



# NSP

## Network Services Platform

Release 23.11

### Transport Slice Controller Guide

---

3HE-18994-AAAC-TQZZA  
Issue 1  
December 2023

---

## Legal notice

Nokia is committed to diversity and inclusion. We are continuously reviewing our customer documentation and consulting with standards bodies to ensure that terminology is inclusive and aligned with the industry. Our future customer documentation will be updated accordingly.

---

This document includes Nokia proprietary and confidential information, which may not be distributed or disclosed to any third parties without the prior written consent of Nokia.

This document is intended for use by Nokia's customers ("You"/"Your") in connection with a product purchased or licensed from any company within Nokia Group of Companies. Use this document as agreed. You agree to notify Nokia of any errors you may find in this document; however, should you elect to use this document for any purpose(s) for which it is not intended, You understand and warrant that any determinations You may make or actions You may take will be based upon Your independent judgment and analysis of the content of this document.

Nokia reserves the right to make changes to this document without notice. At all times, the controlling version is the one available on Nokia's site.

No part of this document may be modified.

NO WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY OF AVAILABILITY, ACCURACY, RELIABILITY, TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE, IS MADE IN RELATION TO THE CONTENT OF THIS DOCUMENT. IN NO EVENT WILL NOKIA BE LIABLE FOR ANY DAMAGES, INCLUDING BUT NOT LIMITED TO SPECIAL, DIRECT, INDIRECT, INCIDENTAL OR CONSEQUENTIAL OR ANY LOSSES, SUCH AS BUT NOT LIMITED TO LOSS OF PROFIT, REVENUE, BUSINESS INTERRUPTION, BUSINESS OPPORTUNITY OR DATA THAT MAY ARISE FROM THE USE OF THIS DOCUMENT OR THE INFORMATION IN IT, EVEN IN THE CASE OF ERRORS IN OR OMISSIONS FROM THIS DOCUMENT OR ITS CONTENT.

Copyright and trademark: Nokia is a registered trademark of Nokia Corporation. Other product names mentioned in this document may be trademarks of their respective owners.

© 2023 Nokia.

---

# Contents

<b>About this document.....</b>	<b>5</b>
<b>1 Transport Slice Controller .....</b>	<b>7</b>
1.1 Overview .....	7
1.2 What is Transport Slice Controller?.....	13
1.3 What TSC functions does NSP provide? .....	14
1.4 How do TSC interfaces work?.....	14
1.5 What are the programmable functions of the TSC? .....	16
<b>2 TSC setup and operation.....</b>	<b>19</b>
2.1 How do I setup TSC? .....	19
2.2 How does the TSC API flow work?.....	19
<b>3 TSC dashboard.....</b>	<b>21</b>
3.1 What is the Transport Slice Controller dashboard?.....	21
3.2 How do I open the Transport Slice Controller? .....	24
3.3 How do I view the health of transport slices in the network? .....	24
3.4 How do I view details of the transport slice realization using L0/L1/L2/L3 services and tunnels/ paths? .....	26
3.5 How do I view reporting data on transport slices? .....	27



---

# About this document

## Purpose

The *Transport Slice Controller Guide* introduces the Transport Slice Controller to operators and administrators.

## Scope

This document covers the central concepts of network slicing and the functions and interfaces of the Transport Slice Controller.

## Document support

Customer documentation and product support URLs:

- [Documentation Center](#)
- [Technical support](#)

## How to comment

Please send your feedback to [Documentation Feedback](#).



---

# 1 Transport Slice Controller

## 1.1 Overview

This chapter describes the concept of end-to-end network slicing and how transport slices relate to NSP functionality as Transport Slice Controller (TSC). This chapter covers the following topics:

- [1.1.1 “What is end-to-end network slicing?” \(p. 7\)](#)
- [1.1.2 “How are network slices created?” \(p. 8\)](#)
- [1.1.3 “What is a transport slice?” \(p. 10\)](#)



**Note:** Please contact your NSP PLM representative to assess your network and use-case if you intend to deploy TSC.

### 1.1.1 What is end-to-end network slicing?

End-to-end network slicing is a technology for concurrent delivery of differentiated services and a key component of moving wireless network use-cases toward a service-driven evolution that supports meeting SLAs deterministically across end-to-end network resources. Network slices are independent, logical self-contained networks representing common physical or virtual network infrastructure that extends from end devices to application servers and includes all intermediate functions and domains.

Network slicing is beneficial to various applications, including the following:

- wireless network slicing
- wholesale business VPNs
- network sharing between operators
- data center connectivity

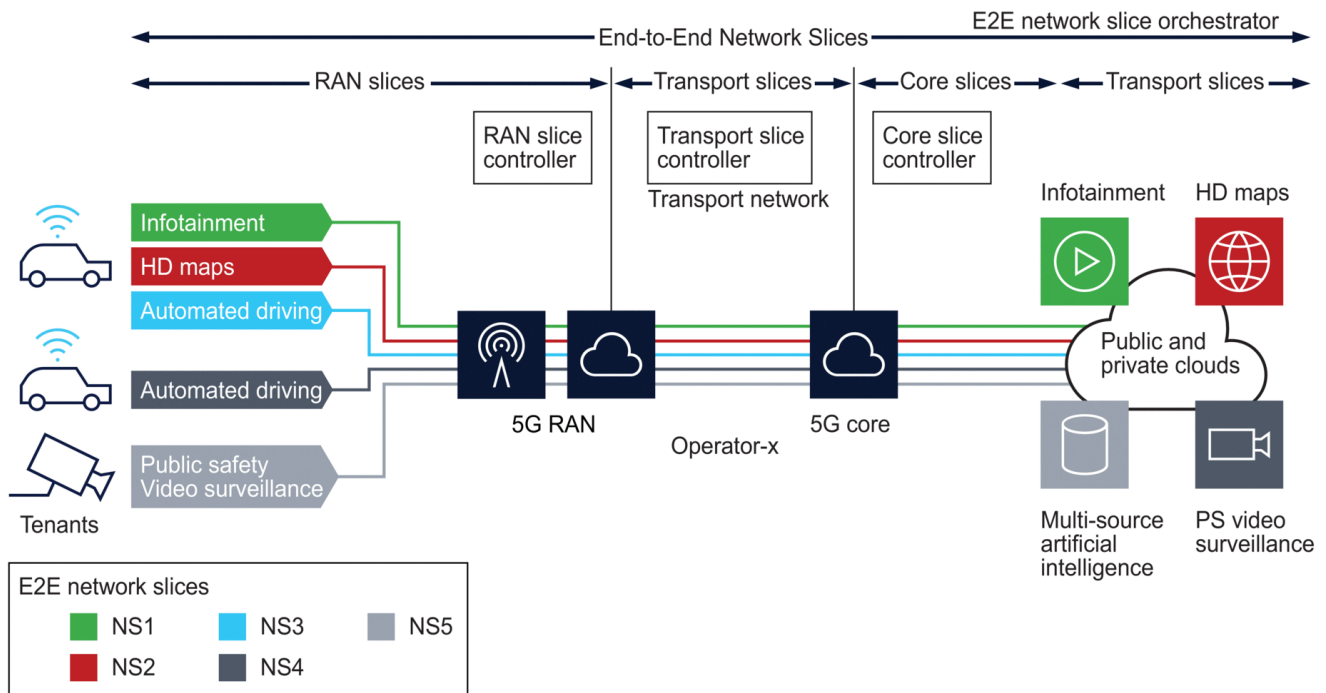
Network slices can include virtual and physical network functions, cloud infrastructure, transport/connectivity, augmented services (for example, network analytics and security services), as well as application functions. Network slices are orchestrated to form service-specific logical networks running on the same physical network that meet certain Service Level Objective (SLO) attributes such as data speed, capacity, latency, reliability, availability, coverage, and security.

To better demonstrate the concept of network slicing, [Figure 1-1, “Typical end-to-end network slice from Operator-X perspective” \(p. 8\)](#) shows a typical network from Mobile Network Operator (MNO) Operator-X that supports network slicing. The operator has three customers (a.k.a. tenants): Public Safety, Enterprise-Y, and Enterprise-Z. Each of these customers asks Operator-X to create one or more logical independent end-to-end networks within its common or shared network infrastructure with SLOs. Each of these end-to-end logical networks is called a “network slice”. In this network, Operator-X has created five network slices (NS1 to NS5), each with specific SLO marked in different colors (SLO red, green, etc.).

Network slicing is required since services and devices have their own specific SLO requirements, many of which vary diversely depending on the application. As shown in [Figure 1-1, “Typical end-to-end network slice from Operator-X perspective” \(p. 8\)](#), each end-to-end network slice consists of

one RAN slice, one Core slice and one or more IETF Network Slices. Note that network slices are inherently an end-to-end concept and used to connect user equipment to tenant-specific applications/servers.

Figure 1-1 Typical end-to-end network slice from Operator-X perspective



37440

In this example network, customer Enterprise-Y has three end-to-end network slices for services “Infotainment”, “HD maps” and “Autonomous driving”, each with its own SLO. Similarly, customers Enterprise-Z and Public Safety each have one end-to-end network slice (NS4 and NS5) for services “Autonomous driving” and “Video Surveillance”, respectively. For instance, NS1 is created by Operator-X for customer Enterprise-Y for service “Infotainment” where the SLO is bandwidth (SLO green) whereas the NS3 is created for the same customer but for “Autonomous driving”, where the SLO is the combination of latency and reliability (SLO blue).

### 1.1.2 How are network slices created?

Continuing the example shown in [Figure 1-1, “Typical end-to-end network slice from Operator-X perspective” \(p. 8\)](#), the following is an example of how end-to-end network slice NS5 is created in the operator’s network:

1. The “E2E Network slice Orchestrator” receives an abstracted Service Catalog instantiation request from the customer portal.
2. Using Network Slice Blueprints, the “E2E Network slice Orchestrator” creates the “Network



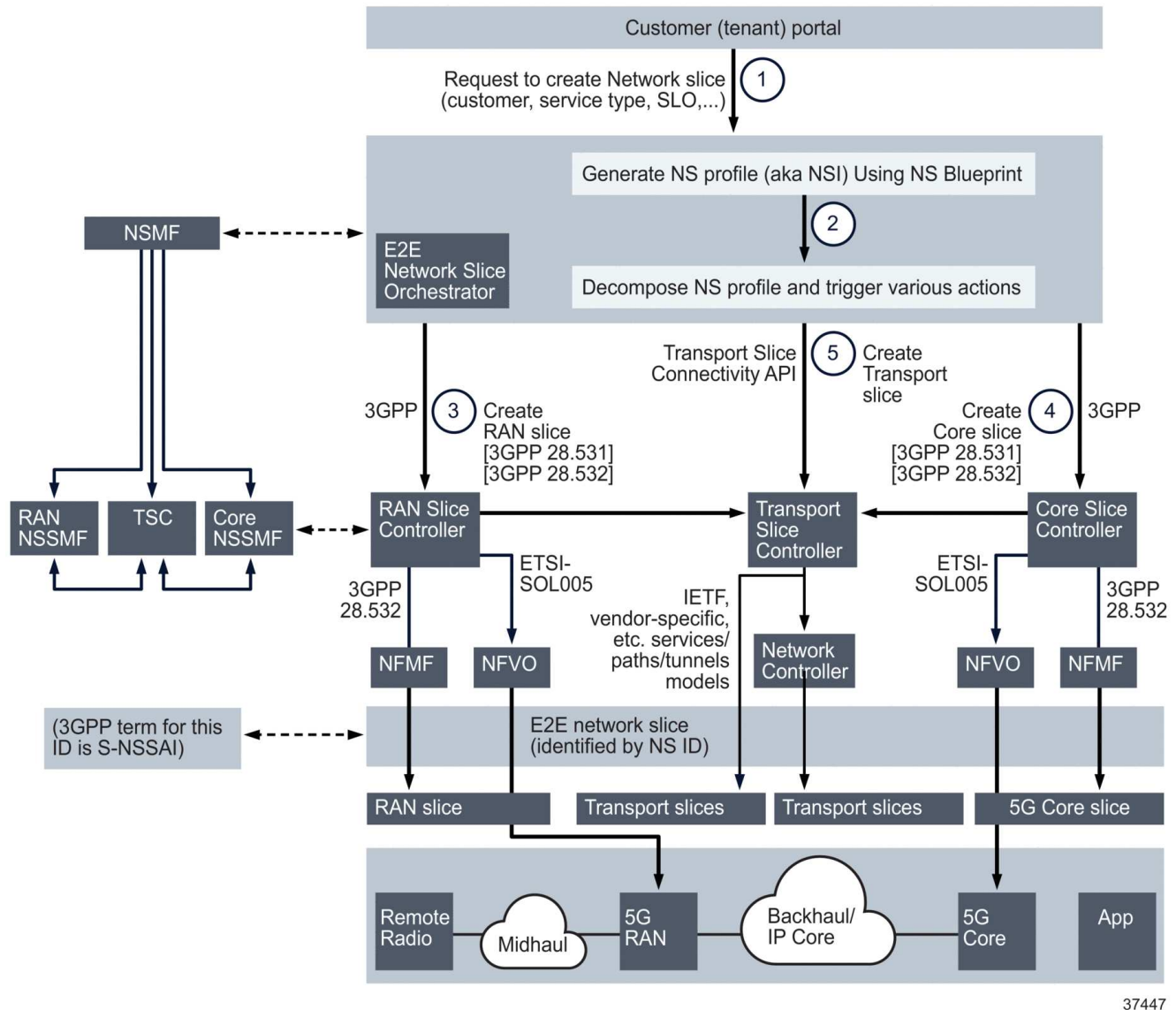
---

Slice Profile (NS Profile)” for this request and starts decomposing the profile request into smaller chunks of network slice subnet sent to various slice domain controllers.

3. The RAN slice controller (i.e. RAN NSSMF) will receive a request to create the RAN Slice. If the RAN is virtualized, it will in turn use the NFVO to initiate VNF creation and then program the RAN slice.
4. Similar to step 3, the Core slice controller (i.e. Core NSSMF) will receive a request to create the Core Slice. If Core is virtualized, it will in turn use the NFVO to initiate VNF creation and then program the Core slice. The Core Slice Controller uses network service descriptors to create new EPC/5GC VNF Components, DC networks, or additional GiLAN SFC VNF Service Chains (e.g. vFW, vNAT, vLoad Balancer).
5. The Transport Slice Controller (i.e. TSC) will act on instructions passed on in the transport slice connectivity API to instantiate various connections between RAN and Core networks, between RAN CU-DU, or RH-DU and Core to applications.

The following figure provides an example for the creation of end-to-end network slice NS5.

Figure 1-2 Network slice creation



Note that Interfaces (3) and (4) are defined in various 3GPP technical specifications such as 28.530, 28.531, 28.540 and 28.541. However, interface (5) currently has no standard. Nokia is working with the IETF and BBF towards standardizing this interface.

### 1.1.3 What is a transport slice?

Transport slices provide technology-agnostic connectivity between various network physical/virtual functions and applications. The example shown in [Figure 1-1, "Typical end-to-end network slice"](#)

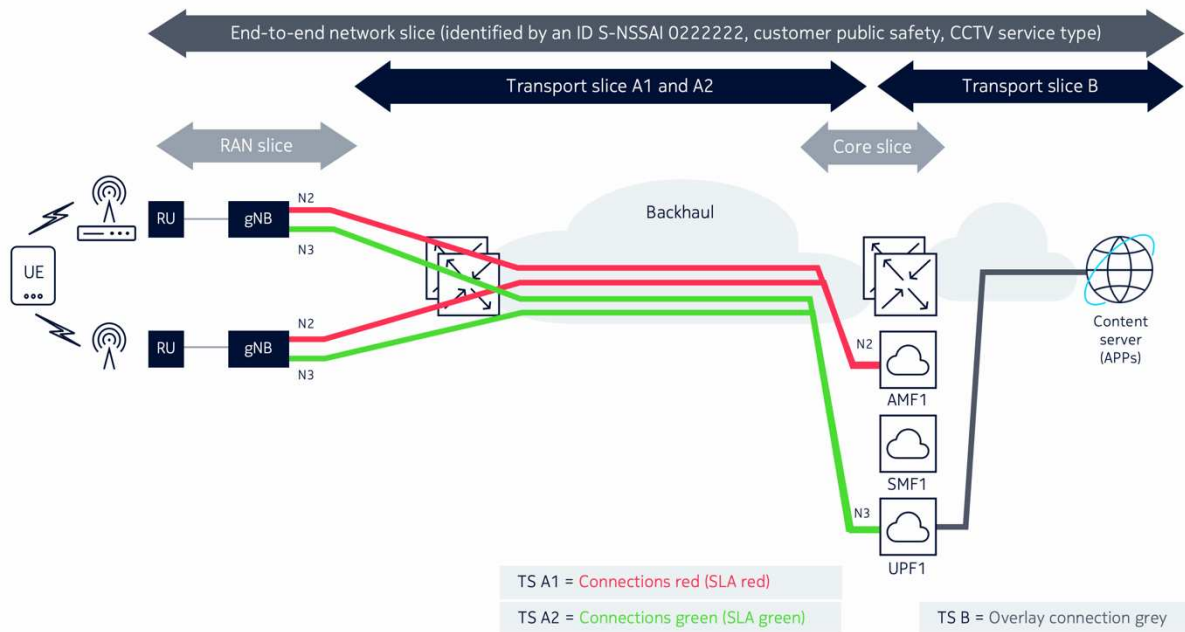
from Operator-X perspective” (p. 11) displays two transport slices; one transport slice is the connection between the RAN and core slices, and the other is the connection between the core slice and application servers. Each transport slice has its own SLA. IETF network slices have deterministic Service-Level Objectives (SLO) in order to achieve the end-to-end network slice SLA.

The following figures illustrate the concept of transport slices in various RAN deployments (Distributed-RAN or D-RAN, Centralized RAN and Cloud RAN or C-RAN) for an end-to-end network slice identified by the following attributes:

- E2E network slice ID S-NSSAI 0222222
- customer public safety
- service type eMBB CCTV
- SLA 10 (MBPS)

A D-RAN deployment is illustrated in Figure 1-3, “Transport slices in Distributed-RAN” (p. 10), where the RAN elements are responsible for all aspects of the radio access functions including signal processing, radio interfacing, and scheduling. This type of deployment is the functional equivalent of the standard 4G network, where eNodeB network elements are responsible for all 4G RAN functions. In this case, the transport slices A1 and A2 are between gNB network elements and core network functions for SLAs Green and Red. The transport slice B (with SLA Grey) is between the core slice and application servers.

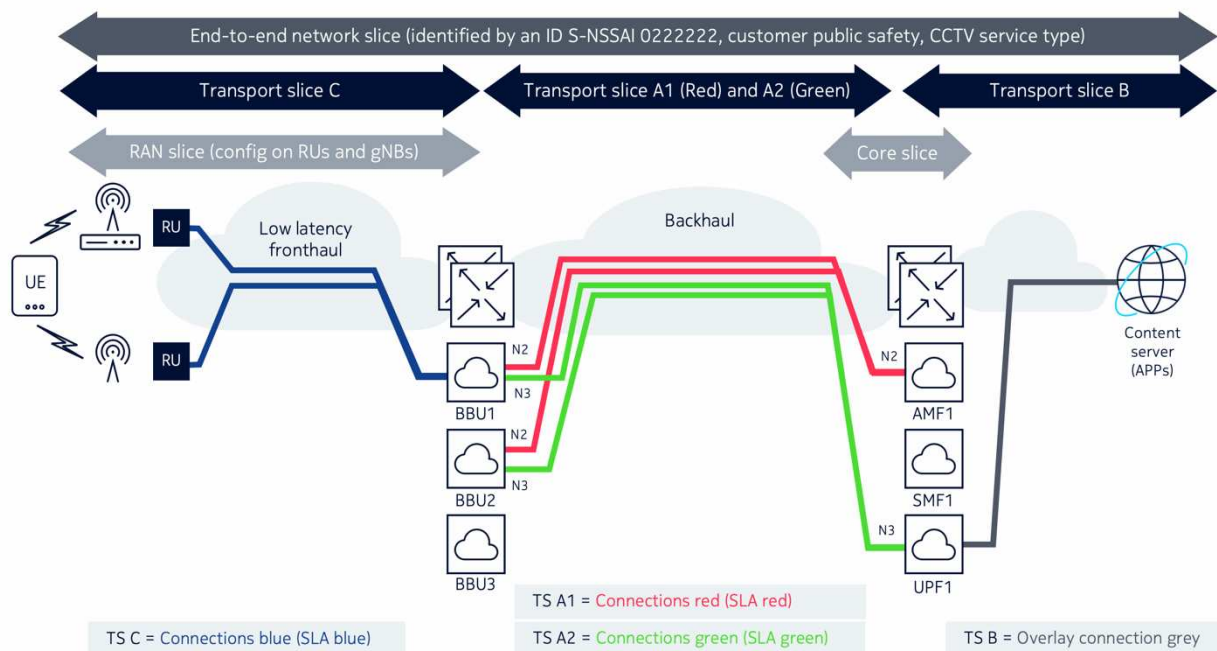
Figure 1-3 Transport slices in Distributed-RAN



As shown in Figure 1-4, “Transport slices in Centralized-RAN” (p. 12), in a Centralized-RAN deployment, the signal processing units (BBU) of the RAN nodes are centralized to simplify network

management and enhance computation capability. Resulting from this, a new network type called a “fronthaul network” is introduced to provide the transport connectivity between the Radio Unit (RU) and the BBU. The following figure illustrates the transport slices for this deployment for the same end-to-end network slice.

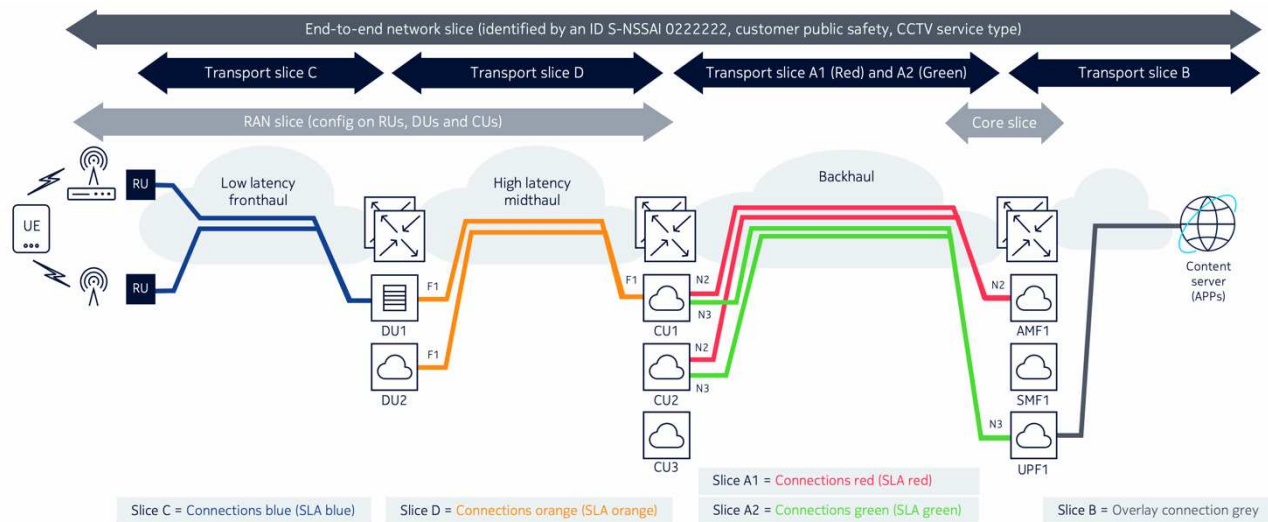
Figure 1-4 Transport slices in Centralized-RAN



Comparing Figure 1-3, “Transport slices in Distributed-RAN” (p. 11) and Figure 1-4, “Transport slices in Centralized-RAN” (p. 12) shows that in addition to transport slices A1, A2, and B, a new transport slice C is required to provide connectivity between the RU network elements and BBU nodes. Transport slice C contains “Blue” connections with strict latency SLA requirements.

In a Cloud-RAN (CRAN) deployment, the BBU network elements can be divided further into Distributed Unit (DU) and Central Unit (CU). In this scenario, a new network called a “midhaul network” is introduced to provide transport connectivity between DUs and CUs. Figure 1-5, “Transport slices in Cloud-RAN” (p. 13) shows a new slice D with SLA Orange is required to interconnect the DU and CU functions in the midhaul transport network.

Figure 1-5 Transport slices in Cloud-RAN



## 1.2 What is Transport Slice Controller?

The Transport Slice Controller (TSC) is a functional block that provides realization, monitoring, and optimization of transport slices in the network. It has a layered architecture, with a northbound interface toward a higher-level system (i.e. end-to-end network slice orchestrator) and southbound interface toward the transport network controllers. The NBI is technology-agnostic while the SBI is technology-specific. TSC provides automation of the transport slices during the realization and also allows pro-active monitoring of transport slices and performs closed-loop optimization in the case of SLA violations.

The TSC provides the following functions:

- automating the creation/modification/deletion of transport slices
- monitoring, analytics and reporting on transport slices
- closed-loop optimization on transport slices

The NSP as TSC provides a programmable function for automating, monitoring, and optimizing 4G/5G transport slices in a managed network. The TSC uses the NSP intent-based framework to provide transport slice realization and SLA assurance across multiple technology types including IP/MPLS, optical, PON, and microwave.

- Transport slice front-end - a new dashboard on NSP which is transports slice-centric and provides a single pane of glass for all transport slices in the network.
- Transport slice back-end - this functionality provides a programmable framework for transport slice lifecycle management. The back-end is a multi-vendor intent-based programmable framework that allows the operator to realize transport slices for any technology in fronthaul, midhaul, and backhaul networks. It also provides the pro-active monitoring of transport slices and preform the closed-loop optimization in case of SLA violations.

---

## 1.3 What TSC functions does NSP provide?

### 1.3.1 Automating the creation, modification, and deletion of transport slices

Automation of transport slice creation, modification, and deletion is accomplished through the end-to-end Network Slice Orchestrator using NS Profiles through the IETF transport slice NBI. The NSP as TSC provides full intent-based programmability of the transport slice. The TSC functionality is itself an intent type, and the TSC can use any intent types available on NSP for realization of transport slices.

### 1.3.2 Monitoring, analytics, and reporting on transport slices

Monitoring, analytics, and reporting on transport slices is accomplished through notifications and datastore updates. Performance status monitoring is provided by three categories of statistics:

- per transport slice SLO group
- per transport slice connection
- per transport slice endpoint

These statistics can be viewed in the TSC dashboard. NSP Analytics can generate historical reports using transport slice bandwidth and latency custom reports. The following report types have been added to NSP Analytics specifically for transport slices:

- latency-based reports
- bandwidth-based reports

Telemetry data (stats, alarms, OAM results) from the network are collected by NSP. The TSC associates this data with transport slices to determine the transport slice SLA. These SLAs are then:

- displayed on the dashboard
- exposed to NSP Analytics for TSC reports

### 1.3.3 Closed-loop optimization of transport slices

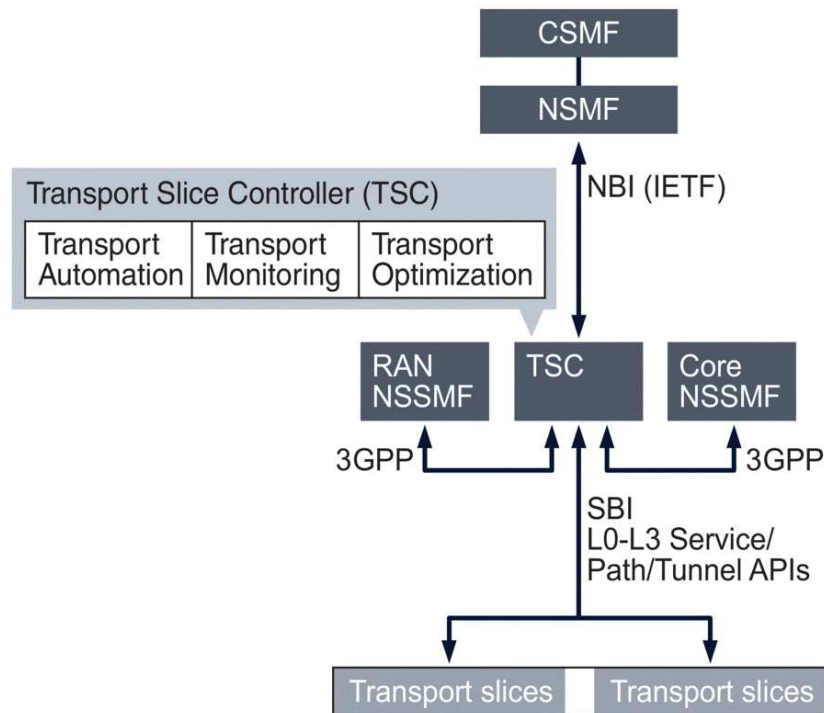
Optimization of transport slices is accomplished through the PCE, which pro-actively monitors the tunnel resources and performs the closed-loop optimization (e.g. LSP re-route). Optimization-based changes are sent to the TSC, which provides optimization reports and sends data on optimization events to the orchestrator.

## 1.4 How do TSC interfaces work?

### 1.4.1 Northbound and southbound interfaces

Figure 1-6, “TSC for automation and assurance of transport slices” (p. 15) shows how the NSP-as-TSC uses its NBI to communicate with the end-to-end network slice orchestrator and its SBI to communicate with the transport network controllers. Various 3GPP interfaces connect the TSC with the RAN and Core NSSMF.

Figure 1-6 TSC for automation and assurance of transport slices



37446

## NBI

The NBI between the TSC and the end-to-end network slice orchestrator (NSMF) is a technology agnostic RESTful interface used to send requests for creation/deletion/modification of transport slices.

## SBI

The SBI are RESTful technology-specific (IP, Optics, PON, Microwave, etc) interfaces toward third-party applications used for transport slice realization.

### 1.4.2 IETF compliance

TSC is compliant with the IETF Standard/Open model. The TSC supports the IETF NBI described by the following specification:

<https://datatracker.ietf.org/doc/draft-ietf-teas-ietf-network-slice-nbi-yang/>

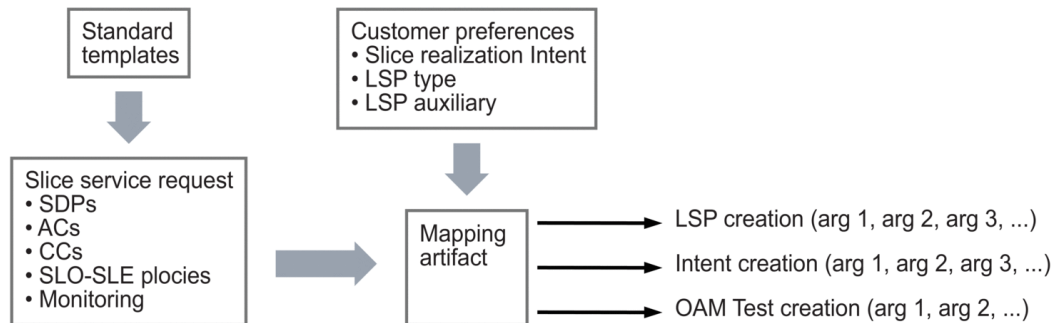


## 1.5 What are the programmable functions of the TSC?

### 1.5.1 Overview

The following figure displays a high-level overview of the user programmable components of the TSC.

Figure 1-7 High-level view of programmable portions of TSC



38933

The IETF NBI payload can be found here:

<https://www.ietf.org/archive/id/draft-ietf-teas-ietf-network-slice-nbi-yang-03.html>

See the [Network Developer Portal](#) for information about the supported elements of the IETF NBI YANG model.

### 1.5.2 Mapping artifacts

The various elements of slice realization require parameter adaptation and mapping, mostly accomplished using mapping artifacts. Mapping is performed by:

- matching ServiceTemplateName in operator-preferences.json to the name of the service intent used to realize slice services
- importing release-specific mapping artifacts, obtainable from Nokia support, as described in [2.1 "How do I setup TSC?"](#) (p. 19)

### 1.5.3 Features

This section describes specific information about key TSC features.

#### N:1 mapping of slice services to resources

The NSP can map slice related intent to a dedicated LSP, or multiple intents that have the same SLO/SLE requirements onto a shared LSP. Configuration of this feature is performed in the operator-preferences.json intent type file. An example of the relevant section is the following:

```
"Lsp": {
```



---

```
"Isolation": false,  
  
"Protection": false,  
  
"SignalingType": "sr-te",  
  
"PrimaryPathPriority": "1",  
  
"SecondaryPathPriority": "2"  
  
}
```

See [2.1.2 “Obtain and set operator preferences” \(p. 19\)](#) for more information about the operator-preferences.json intent type file.

### LSP isolation

In operator-preferences.json, operators can set the *Isolation* parameter to “true” to enable dedicated LSPs per slice service, or set it to “false” and reuse existing LSPs. The default setting is “false”.

### Path protection

The *Protection* parameter provides protection for LSP paths. When set to “true”, LSPs are deployed and even reused with primary and secondary paths. When set to “false”, LSPs have a primary path only. The default setting is “false”.

### Signaling type

The *SignalingType* parameter allows operators to choose the LSP signaling type. The options are “rsvp” and “sr-te”. The default is “sr-te”.

### Path priority

Operators can set the *PrimaryPathPriority* and *SecondaryPathPriority* levels for LSP paths using these parameters. The default for *PrimaryPathPriority* is “1” and the default for *SecondaryPathPriority* is “2”.



---

## 2 TSC setup and operation

### 2.1 How do I setup TSC?

This section describes the pre-configuration tasks required to use the TSC.

#### 2.1.1 Obtain and import the required intent bundles and mapping files

TSC requires the following intent bundles:

- NSP\_Predefined\_IETF\_Intents\_rel.zip
- NSP\_Predefined\_Service\_Intents\_rel.zip
- NSP\_Transport\_Slice\_Intent\_rel.zip

See the *NSP Network Automation Guide* for more information about obtaining and installing intent bundles.

TSC requires the following mapping files:

- NSP\_Predefined\_IETF\_Intents\_rel.zip
- NSP\_Predefined\_Service\_Intents\_rel.zip

See the *NSP System Administrator Guide* for information about importing MDM mapping files.

#### 2.1.2 Obtain and set operator preferences

Obtain the operator-preferences.json intent type file for the current release from Nokia support. This file contains configuration data that is pre-configured and imported by the NSP as a setup task for the TSC. The import operation is generally performed only once.

##### Important preferences settings

Make sure to set the following correctly:

- AS number (*autonomous-system*)
- Service type—the *ServiceType* parameter must be set to the same value as *slo-sle-template* in the IETF NBI data model.
- Service template name—the *ServiceTemplateName* specifies the service intent used to realize the slice service.
- Isolation

### 2.2 How does the TSC API flow work?

#### 2.2.1 Link API

Link API creates the binding between SDP nodes and router ports. Link API is invoked:

- initially for all existing nodes

- 
- for every new node added to the network

### 2.2.2 IETF API

The IETF NBI is invoked:

- for every new slice request
- for every slice modification

The IETF NBI is also used to provide status updates, alarms, and other information from nodes to the TSC.

### 2.2.3 API documentation

Tutorials for the Transport Slice Controller API are available on the [Network Developer Portal](#) .

---

## 3 TSC dashboard

### 3.1 What is the Transport Slice Controller dashboard?

#### 3.1.1 Functionality provided by the TSC dashboard

The Transport Slice Controller dashboard provides the following functionality for managing transport slices in the network:

- overall health summary for all managed transport slices
- mapping between transport slices and network slices
- drill-down into health views on a per-slice basis
- details on L0/L1/L2/L3 services and tunnels/paths used during the realization of transport slices
- pro-active monitoring telemetry data and reports on transport slices
- alarm correlation between slices, NEs, IP services, and LSPs
- closed-loop optimization on transport slices

See the following workflows for more information about what you can do with the TSC:

- [3.3 “How do I view the health of transport slices in the network?” \(p. 24\)](#)
- [3.4 “How do I view details of the transport slice realization using L0/L1/L2/L3 services and tunnels/paths?” \(p. 26\)](#)
- [3.5 “How do I view reporting data on transport slices?” \(p. 27\)](#)

The TSC dashboard provides two main views of transport slices:

- overall, for all slices
- detailed, per slice

[Figure 3-1, “Overall health of all slices” \(p. 22\)](#) shows the TSC dashboard for overall health of all transport slices, at-a-glance mapping information, and correlation between transport slices and network slices.

Figure 3-1 Overall health of all slices

## 5G Transport Slice dashboard

Health of all transport slices

Overall health of all transport slices in the network

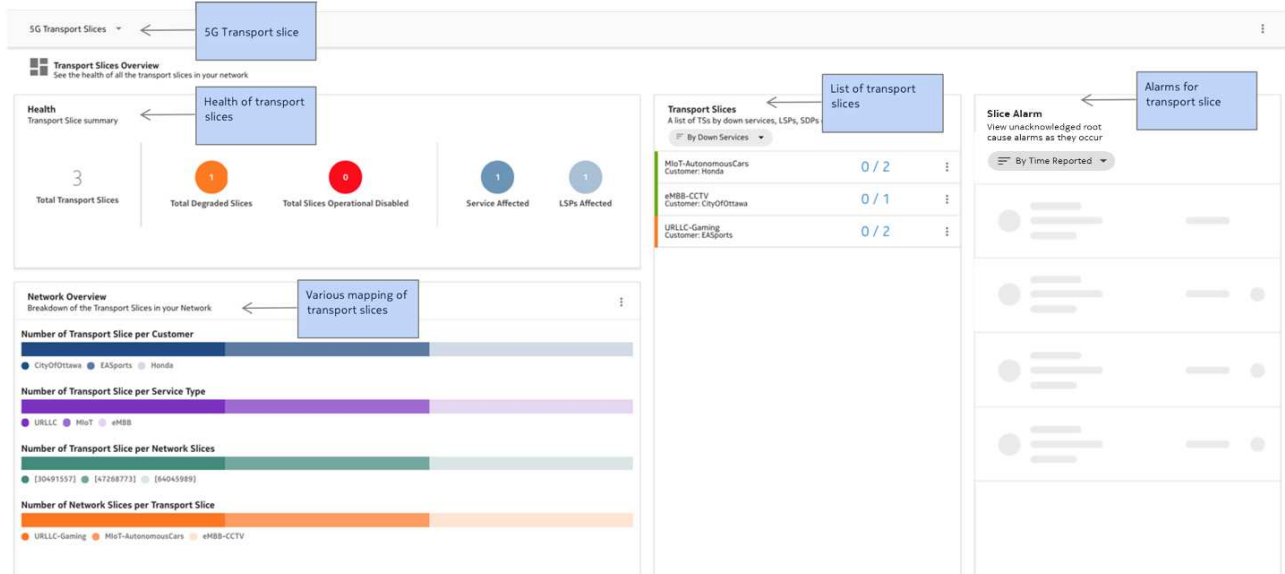


Figure 3-2, “Detailed view of a specific slice” (p. 23) shows the detailed view, available on a per-slice basis, displaying a drilled-down view of the selected slice. This view displays identifying information with SLA information and the core/RAN slices associated with the transport slice. Cross-launch actions to other NSP functions are available in this view.

The TSC detailed dashboard view displays the following information for a single transport slice:

- transport slice details
- network slice which uses the transport slice
- network slice information such as customer, service type (such as URLLC, CCTV, Infotainment), network slice ID (S-NSSAI)
- overall health of this transport slice (OK, Down)
- number of L0-L3 services and tunnels used/created/modified to realize the transport slice
- details of the Requested SLA and Current SLA (such as min/average/max SLA)
- details of the L0-L3 services used during the transport slice realization
- details of all tunnels/LSPs used during the transport slice realization
- how this transport slice is connected to RAN and Core slices

Figure 3-2 Detailed view of a specific slice

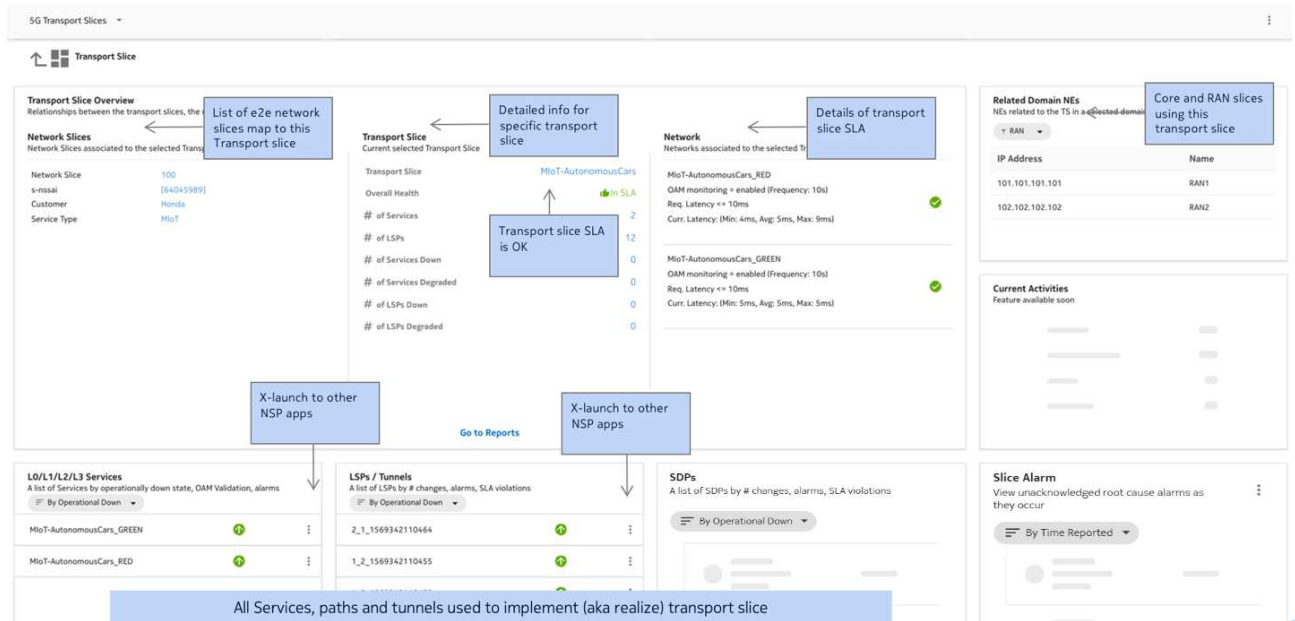
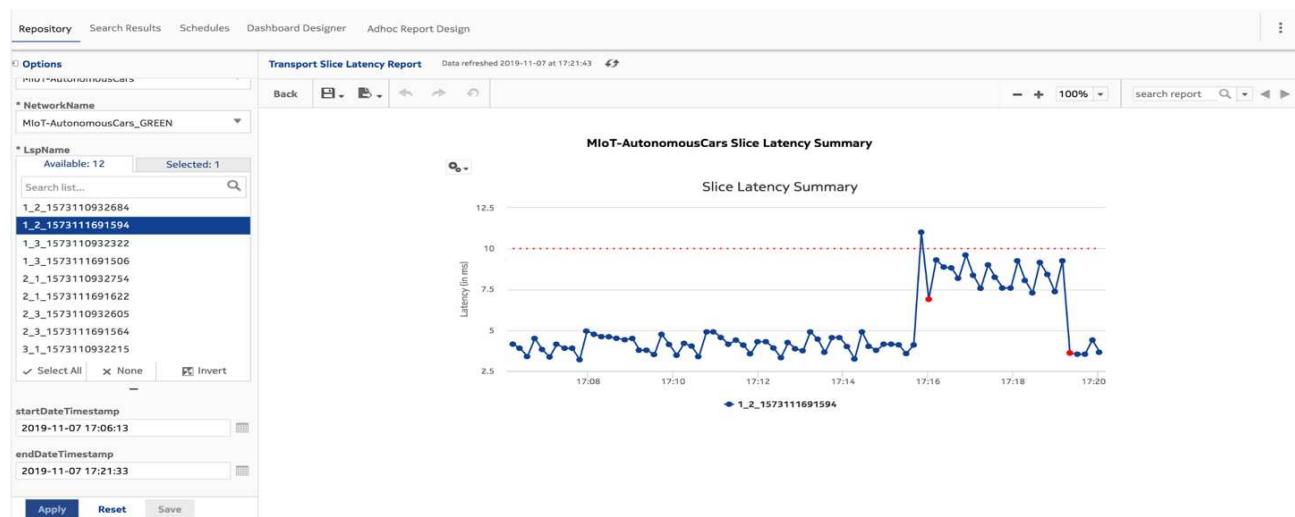


Figure 3-3, “Transport slices in Analytics Reports” (p. 23) shows the transport slice reporting supported by TSC in **Data Collection and Analysis, Analytics Reports**. The reports dashboard provides two new reports for bandwidth and latency in context of a specific transport slices. Reports displayed here can also be saved.

Figure 3-3 Transport slices in Analytics Reports



---

## 3.2 How do I open the Transport Slice Controller?

1

Open **Network Health and Map, Overview**.

2

Click **TRANSPORT SLICES**. The Transport Slice Controller opens in a new tab.

END OF STEPS

---

## 3.3 How do I view the health of transport slices in the network?

This workflow describes the various main sections of the TSC dashboard. Each section provides a different view of transport slices and associated network components. The following sections are described here:

- **Transport Slices Overview**: this section provides at-a-glance views of all transport slices in the network, transport slices with down/degraded services or associated components, and unacknowledged alarms.
- **Data Page**: this section provides associated network component information for a selected transport slice bubble
- **Transport Slice**: this view provides detailed information about a specific transport slice and provides cross-launch capability where applicable

### Transport Slices Overview

1

View the **Health** panel.


This panel displays an at-a-glance view of all transport slices, including the Total Transport Slices, Total Degraded and Operational Disabled slices, and the affected services/LSPs.

You can click on one of the bubbles to automatically scroll down to the **Data Page** and view more detailed status information for those slices.

2

View the **Transport Slices** panel.

This panel displays list of all transport slices sortable by downed services, LSPs, SDPs, or SLA violations.

The following actions are available for transport slices in this list, by clicking on  for a transport slice:

- Transport Slice Detail: selecting this will scroll down to the **Data Page**.
- Resync RAN Applications



---

3

View the **Network Overview** panel.

This panel displays a breakdown of the transport slices in the network. The following are available:

- Number of Transport Slice per Customer
- Number of Transport Slice per Service Type

## Data Page

---

4

Select a transport slice bubble in **Transport Slices Overview** to automatically scroll to the **Data Page** section and view details for the slices.

This is useful for viewing all affected components associated with the total degraded or total operationally disabled slices.

---

5

View the following panels for information on network components associated with the currently selected transport slices:

- Transport Slices (this an at-a-glance view of the transport slices)
- e2e Network Slices
- Customers
- Services
- RAN Nodes
- Core Nodes
- BR Nodes
- PON Nodes

## Transport slice

---

6

Select a transport slice in **Transport Slices** panel and choose “Transport Slice Detail” from the contextual menu to automatically scroll to the **Transport Slice** section and view details for the slice.

You can also click on the Transport Slice Detail button in the **Data Page**.

---

7

This view contains several panels. They display network slices associated with the selected transport slice, and other information including the following:

- overall health

- 
- number of associated services, tunnels, endpoints, RAN applications, PON services, and the current status (down/degraded) of these items
  - Related Domain NEs

---

8

This view also contains various sortable list panels for the following:

- L0/L1/L2/L3 Services
- LSPs / Tunnels
- SDPs
- PON Services
- RAN Applications
- Alarms


### 3.4 How do I view details of the transport slice realization using L0/L1/L2/L3 services and tunnels/paths?

This workflow describes how to view transport slice information for services and tunnels/paths and cross-launch NSP views to display these components in further detail.

---

1

**View details:** Select “Transport Slice Detail” for a specific transport slice to navigate to the details view. You have two options for doing this:

- Under **Transport Slices Overview**, find the required transport slice in the Transport Slices panel, click , and select “Transport Slice Detail”.
- Under **Transport Slices Overview**, click on the bubble that contains the required transport slice. The view scrolls to the Transport Slices panel in the **Data Page**. In this list, find the required transport slice and click the “Transport Slice Detail” button.

The details for the transport slice are displayed in **Transport Slice** section.

---

2

View the information in the following panels:

- L0/L1/L2/L3 Services
- LSPs/Tunnels

---

3

**Cross-launch functions:** You can cross-launch other NSP functions from these panels.

- **L0/L1/L2/L3 Services:** cross-launch **Service Management**
- **LSPs/Tunnels:** cross-launch **Path Control**

- 
- **Radio Planes:** cross-launch **Network Map and Health**

END OF STEPS

---

### 3.5 How do I view reporting data on transport slices?

Reporting data for transport slices can be viewed in Analytics Reports, which you can cross-launch from the **Transport Slices** section of the TSC.

1

---

Select “Transport Slice Detail” for a specific transport slice to navigate to the details view. You have two options for doing this:

- Under **Transport Slices Overview**, find the required transport slice in the Transport Slices panel, click  , and select “Transport Slice Detail”.
- Under **Transport Slices Overview**, click on the bubble that contains the required transport slice. The view scrolls to the Transport Slices panel in the **Data Page**. In this list, find the required transport slice and click the “Transport Slice Detail” button.

The details for the transport slice are now displayed in **Transport Slice** section.

2

---

In the **Transport Slice** section, go to the Transport Slice panel.

3

---

Click **Go to Reports**. The Analytics Reports view opens. See the *NSP Data Collection and Analysis Guide* for more information about using this NSP function.

END OF STEPS

---

