Developer Days
Automation
Cisco Transport Slice Automation Design & Deploy
- using Intent-Based Model-Driven Principles

John Mullooly - Cisco Distinguished Architect - Americas SP
11/30/23
What is Network Slicing?

An E2E service originally defined by the 5G community to Deliver Differentiated Services and new revenue streams initially targeting SP enterprise/business use-cases

Network Slicing is fundamentally an end-to-end partitioning of the network resources and network functions so that selected applications/services/connections may run in isolation from each other for a specific business purpose and meet a business level SLA, with the benefits of user simplicity and flexibility

It’s all about offering a business Outcome:

1) End to End Service Level Agreements (SLAs)
2) SLOs: Delay, jitter, loss, availability
3) SLEs: e.g. Disjoint paths, encrypted paths, regional avoidance, country sovereignty.

SLA = SLO + SLE
SLO = Service Level Objective (the “measurable” stuff)
SLE = Service Level Expectation (unmeasurable)

Critical Take-away: While Slicing was driven by 5G needs, it concept can also be used independently of 5G services for transport only. Thus, it becomes a generic automation solution for any transport service.
Why do SP’s care so much about 5G Network Slicing?

Network Slicing can drive additional revenue, value and relevance to the SP’s

**New Customers**
Increase adoption of 5G among Enterprises

**New Services**
Capture a larger share of the value chain

**New Business Models**
Premium pricing for demanding use cases
Defining Transport Slicing Scope:
3GPP reference architecture for 5G network slicing

NSMF = Network Slice Mgmt. Function
NSSMF = Network Slice Subnet Mgmt. Function
NST = Network Slice Template
NSST = Network Slice Subnet Template
NSI = Network Slice Instance
NSSI = Network Slice Subnet Instance

Scope of 5G *Network* slice management
Scope of *Transport* slice management
/** Per-Domain Slice Technologies 
* Many “knobs” (primitives) for driving per-domain slice behaviors */

- **QoS Flow Indicators for bearer prioritization**
- **Numerology for Orthogonal NR**
- **vRAN VNFs vCU/vDU**

- **PE-to-PE Traffic Behaviors**
- **QoS PHB, DSCP, H-QOS**
- **Logical Separation: MPLS/SRv6, MP-BGP, VRFs, L3/L2 VPNs, Flex-Algo, logical routers, VLANs**
- **Path selection (SR-TE) based on policy (min-latency vs BW) with On-Demand-NextHop/AS/Coloring**
- **SR-PM for Service Assurance**

- **Cloud Native NFs**
- **Per-NF capabilities**
- **Shared/Dedicated combos**
- **CUPS, NSSAI (ST/SD)**
- **APNs**
- **K8s/containers/micro-services**
- **NFV/VNFs**
- **ETSi MANO/Openstack**
Review: Cisco’s Toolset for Transport level slicing

- QoS and H-QoS: Core and edge
- Forwarding Planes: Shortest Path / SR policies (SRv6 / SR-TE / Flex-algo / Circuit-Style (future))
- SR underlay performance management tools (SR-PM)
- PCE/COE

Creating and managing the forwarding plane (underlay)

Combining these offer different levels of transport slice separation

But how do we simplify this for the end user?!

- Virtual Private networks: L2 / L3 VPNs
- ODN and Automated traffic Steering (AS)
- VPN performance management tools (Y1731)

Endpoint selection, Slice isolation and mapping to slice forwarding planes. (overlay)

Overlay to Underlay ratio can be N:1, multiple slices can use the same underlay (SR-TE)
Intent-based interface principles

1. Writing your intent is enough
2. The system strives to execute on the intent
3. Intents are idempotent – multiple requests with the same intent has no additional effect
4. You can always write intent regardless of current state

Intent is configuration done right!

Declarative Programming Model
Kubernetes is one example, so is Cisco Crosswork Network Controller (CNC)
Models describe intent

- A service model describes all the things you can do with the service
- Models allow for
  - A common abstraction of intent
  - A standard way of exchanging data (w/protocol)
  - Automated rendering of APIs and automatic data validation
  - Abstract reasoning and discussion of problems and services
- Models reduce the integration complexity—Define the contract
- **YANG** Modeling language is becoming the clear leader here
Cisco Slicing Following IETF Slicing Drafts

- IETF TEAS working group is defining Transport/Network Slices: Framework, Use Cases, Models...
  - [draft-ietf-teas-ietf-network-slices-25](#)
  - [draft-ietf-teas-ietf-network-slice-nbi-yang-08](#)
- Cisco is actively contributing to those drafts
- CNC will implement the Slice NBI Service Yang models and follow IETF guidelines in general
A Service Level Indicator (SLI) is a quantifiable measure of an aspect of the performance of a network. For example, it may be a measure of throughput in bits per second, or it may be a measure of latency in milliseconds.

A Service Level Objective (SLO) is a target value or range for the measurements returned by observation of an SLI. For example, an SLO may be expressed as "SLI <= target", or "lower bound <= SLI <= upper bound".

A Service Level Expectation (SLE) is an expression of an unmeasurable service-related request that a customer makes of the provider. An SLE is distinct from an SLO because the customer may have little or no way of determining whether the SLE is being met, but they still contract with the provider for a service that meets the expectation.

See: draft-ietf-teas-ietf-network-slices
draft-ietf-teas-ietf-network-slice-nbi-yang
Abstracting the Service Intent

### Before

**Imperative, Not intent based, Not tied together**

**Northbound Workflow/OSS/BSS**
- Create SR-TE policy
  - Head-ends
  - BGP colors
  - Path preferences
  - PCE based
  - Constraints
  - Etc.
- Create QoS policy
  - Policy-maps
  - Class-maps
  - QoS groups
  - Etc.
- Create route policy
  - Color specific prefixes
- Create VPN-profile
- Create L3VPN
  - VPN-id
  - Route-targets
  - ASNs
  - Endpoints
  - PE-CE addressing
  - Create SA probes/external SLA tools

**Traditional Transport Orchestrators**

### After

**Declarative, Intent based, Service oriented**

**Lite-Weight Workflow/OSS/BSS**
- Connect endpoints X, Y and Z in an any-to-any topology with:
  - SLO of <10ms delay
  - <.00009% Loss
  - SLE to use only high speed (>100G) links with encryption

**Slice YANG Model**

**Next-Gen Transport Service Orchestrator**
- Transport Device Provisioning
- SLA

**SLA Engine**

**Telemetry collection**
What is a Transport Slice Instance?

The four legs of the table that make up a Transport Slice Service Instance

![Transport Slice Instance](image)

**Important:** A Transport Slice Instance (or Service) is the combination of all these components.

Scale goals:
- Slice “types” defined in catalog = ~10-20?
- Slice “instances” (differentiated by VPNs/endpoints) = ~1000s
Sample Transport Slice “Intent” Types based on SLE/SLO

<table>
<thead>
<tr>
<th>Slice Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Encrypted Link Services</td>
<td>Only transit encrypted links</td>
</tr>
<tr>
<td>Disjoint Path Services</td>
<td>Network has multiple forwarding planes with no common nodes/links</td>
</tr>
<tr>
<td>High speed links only</td>
<td>Only transit high speed links (say &gt;=100Gbs, typically for elephant flows)</td>
</tr>
<tr>
<td>Lowest Latency</td>
<td>Always take lowest latency path (no SLO given)</td>
</tr>
<tr>
<td>SLO based</td>
<td>Specific thresholds not to exceed (latency, jitter, reliability..)</td>
</tr>
<tr>
<td>Regional Avoidance</td>
<td>Do not use nodes/link in specific regions</td>
</tr>
<tr>
<td>Trusted Nodes</td>
<td>Only use trusted nodes (SW verified and not in common carrier space)</td>
</tr>
<tr>
<td>L4-L7 Services</td>
<td>Perform “in-line” L4-L7 service on traffic (typically security services)</td>
</tr>
<tr>
<td>Reliable Links</td>
<td>Only transit links that have optical protection and L1 diversity</td>
</tr>
<tr>
<td>“Circuit-Style” Services</td>
<td>Provide L1 “circuit” like connectivity</td>
</tr>
<tr>
<td>Gaming Services</td>
<td>Network is optimized for network gamers (latency, BW)</td>
</tr>
<tr>
<td>Connected Car</td>
<td>Network is optimized for network connected cars (latency, proximity)</td>
</tr>
<tr>
<td>Cloud Provider XYZ</td>
<td>Connect me to secure “walled-garden” for cloud provider (AWS, Azure..) with SLA</td>
</tr>
</tbody>
</table>
Based Path Forwarding is the key to Service Intent

- Low latency / AR
- Wholesale backhaul
- Encrypt / MACSec
- FWA / Video
- Disjoint
Refresh: Path Computation ("SDN for Adults")

CNC uses SR-PCE + Optimization engine components to do path calc

1. User requests Slice (with SLA (e.g., bandwidth, latency)) from CNC UI or API
2. CNC/NSO configures Service & SR-TE policy at headend
3. Headend requests path from SR-PCE via PCEP
4. If request involves bandwidth, SR-PCE gets path from Optimization Engine (CoE). CoE has a BW utilization model built by polling interface counters using SNMP via Crosswork Data GW (CDG)
5. SR-PCE sends path to headend via PCEP
6. If request involves bandwidth and the path needs to change, Optimization Engine pushes new path to SR-PCE
7. SR-PCE updates headend via PCEP for path changes

<table>
<thead>
<tr>
<th>Objective</th>
<th>SR Policy Optimization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constraints</td>
<td>Latency/IGP/TE Metric Minimization</td>
</tr>
<tr>
<td>Affinities, Disjoint Paths, Bandwidth</td>
<td></td>
</tr>
</tbody>
</table>
IP prefix BGP coloring for Transport Slice Path Selection:  
*Intent-based SRTE with On-Demand Next Hop (ODN) & Automated Steering (AS)*

**Solution**

Edge router **automatically computes** or **requests** SR PCE a path to the remote service endpoint.

The path can either be for simple best effort reachability or for reachability with **SLA contract**.

**Benefits**

- **Intent-based**  
  SLA-aware BGP service  
  Decoupled service and transport provisioning

- **Scalability**  
  No a-priori full-mesh of connectivity

---

**Example:**

Head End executes color PURPLE ODN template  
Example: request min-delay path computation to SR-PCE  

BGP update: T/t via Node 4 with **color PURPLE**  

**Intent:**  
Min-delay metric transport path  

T/t  

**D:** Measured link delay (default 1)  

**Destinations**  

Head End instantiates an on-demand SR policy towards BGP NH for required color (intent)
Disjoint Paths Intent

Applicability Examples

- Financial Transactions
- Sensitive Data
- Transport Redundancy
- Different Fiber Conduits

Solution

Compute path disjointness to fulfill defined constraints

- Head-end router or network-wide disjointness
- Live-Live / Primary-Backup transport
- Disjoint paths do not share any (or limited) network resources

Benefits

- Simplicity and Automation
  - Colored plane topology - Use of SR Anycast-SID
  - Non-dual plane topology - Use SR-PCE to compute disjoint path or use SR Flex Algorithm
- Flexibility
  - Link, Node, SRLG, Remote-SRLG, Node+SRLG
  - Disjointness

Interesting? Segment Routing Traffic Engineering (SRTE) on segment-routing.net

Unique Functionality

Compute node-disjoint transport paths between Node1-to-Node4 and Node5-to-Node8

Default link metric: 10

1. Node1 first requests path to Node4
   - SR-PCE computes

2. Node5 later requests path to Node8
   - SR-PCE reoptimizes first LSP in order to find a node-disjoint solution
Real-Time Low Delay Intent

Applicability Examples

<table>
<thead>
<tr>
<th>Extreme Real-time Communications</th>
<th>Tactile Internet</th>
</tr>
</thead>
<tbody>
<tr>
<td>Voice Communications</td>
<td>Fixed / mobile</td>
</tr>
</tbody>
</table>

Solution

Compute Low Latency path based on measured link delay/jitter/drop with Performance Monitoring (PM)

Benefits

Simplicity and Automation
Performance Management to measure real time link delay, jitter and loss
detect optical path reroute, by measuring delay/jitter/loss variations in real-time

Troubleshooting tool
Meet, Maintain and Monitor SLAs at all times

MPLS Performance Monitoring (PM)

Exhaustive Telemetry every 30s
ISIS update upon significant change SRTE re-optimization
Probe every 3s

Find the best delay-optimized path to Node 4

Interested? Delay Measurement on segment-routing.net
Shortest IGP path with cumulative Delay Bound

Applicability Examples

<table>
<thead>
<tr>
<th>Extreme</th>
<th>Real-time Communications</th>
</tr>
</thead>
<tbody>
<tr>
<td>Voice</td>
<td>Communications</td>
</tr>
<tr>
<td>Tactile Internet</td>
<td>Fixed / mobile</td>
</tr>
</tbody>
</table>

Solution

Generic “cumulative metric-bounded shortest path” solution for SR-TE

Benefits

Better utilization of network capacity
- If multiple paths within the delay bound have been found, SRTE will use the one with the lowest IGP metric ensuring that high bandwidth links will be used closer to the core and low capacity links closer to the edge

Meet, Maintain and Monitor SLAs at all times
Crosswork Network Controller (CNC) Simplify operational lifecycle

Challenges
1. Time-consuming service provisioning
2. BW swings, over capacity
3. Congestions and service impacting conditions
4. Siloed, ineffective tools (high OpEx)

Outcomes
1. Intent-based automated provisioning
2. Dynamic BW management
3. Closed loop automation
4. Turnkey solution – Across lifecycle

Cisco Crosswork Network Controller

Programmable | Distributed | Dynamic

Access
Fronthaul
Midhaul/backhaul
Midhaul/backhaul
Core

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public
Crosswork Network Controller (CNC)

Integrated Service and Device Management

Transport Slicing **NEW** in CNC 6.0!

- **OSS/BSS and/or Hierarchical Controller**
  - REST/RESTCONF Model-based

---

**Crosswork Network Controller**

- **Service and transport provisioning**
  - YANG service models
  - 170+ multivendor devices supported in addition to Netconf
  - Transport Slice, L2VPN, L3VPN, SR-TE, SRv6, RSVP-TE, Tree-SID and more

- **Visualization**
  - Programmable to support new services and devices
  - Service, network, and device views
  - End-to-end dynamic path and policy view
  - Traffic engineering dashboard
  - Flexible algorithm

- **Service assurance**
  - Embedded probes
  - External probes
  - Transport slices and slice catalog
  - Inventory, fault, performance

- **Element Management**
  - Dynamic traffic engineering

---

**Network Architecture**

- **Access**
- **Pre-Aggregation**
- **Aggregation**
- **Core**

- **Fronthaul**
- **Midhaul/Backhaul**

**Technology**

- 3G
- 4G
- 5G
- DSL/PON
- SR-TE
- SVPN

**Closed-loop automation**

---

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public
Transport Slicing Automation Made Easy with CI

Transport Slice Orchestration
- Intent-based slice definition
- Abstracted Slice model based on IETF model
- Slice Template Catalog
- Simplified provisioning User Interface
- SR/SRv6 Optimisation

Transport Slice Visualisation
- Overlay maps
- Slice details: Type, Template, VPNs, Transport
- Navigation between Slice components
- Infrastructure visibility
- Multi-layer view and analysis of the slice

Transport Slice SLA Monitoring
- Proactive SLA monitoring per Slice
- Leveraging Cisco (SR-PM) or Accedian instrumentation
- SLA breach notification
- SLA reporting

Transport Slice Health Monitoring
- Service Centric approach to health monitoring
- Dynamically tie Slice Intent to infrastructure telemetry
- Correlate active probing with infrastructure monitoring
Transport Slice Template Catalog

- User-defined slice templates
- Template includes:
  - Template Name
  - Description
  - Input/Output QOS
  - Forwarding plane policy
  - Allow for BWoD customizations
  - Service Assurance parameters
- Templates can be created through GUI or API
- Path forwarding policies can be shared across multiple Slice Instances for scale.
Simplified Transport Slice Creation - 4 Steps!

1. **Specify Slice ID, Service Type, Customer, Description, NSSAI (optional)**...
2. **Specify Slice Intent from Template Catalog**
3. **Specify Connectivity Type, Isolation, Bandwidth...**
4. **Define Service Demarcation Points (endpoints)**
Transport Slicing in Crosswork Network Controller Visualization

New menu for Slice Instances and Template Catalog

Slice list including Intent and provisioning state
Navigate the Slice components: VPN, Transport

From the VPN list, display VPN Service details including Service Health, Alarms and Assurance Graph data.

From the Transport list, display SR TE details including SR-PM data if SR-PM is enabled.

Display a slice on the map.

Drill-down to VPN and/or Transport details.
Drill Down into Slice Service Assurance and Performance
CNC Slice Provisioning Options—Pick what works best for you

NSO CLI/load merge

```
admin@ncs-6.1(config)# load merge
a_L3_A2A_dedicated_URLLC_NoFA_PM.cli
Loading.
1.69 KiB parsed in 0.34 sec (4.87 KiB/sec)
admin@ncs-6.1(config)# commit
```

CNC Custom Slice

```
admin@ncs-6.1# show running-configure network-slice-services slice-service
network-slice-services slice-service slice-L2p2p-Foo
service-tags tag-type service-tag-customer
value [ FOO ]
! service-tags tag-type service-tag-service
value [ L2 ]
!
```

CNC/NSO API

```
slice-match-criteria match-criteria target-connection-group-id group:
```

Notes:
- CNC UI/API acts as “proxy” to NSO
- CNC will learn (and visualize) any NSO provisioned slice services via NSO notifications.
Changing Slice Intent "on-the-fly" for an active Slice Instance

It's Cloud-like!

Before Change:

RP/0/RSP0/CPU0:NY-Regional-PE# **ping vrf slice-coke**
Sending 5, 100-byte ICMP Echos to 200.200.200.200, timeout is 2 seconds: !!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 22/22/23 ms

High BW- High Latency Path:

After Change:

RP/0/RSP0/CPU0:NY-Regional-PE# **ping vrf slice-coke**
Sending 5, 100-byte ICMP Echos to 200.200.200.200, timeout is 2 seconds: !!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 2/2/6 ms

Low BW- Low Latency Path:

admin@ncs(config)# network-slice-services slice-service slice-coke slo-sle-template ?
Possible completions:
[eMBB]  URLLC_PM  Use Low Latency links
eMBB   Use High BW links
admin@ncs(config)# network-slice-services slice-service slice-coke slo-sle-template URLLC_PM
admin@ncs(config)# commit
Transport Slicing is about user “Intent” (or outcome) and using a simplified service request.

Declarative and Intent based service models are the future. Key cloud principle. The network can become an “on-demand” utility with ultimate flexibility.

Cisco is taking a lead industry position to define the key dimensions of Transport Slicing (IETF contributor) along with delivered industry leading underpinnings (SR, fancy QoS, PM, PCE/COE).

Cisco has several key differentiators:
- Rich MPLS-SR and SRv6 features and roadmap for both function and scale
- A declarative, powerful (FASTMAP), network commit based network orchestrator (NSO) underpinning CNC
- Shared Slice concept with Single-Sided provisioning to maintain bi-directional slice intent.
- Slice automatic connectivity construct matrix creation
- Service Assurance integration with SR-PM, Y1731 and CNC Automated Assurance

Cisco’s Transport Slicing is being delivered in CNC6.0! Come celebrate today at 3:00 during break!
The bridge to possible