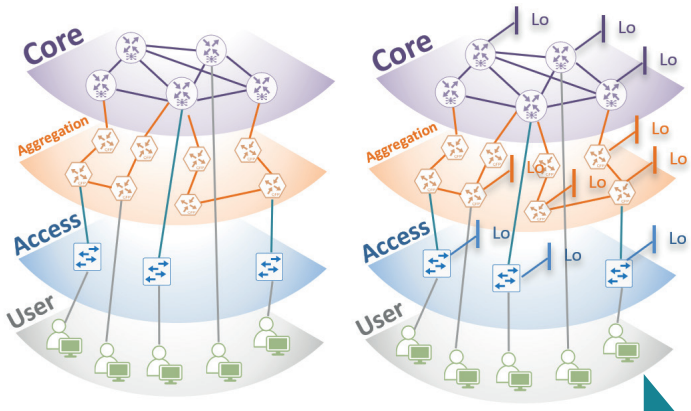
- Automatically verify L2 services after deployment
- Automatically create a document with proof of service at provisioning time

SCENARIO

+ Deploy L2 service
+ Temporarily create pingable L3 interfaces
+ Automatically perform full-mesh ping tests and store results
+ Create a report as a proof of service
+ Remove L3 pingable interfaces

+ Easily deployed as part of provisioning or at any point in service lifetime
+ Deployed with a single command to hundreds of endpoints – impossible to do manually
+ Zero leftover configuration after checks are made
+ Infinitely scalable, can be deployed for thousands of endpoints at a time
+ Does not require any detailed input from operator

Move from Service Topology to Test Topology and back with a single service command!



Yes, its that simple!

Other examples of use cases:

+ Work arround failures, software bugs until fixed
+ Easily activate and deactivate OAM configs
+ Accommodate temporary customer requests
+ Change between EVPN signalling and LDP signalling to quickly fix connectivity issues

Solution

✓ Create pingable interfaces

```
# vpn l2vpn muiltipoint TEST-MP
    create-pingable-interfaces
```

+ Requires a single command
+ Automatically creates as many loopbacks as required by service
+ Operator does not have to supply VPN parameters or IP addressing – Everything is calculated automatically!

✓ Ping as much as required

+ All service tests are automatically created
+ No need to run ping commands manually
+ Can be accessed from command line or by API
+ Automatically create report(s)

✓ Remove pingable interfaces

```
# vpn l2vpn muiltipoint TEST-MP
    no create-pingable-interfaces
```

+ Again, just a single command
+ No leftover configuration regardless of the actual service complexity

NSO USE CASES – SERVICE MIGRATION

CHALLENGE

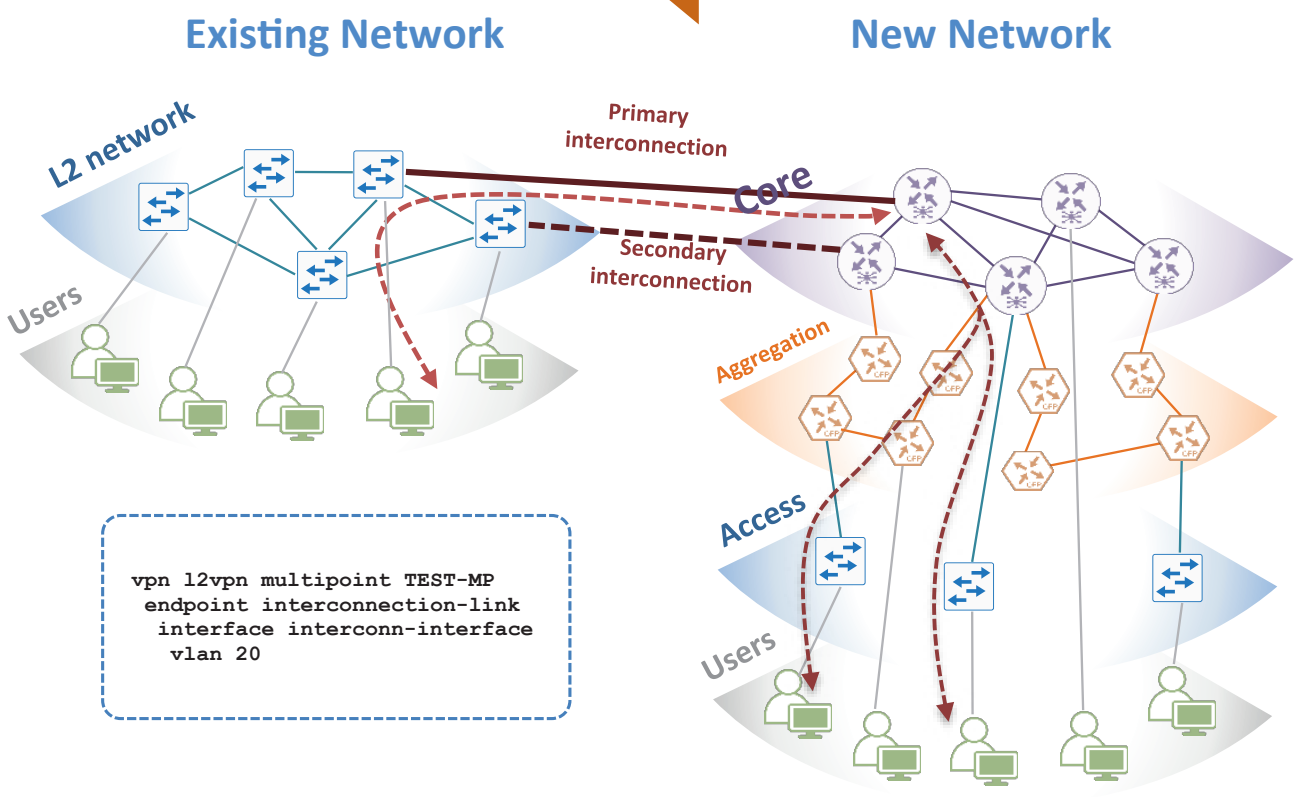
- Migrate services to completely new topology, site-by-site, customer-by-customer or in bulk
- Take as much time as you need - use step-by-step approach
- Run services transparently across both infrastructures until migration is complete
- Ensure zero down time for end users

SOLUTION

+ Deploy network interconnection points
+ Map services in the old network to the new network
+ Allow for gradual and transparent service migration
+ Keep all configurations in the old network until all services are migrated
+ Completely remove parts of configurations that are related to migration after decommissioning
+ Allow for quick rollback in case of problems

+ Easily move all services from primary to secondary interconnection link in case of failure.
  This cannot be done manually
+ Interconnection link is defined as a virtual endpoint. When a service is provisioned, operator selects "interconnection link" without worrying where it is physically connected.
  NSO takes care of configuration specifics
+ During provisioning, **NSO** performs service tests automatically. If something goes wrong, service can be rolled back at any point during migration
+ It is very easy to change endpoint parameters such as port, VLAN, IP address at any time during migration

Zero mistakes during migration




```
vpn l2vpn multipoint TEST-MP
 endpoint interconnection-link
  interface interconn-interface
   vlan 20
```

NSO USE CASES – UNIFIED CLI

VERSO ALTIMA

CHALLENGE

- Make sure that critical configuration is always 100% synchronized with service requirements
- Always use exactly the same service commands to provision or poll devices, without worry about vendor, OS or version specifics
- Work around bugs for uniform and complete show command output

SOLUTION

+ + + Security – Unwanted config changes are automatically reverted to proper state

+ + + Consistency – The same look and feel across platforms and vendors

+ + + Enforcement – Real-time service audits

Configuration enforcement

NSO provides so much more than simple compliance

✓ **Schedule periodic configuration checks and delete unauthorized changes**

For critical services, have complete control over any manual configuration changes that are not allowed

✓ **Enforce configuration rules for critical parts of configuration**

Make sure that firewall rules are not changeable through manual intervention, automatically roll back unwanted changes

✓ **Reverse the changes that are done outside of service provisioning framework**

NSO makes sure that most important parts of configuration are aligned with rules

Make sure that service requirements are always accurately reflected in specific device configuration

Enhance and unify show command output

- ✓ Use exactly the same show command on any network asset
- ✓ Make the output of show commands uniform across vendors, software types and versions
- ✓ Easily work around bugs and version differences
- ✓ Provide uniform input to automation scripts and monitoring tools
- ✓ Request real-time data from devices by always calling the same API for any device type
- ✓ API always presents output in the same format. Makes automation very easy.

```
# show devices device CSCORTR1 live-status ios-stats:interfaces
  TYPE                 NAME   STATUS  IP ADDRESS        MAC ADDRESS     DUPLEX  SPEED  MTU   TYPE
  ---------------------------------------------------------------------------------------------------------
  GigabitEthernet      0      up      10.10.10.90/24    848a.8d4b.7336  Full    100    1500  RJ45
  TenGigabitEthernet   0/0/8  up      10.255.255.1/30   848a.8d4b.7308  Full    10000  9198  SFP-ER
```