5G transport slice control in end-to-end 5G networks

Best practices for a transport slice controller

White paper

Network slicing is a fundamental technology in 5G networking to enable concurrent delivery of end-to-end (E2E) services over a common physical transport network. Transport slicing in E2E 5G networks is critical for 5G success because it maximizes operational efficiency for zero-touch service delivery with support for deterministic SLAs on throughput, latency and availability.

A transport slice controller (TSC) is a key building block that coordinates the creation, monitoring and optimization of 5G transport slices in multi-domain, multi-technology and multivendor environments. This paper focuses on best practices for a TSC.

Contents

The importance of a 5G transport slice controller

Nokia views 5G transport slicing as highly important to the operational success of E2E 5G networks. There are several advantages for using transport slicing in E2E 5G networks, some of which are improved SLA adherence, operational simplification, ease of integration and implementation flexibility. We discuss transport slicing in detail and its benefits in our white paper "Transport slicing in end-to-end 5G networks: Maximize operational flexibility and efficiency to meet the demands of 5G".

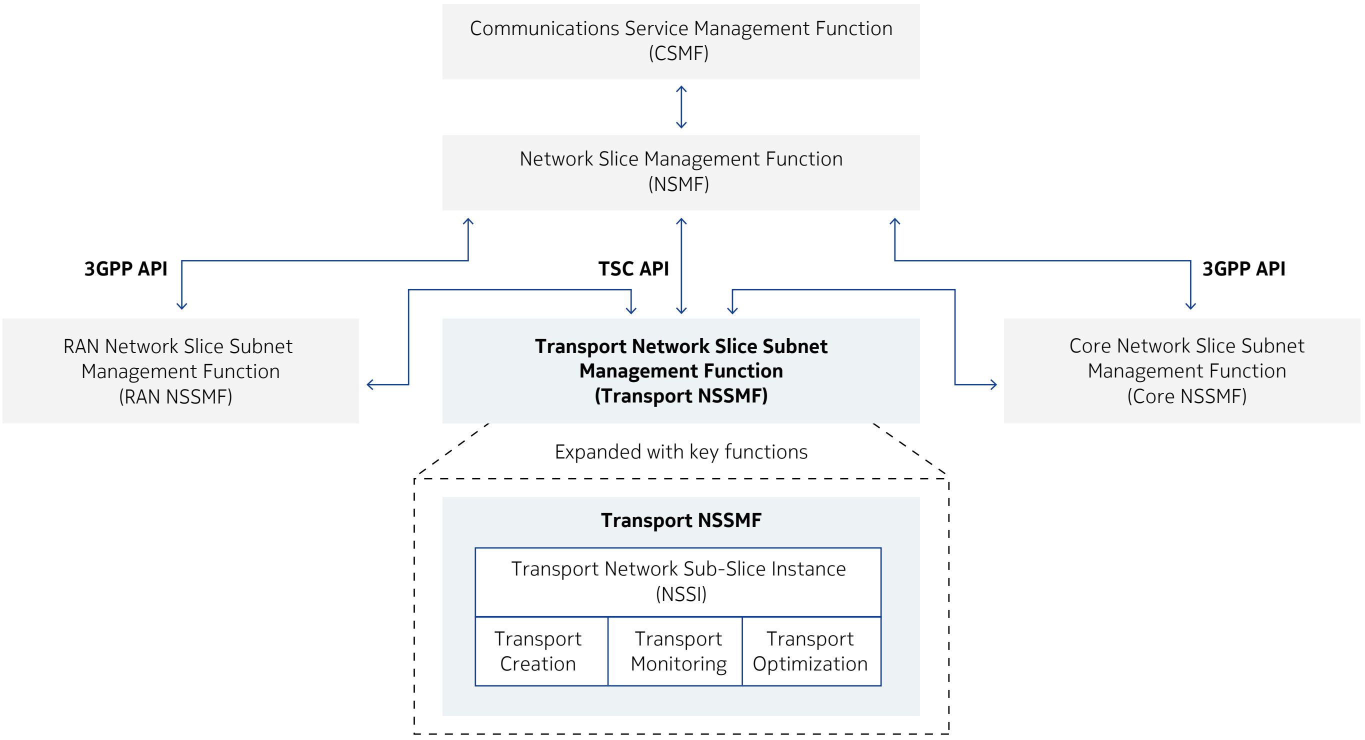
The introduction of a transport slice controller (TSC) will become most important for maximizing operational flexibility and efficiency as communications service providers (CSPs) evolve to E2E 5G network slices. This will become critical to accelerating a large range of E2E network deployment scenarios for 5G, including for 4G/5G hybrid networks. In turn, that will allow CSPs to leverage existing transport network investments to ease transformation to E2E 5G networks. 5G transport domain control will also enable the ability to better and more consistently enforce SLAs across all of the transport slice's implementation (realization) in the network.

The role of a TSC within the 3GPP 5G architecture

Figure 1 shows that with the 3GPP 5G architecture, the Communications Service Management Function (CSMF) interacts with a Network Slice Management Function (NSMF). This NSMF interacts with domain-specific Network Slice Subnet Management Functions (NSSMFs) for the RAN, transport and core, respectively.

Within this paper, the NSMF is referred to as the E2E network slice orchestrator, and the domain-specific NSSMFs are referred to as RAN, transport and core slice controllers.

Figure 1. 3GPP 5G architecture



Communications Service Management Function
(CSMF)

Network Slice Management Function
(NSMF)

3GPP API

TSC API

3GPP API

RAN Network Slice Subnet
Management Function
(RAN NSSMF)

Transport Network Slice Subnet
Management Function
(Transport NSSMF)

Core Network Slice Subnet
Management Function
(Core NSSMF)

Expanded with key functions

Transport NSSMF

Transport Network Sub-Slice Instance
(NSSI)

Transport Creation	Transport Monitoring	Transport Optimization

The Transport NSSMF plays an important role in the 3GPP 5G architecture in that it ensures that all transport slices belong to a single E2E 5G network slice adhering to required service level agreements (SLAs). Transport slices are called Transport Network Sub-Slice Instances (NSSIs) in 3GPP standards, and the Transport NSSMF performs the key functions of transport slice creation, monitoring and optimization. The Transport NSSMF is referred to in this paper as transport slice controller (TSC).

The 3GPP currently defines the interfaces for the NSMF to communicate to both the RAN NSSMF and core NSSMF. However, currently the 3GPP has not yet defined the same interface for the transport domain, which is referred to as the TSC API or “transport slice connectivity interface” within this paper. The TSC presents the E2E network slice orchestrator with an abstract interface for the programming of transport slices across fronthaul, midhaul, backhaul, cloud interconnect and overlay network connectivity.

Best practices for a TSC

A fundamental capability of a TSC is to allow transport slices to be implemented across various vendors’ equipment with any supporting technology. For example, the multivendor support of a TSC should allow transport slices to use the following technologies and services:

- **Networking technologies**: IP, optical, passive optical network (PON) or microwave
- **Tunnel types**: IP, MPLS, segment routing or optical data unit (ODU)/optical channel (OCH)
- **Service types**: L0/L1/L2/L3

The TSC should enable the ability to better and more continuously enforce SLAs across all of the transport network slices’ implementation end-to-end (i.e. across fronthaul, midhaul, backhaul, cloud interconnect and overlay network connectivity).

As TSCs are introduced to the market, it is expected that many solutions may initially fall short in performing all the required functions for full life-cycle management (LCM) and control of transport slices or may perform functions with a limited single vendor focus. It should be expected that in many cases, vendors that do not sell E2E networking solutions will have little choice but to integrate external products, microservices platforms and architectures to fill gaps. This reality underscores the importance of open programmability across the multiple 5G transport vendors to enable third-party integration. This is a hard requirement for all TSC vendors.

However, it is also ideal (where possible) for much of this open programmability and multivendor support to be provided by a single, unified, programmable interface into a multivendor network automation platform. Especially in modern IP and optical networks, intent-based networking (IBN) systems with model-driven frameworks improve traditional transport domain control and management systems by enabling DevOps. This approach offers significantly faster development times for natively supporting network equipment from multiple vendors as well as enabling more rapid systems integration.

It is important to recognize the TCO, TTM agility and reliability advantage of pre-integrated solutions. Bell Labs Consulting has recently released a publicly available report entitled “Single Integrated Networking Solutions,”[1] which studies the value of a single integrated solution (SIS) toward digital transformation, concluding that a SIS provides a superior alternative to today’s multivendor solutions. This is a significant consideration specifically for delivering pre-integrated TSC functions for automation, optimization and assurance of transport slices within a SIS.

---

[1] Bell Labs Consulting: Narayan Raman et al, “Single Integrated network solutions - The TCO, agility and reliability advantage of pre-integrated solutions”, February 2019.

The remainder of this paper will outline specific best practices Nokia followed when implementing TSC programmability with a transport slice connectivity interface as well as supporting capabilities for automating the creation, assurance and optimization of transport slices.

Programmability with a transport slice connectivity interface

As discussed with Figure 1, to integrate with the E2E 5G network slice orchestrator, the TSC must provide an API that abstracts and simplifies integration and implementation for 5G transport slice automation and control. The flexibility and capabilities exposed by this TSC API should allow for easy setup of monitoring policies, closed-loop automation policies and fine-grained programmability for transport slice implementation. Minimally the TSC API must provide capabilities for the creation, assurance and optimization of 5G transport slices.

Nokia believes that the information model for this TSC API is of key importance for 5G success. Although the 3GPP standards for defining the TSC API are still evolving, Nokia is working to lead IETF standardization to set out best practices for implementing a 5G transport slice connectivity interface, including its information model.[2]

As described in this IETF draft (pages 21–22), the proposed transport slice information model should include the following building blocks (for more information on these building blocks, please refer to (draft-rokui-5G-transport-slice):

- **transport-slice-info**: Information related to transport slice
- **e2e-network-slice-info**: A list of all E2E network slices mapped to transport slice
- **transport-slice-groups**: Each transport slice is a set of networks. Each network contains:
  - **list of endpoints** (i.e. list of RAN and Core network functions)
  - **list of connection-links** (i.e. list of connections between nodes)
  - **list of transport-slice-policies** (i.e. various SLA, Selection and Monitoring policies)

See Appendix A at the end of this paper for more information on how the TSC API fits in with the logical flow for creation of E2E 5G network slices.

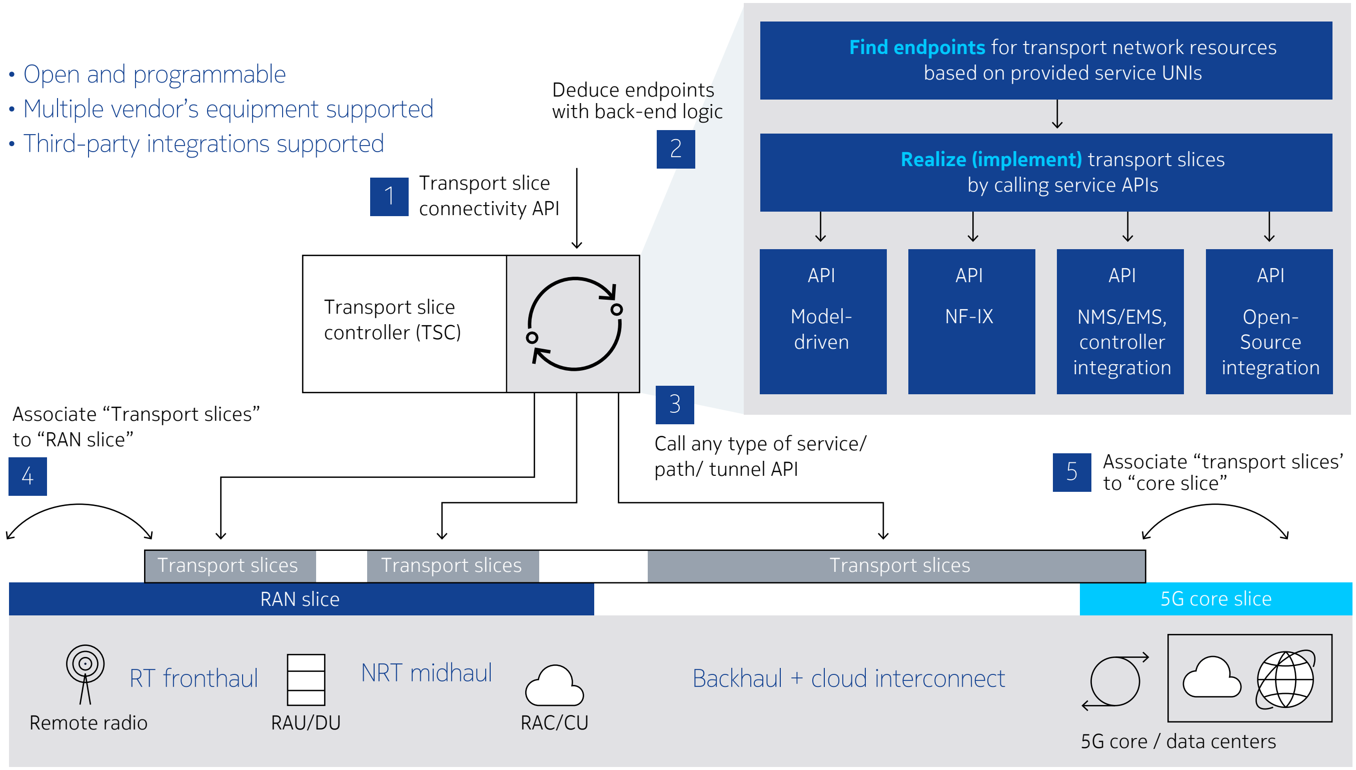
Automating transport slice creation

The primary role of a TSC is to provide network automation and programmability for the creation and maintenance of transport slices. Figure 2 is an example that shows the level of flexibility and programmability required for a best-in-class TSC when receiving requests through its northbound API and how the TSC should function to realize the implementation of the transport slice.

---

[2] Reza Rokui (Nokia), IETF Draft, “5G Transport Slice Connectivity Interface”, https://tools.ietf.org/html/draft-rokui-5g-transport-slice-00, July 2019.

Figure 2. TSC sequence flow from API request to transport slice implementation



Step 1 (in Figure 2): API creation request

As discussed previously, the TSC will typically receive a transport slice creation request from the E2E network slice orchestrator through TSC APIs (See Figure 2, step 1.) This transport slice will contain the connections between various network functions, for example between 5G RAN network functions (the RAN slice) and 5G Core network functions (the Core slice). In the case of a cloud RAN, this might also be connections in fronthaul or midhaul networks between RUs, DUs and CUs.

The request using the TSC API will supply a minimal set of information (as previously described when discussing the transport slice information model). The information always includes a "transport-slice-policy" that represents, in a generic and technology-agnostic way, the SLA requirement needed to realize the transport slice (e.g., it contains the requirements such as bounded latency, bandwidth, reliability, and security).

Note that superior abstraction concepts may be provided by the TSC API through implementing the TSC as an Intent-based Networking System (IBNS), which would provide further benefits to ease integration development simplification and enhance programmability through model-driven workflows.

Step 2 (in Figure 2): Finding endpoints for transport network resources

When it comes to implementing API requests for a TSC, it can be made significantly easier for integrations to be developed (and with improved automation) if the TSC does not need to be queried to understand transport network endpoints, network equipment or connections.

The recommended best practice is for the TSC to allow RAN and/or core endpoints to be specified and for the TSC to deduce the specific transport endpoints for the requested transport slice. For example, this could be for cloud RANs to connect an RU to a DU (fronthaul), or DU to CU (midhaul), or CU to the core (backhaul). In any of these cases, by using back-end logic, the TSC would find the correct border routers and their respective endpoints toward the 5G RAN and/or 5G Core (as Service UNIs), without integrations having to understand the transport network. This would include finding additional details required (e.g., VLAN-ID and IP addresses), which would enable the appropriate transport network resources to be found to implement the slice on. Note that this back-end logic requires the TSC to perform abstract topology discovery to understand connectivity between domains. Service UNIs passed into the TSC API could either be auto-assigned or provided by an external engine or whatever is already in use by the RAN.

Taking this best practice further, any deduced transport endpoint information that may be required to pass on to other systems (regardless of vendor or domain) can be stored by the TSC in an easily accessible, persistent format, such as a "transport-slice-endpoint-information" file, which may be securely transferred between systems, or made accessible through a set of open APIs. For example, this information could be used by the RAN or core slice controller or E2E network orchestrator to configure another vendor's equipment in either the RAN or core domain to complete the connection on the other side. More specifically, this could be for configuring RUs, DUs or CUs, in the case of a cloud RAN, or physical RAN equipment, or for 4G or 5G core endpoints.

The approach keeps the access to information sharing between systems open and flexible so that CSPs are not limited. For example, with this capability, a E2E 5G network slice orchestrator could be programmed to either directly retrieve the "transport-slice-endpoint-information" to pass it to a RAN or core slice controller, or pass the location (e.g., file path) to the controller for it to obtain the "transport-slice-endpoint-information" from a direct integration through an open API.

Step 3 (in Figure 2): Implementing (realizing) transport slices

When implementing transport slices, a best principle in TSC design would require the flexibility and programmability for the TSC to call on various service/path/tunnel APIs, not only to be able to use inherent multivendor TSC capabilities, but also to be able to call third-party solutions. It is important that this logic on the TSC be fully flexible and programmable so that it can dynamically implement the transport slices in the transport network based on transport technology, resource availability, various policies, and profiles that will insure that the desired L0-L3 services, paths and tunnels are created in the multivendor network based on characteristics requested of the transport slice.

Please see Appendix B at the end of this paper for more information on TSC service/tunnel/path APIs, including the:

- Model-driven API
- Network Functions Interconnect (NF-IX) API
- NMS/EMS controller integration API
- Open-source integration API.

Step 4 and 5 (in Figure 2): Associate "transport slices" to "RAN slice" and "core slice"

The next step is to associate the transport slices to the associated RAN slice. The TSC must provide an open API to the E2E network slice orchestrator or RAN slice controller in order to enable the transport slices to be associated to the RAN slice for the specific E2E 5G network slice. This E2E 5G network slice is identified by an ID called the Single Network Slice Selection Assistance Information (S-NSSAI).

After the RAN slice is associated, the next step is to associate the transport slices to the core slice. Similarly, the TSC open API to the E2E network slice orchestrator or core slice controller enables the transport slices to be associated to the core slice for the specific E2E 5G network slice, identified by the same S-NSSAI.

See Appendix C at the end of this paper for additional related information on using cloud interconnection automation for associating transport slices with a 5G cloud RAN and/or core.

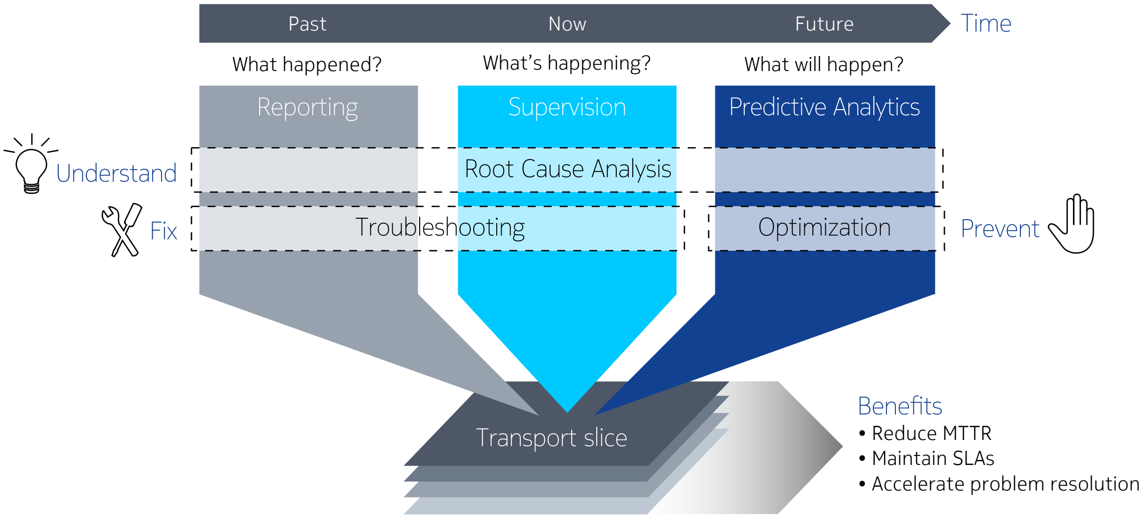
Monitoring and analytics for transport slice assurance

To make transport slice automation work well in live network deployments, it is critical that a TSC needs to support the full operations life cycle, which includes assurance providing real-time monitoring and analytics on 5G transport slices.

The "transport-slice-assurance-policy" is used, as specified within the IETF draft transport slice information model (draft-rokui-5G-transport-slice). It contains the type of assurance needed, time interval, how often to inform the E2E network slice orchestrator and other assurance and monitoring-related criteria. The TSC will receive the telemetry data continuously from the network for all the resources used during the transport slice implementation and aggregate them into key metrics to measure for SLA adherence. These aggregated key metrics form part of the real-time transport slice SLA, which will be used for various analytics inside the TSC as well as for enabling various types of reporting on transport slices. In addition, these key metrics will also be sent to the E2E network slice orchestrator for further analytics and aggregation.

To reduce mean time to repair (MTTR) and maintain SLAs, as a best practice, the TSC should augment transport slice assurance and monitoring to provide dynamic supervision and reporting capabilities for helping operations to understand what is happening in the PRESENT (now) in the network (in near-real-time), what has happened in the PAST, as well as predictive analysis that helps operators look toward what will likely happen in the FUTURE (see Figure 3).

Figure 3. Augment transport slice assurance and monitoring to understand past, present and future

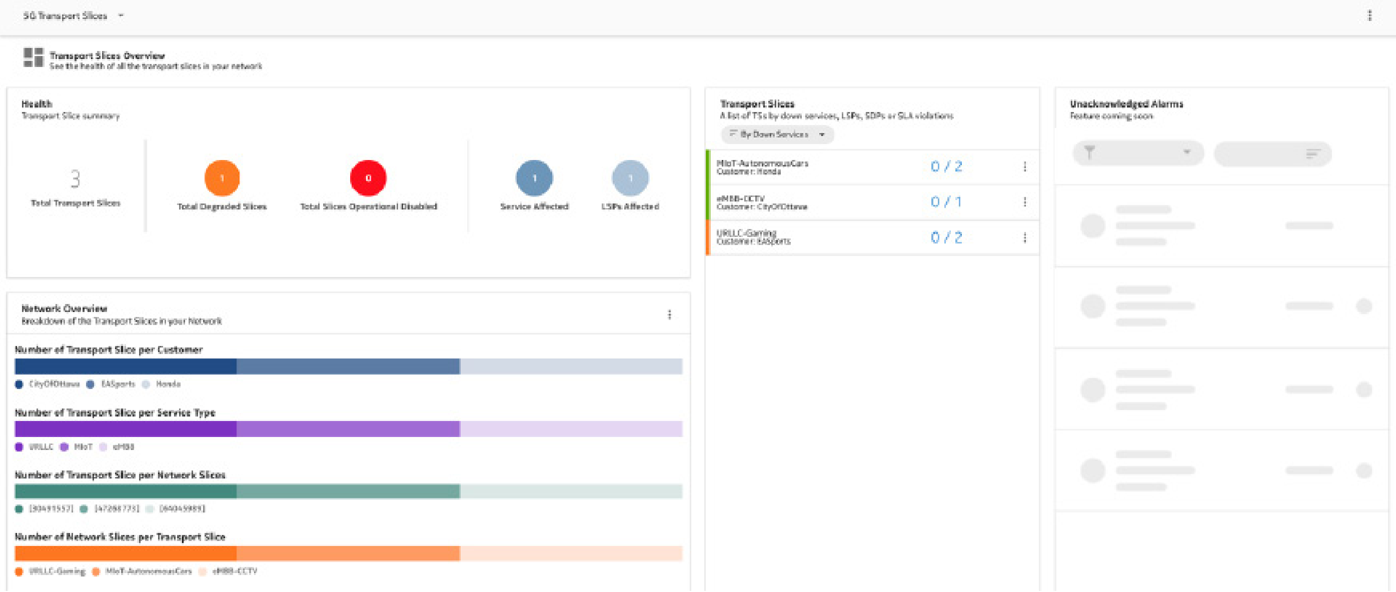


A TSC should provide efficient tools needed to quickly troubleshoot issues with transport slices, provide root cause analysis, and bring visibility to fixed (past) and current issues with these slices. It should also enable operators to make changes to prevent issues before these incidents actually occur and impact SLAs (e.g., to trigger optimizations, as discussed in the next section on transport slice optimization). Advanced TSC use cases that dynamically trigger closed-loop assurance can be achieved by implementing the TSC as an IBNS.

To best support assurance and monitoring for transport slices, a TSC best practice is to implement Transport Slice Operations applications. For example, "Transport Slice Health" dashboard views can provide a summary of the overall health of all transport slices in the network, as well as allow drill-down into health views on a per-slice basis.

Recommended aspects for the TSC to show in the health dashboard view on overall health of all transport slices in the network would include the health of the transport slices, various mappings of transport slices, and a list of transport slices and alarms for each transport slice listed (shown in Figure 4).

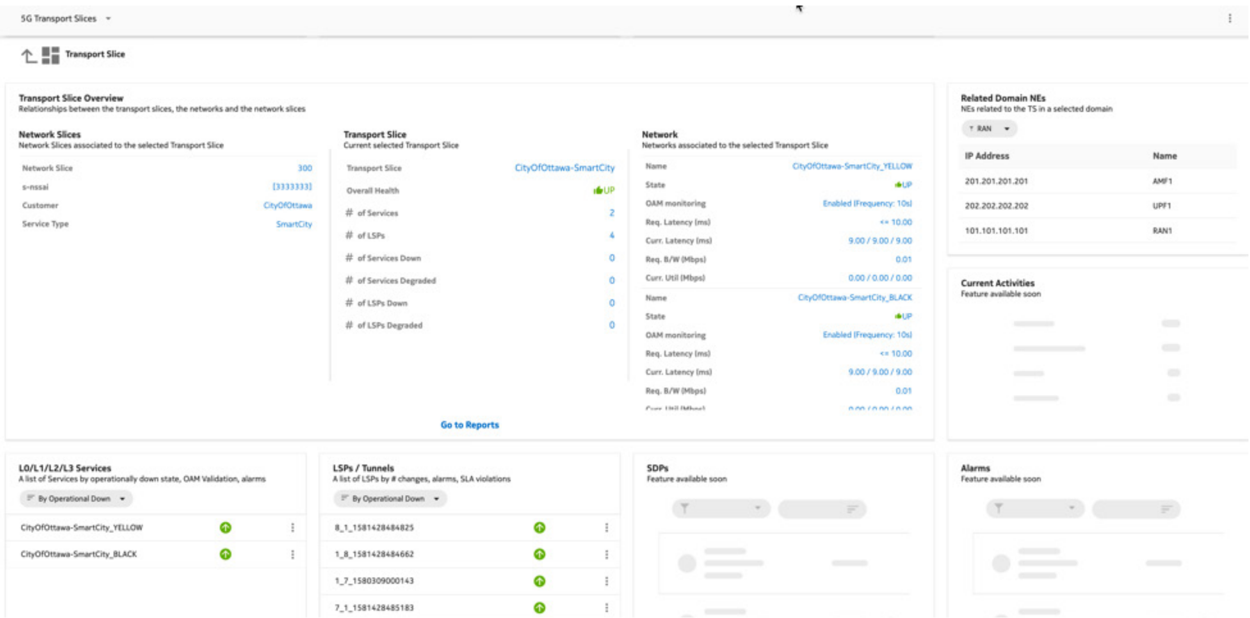
Figure 4. Example of health dashboard for all transport slices in network



MIoT-AutonomousCars Customer: Honda	0 / 2	⋮
eMBB-CCTV Customer: CityOfOttawa	0 / 1	⋮
URLLC-Gaming Customer: EASports	0 / 2	⋮

Recommended aspects for the TSC to show in the health dashboard view, which summarizes the health per transport slice, would include a list of E2E network slices that map to the specific transport slice being shown, as well as the 5G RAN and core slices using it. The health dashboard view would also show more detailed information for this transport slice's SLA, including whether all of the various parts of the SLA are being met. It is also recommended that health information be provided on all services, paths and tunnels used to implement (realize) the specific transport slice (shown in Figure 5).

Figure 5. Example of dashboard showing more details on a specific transport slice



Network Slices
Network Slices associated to the selected Transport Slice

Network Slice	300
s-nssai	[3333333]
Customer	CityOfOttawa
Service Type	SmartCity

Transport Slice
Current selected Transport Slice

Transport Slice	CityOfOttawa-SmartCity
Overall Health	UP
# of Services	2
# of LSPs	4
# of Services Down	0
# of Services Degraded	0
# of LSPs Down	0
# of LSPs Degraded	0

Network
Networks associated to the selected Transport Slice

Name	CityOfOttawa-SmartCity_YELLOW
State	UP
OAM monitoring	Enabled (Frequency: 10s)
Req. Latency (ms)	<= 10.00
Curr. Latency (ms)	9.00 / 9.00 / 9.00
Req. B/W (Mbps)	0.01
Curr. Util (Mbps)	0.00 / 0.00 / 0.00
Name	CityOfOttawa-SmartCity_BLACK
State	UP
OAM monitoring	Enabled (Frequency: 10s)
Req. Latency (ms)	<= 10.00
Curr. Latency (ms)	9.00 / 9.00 / 9.00
Req. B/W (Mbps)	0.01

IP Address	Name
201.201.201.201	AMF1
202.202.202.202	UPF1
101.101.101.101	RAN1

CityOfOttawa-SmartCity_YELLOW	⬆
CityOfOttawa-SmartCity_BLACK	⬆

8_1_1581428484825	⬆
1_8_1581428484662	⬆
1_7_1580309000143	⬆
7_1_1581428485183	⬆

Another important assurance capability for TSC health dashboards would be to allow operations to seamlessly cross-launch surrounding assurance applications for problem investigation and troubleshooting. For example, within the Nokia Network Services Platform (NSP), 5G TSC dashboards allow for cross-launching of applications for Network Supervision, Service Supervision, Analytics & Reporting, and Fault Management.

In addition, as stated previously when we introduced best practices for a TSC, it is important to recognize the TCO, TTM agility and reliability advantage of a single integrated solution (SIS). There are added assurance values for TSCs that are implemented as a SIS together with a domain controller for the 5G core. In these cases, the TSC transport slice dashboards may also allow for cross-launch to pre-integrated Core Slice Operations applications and dashboards, which focus on assuring the core slice associated to the transport slices being examined.

Closing the loop on transport slice optimization

The closed-loop optimization of transport slices is important for monitoring transport slice SLAs so that transport slices can avoid latency and congestion across the transport network. With a key goal of maintaining the highest reliability and quality, transport network optimization has significant importance for 5G network slices by enabling them to meet requirements for dynamically adapting to changing traffic patterns. The TSC should provide automation for network engineering and traffic steering to ensure strict adherence to SLAs (such as for low latency) and to meet increasing bandwidth demands from the transport network. This will enable greater 5G service innovation by making scale-out of new services easier and faster. It will also allow for load-balancing network usage on existing assets to enable OPEX and CAPEX savings.

Without a TSC, traffic steering decisions cannot be based on actual utilization or throughput SLAs, and there is no way to control paths across transport networks spanning multiple domains. To avoid congestion a TSC is required for creating and enforcing “transport-path-selection-policies” that address latency and bandwidth guarantees.

In some deployments, the E2E network slice orchestrator might want to assist the TSC on how to realize a transport slice by providing some information regarding the type of technologies and tunnels. This information will be provided in a “transport-slice-selection-policy”, as specified within the IETF draft transport slice information model (draft-rokui-5G-transport-slice).

Conclusion

To accelerate evolution toward 5G, it is important for a TSC to fit within the 3GPP 5G architecture as well as deliver on best practices that will enable operators to maximize efficiency. Following best practices for programmability, automation, optimization and assurance of transport slices within the context of E2E 5G networks will enable zero-touch service delivery and full LCM toward meeting the deterministic SLAs needed for 5G services.

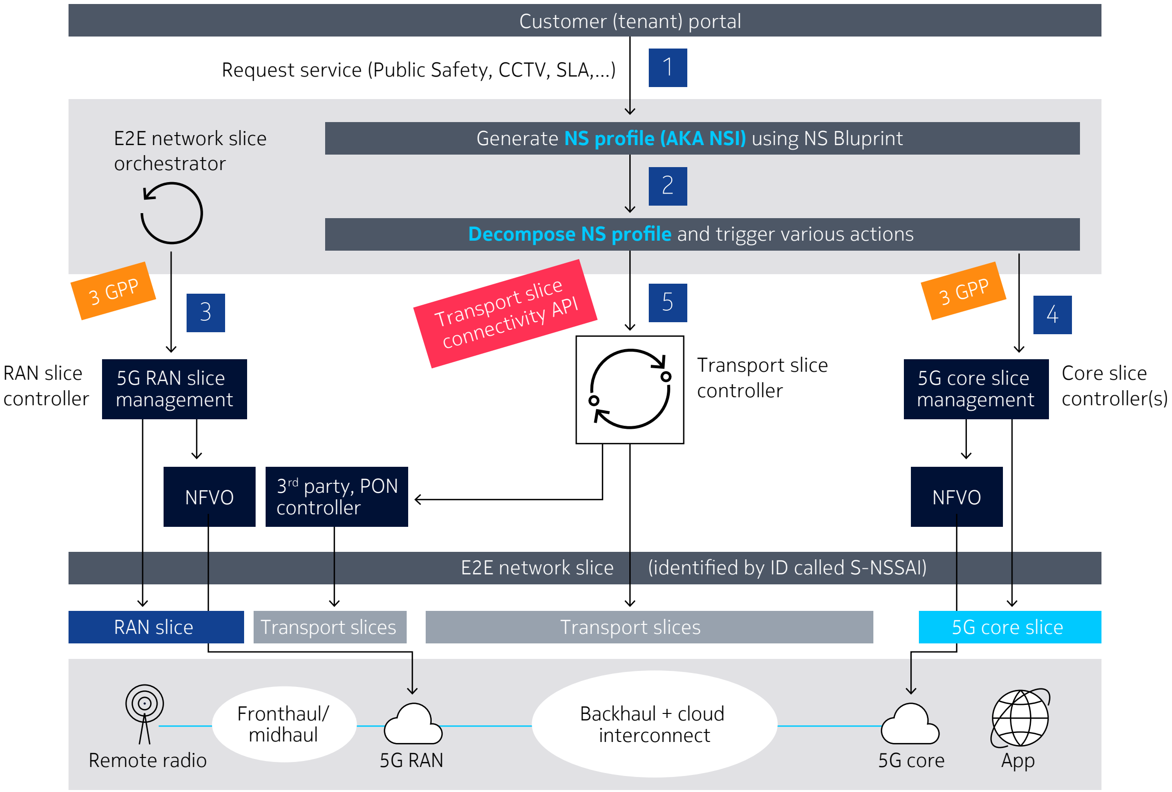
Nokia is pioneering best practices for 5G transport and core slice controllers through the open programmability, network automation and control delivered by the Network Services Platform (NSP).

For more information, please refer to our related resources, including our IETF presentation on 5G impact on networks, edge cloud and slicing.

Appendix A: Logical flow for creation of E2E 5G network slice

Figure 6 shows the logical flow for creation of an E2E 5G service for customer “Public Safety” and service type “CCTV Video Surveillance”.

Figure 6. Logical flow across various controllers for automation of an E2E network slice



The E2E network slice orchestrator receives an abstracted Service Catalog instantiation request from the customer portal. 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 subnets sent to various slice domain controllers.

The 5G 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.

Similarly, the 5G core slice controller (i.e., Core NSSMF) will receive a request to create the core slice. If the core is virtualized, it will in turn use the NFVO to initiate the VNF to create and then program the core slice. The 5G core slice controller uses a network service descriptor to create new EPC/5GC VNF components or data center networks if needed. At the end the TSC (i.e., Transport NSSMF) will act on instructions passed on in the transport slice connectivity API to instantiate various connections known as “transport slices” between RAN to core, between RAN RU-DU, or DU-CU and core to applications.

Note that interfaces to RAN and core controllers are defined in various 3GPP technical specifications, such as 28.531, 28.532, 28.540 and 28.541. However, currently there is no 3GPP standard defined for transport slices. Nokia is working with the IETF, BBF and several leading CSPs with the aim of providing clarification on this transport slicing interface, including the information model of this interface for automating the creation, monitoring and optimization of transport slices.

It is important also to emphasize that RAN, transport and core slice controllers and the E2E 5G network slice orchestrator will perform not only the automated creation of various slices, but the assurance and optimization as well (including monitoring, analytics and closed-loop optimization). In particular, the TSC enables CSPs to maximize efficiency by providing automation for transport slices and easing integration and implementation flexibility through abstracting the complexity of the underlying network for the E2E orchestrator. This allows the TSC to assist with the full LCM of transport slices by automating connectivity creation, optimization and monitoring.

Appendix B: TSC service/tunnel/path APIs

Model-driven API

This API provides communications with a multivendor framework for model-driven mediation (MDM) to support new equipment releases and service models at just-in-time speed. With MDM, device upgrades can be decoupled from full TSC upgrades. Forward-compatibility for supporting new devices and service models can be inherently provided by the TSC with MDM as well—without requiring a platform upgrade. This new paradigm shift to model-driven management is fundamentally different from the present mode of operations and delivers a dramatic improvement over the current process. For example, in the past, operators may have had to wait many months for some new equipment releases to be supported in transport domain controllers and EMS/NMS because equipment feature support and the necessary device and service object models needed to be changed in the controller/management system code base (by vendor software designers in the case of a vendor-supplied system, or by in-house developers in the case of home-grown systems). In many cases there were also further delays that resulted from waiting for the new release to be made available in the next upcoming vendor release cycle. In addition, the deployment timeline for vendor software needs to be planned, implemented and tested for platform-wide upgrades, which adds many more months to go live—especially when OSS integrations also need to be re-validated.

With MDM, a TSC can significantly reduce these deployment delays, which sometimes may last months, to a minimum—as little as hours or days in many cases, depending on the project scope. With MDM, new device features can efficiently be exposed to northbound systems by adopting new southbound and northbound models and by creating new adaptation scripts to translate between the two. There is no longer a need to change internal models.

The maximum level of automation is enabled by leveraging YANG modeling, which has become predominant in modern IP networks. With the YANG model being hot-deployed, a TSC support can be ready to use as soon as the YANG model is made available and deployed using the MDM. This is possible because the object models and TSC support are automatically derived from the YANG model. Support should also be provided out-of-the-box for many standardized YANG models, such as IETF L2 and L3 service models. In addition, GUIs (such as for model-driven configuration) and RESTCONF northbound APIs can be auto-generated from YANG models.

MDM and model-driven automation are key features of an IBNS, an advanced best practice for implementing a TSC. A TSC's model-driven API may also be further enhanced with programmable workflow management for fine-grained network and service control.

This model-driven API includes capabilities for creation of multivendor services (L0, L1, L2 and L3) by using operator-defined policies to guide dynamic network resource selection and automated provisioning. As a best practice, the TSC should provide network-aware service automation that uses real-time network visibility to meet service objectives in the most optimal way. A real-time view of the network (including link and tunnel status/utilization) is important for the TSC to be able to map service connection requests to the best available tunnels/paths (Layer 0, 1, 2, 3) that meet the SLA requirements and the operator's network efficiency goals (e.g., to perform service path selection based on latency versus bandwidth utilization versus traffic engineering cost).

Depending on the service type being implemented, to support third-party vendors in the transport network, this may involve leveraging a model-driven framework to support evolving YANG model standards (e.g., support IETF L2 and L3 service models and specific vendor's YANG models).

It may also involve integrations with third-party EMS/NMS or controllers or open-source software platforms (as described in each respective API description to follow).

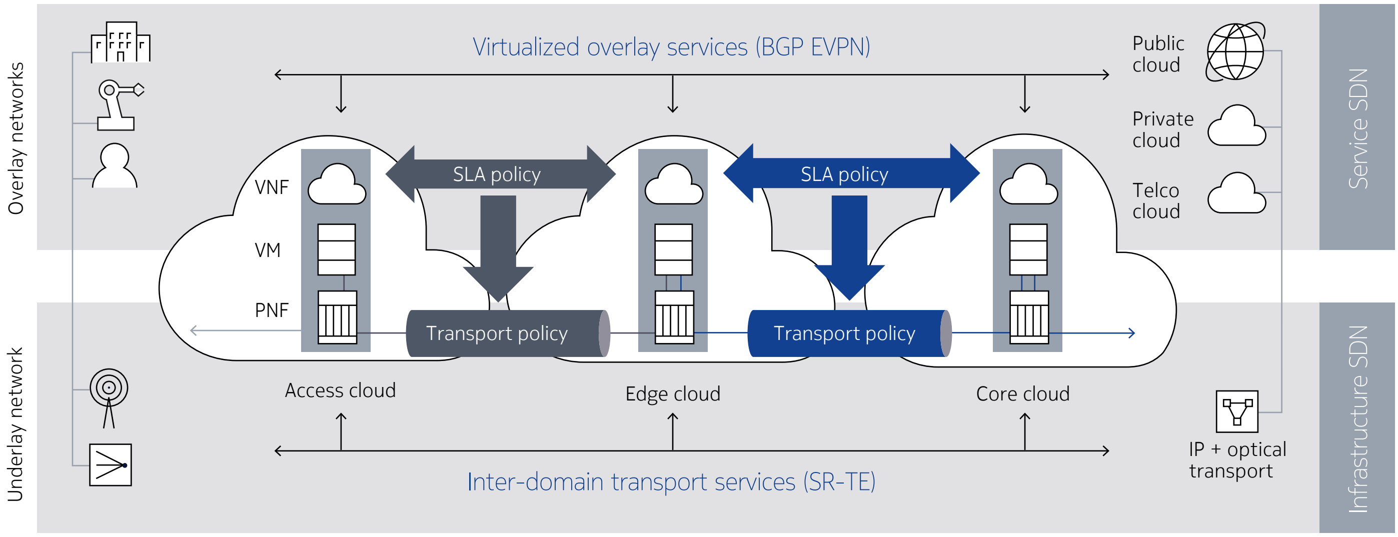
NF-IX API

This API enables communications with a Segment Routed Interconnect Controller (SR-IC) used with a Network Functions Interconnect (NF-IX) architecture. NF-IX envisions a unified, smart and dynamic network fabric that is designed to enable CSPs on the path to 5G to deliver programmable and cloud-optimized services with support for deterministic performance requirements.

NF-IX proposes allowing facilities to be shared by different slices in a robust and secure manner. This allows operators to construct efficient, high-capacity transport pipes between sites that are able to carry all the required traffic, rather than having to physically partition and provision many smaller pipes.

NF-IX allows the creation of a deterministic path or tunnel between various physical or virtual network functions (PNFs or VNFs) end to end (see Figure 7).

Figure 7. Nokia NF-IX connecting PNFs and VNFs from one data center to another



For example, the Nokia NF-IX can be used to connect PNFs and VNFs from one data center to another through the WAN underlay by leveraging the Border Gateway Protocol (BGP) and BGP MPLS-based Ethernet VPNs (RFC 7432) to acquire topology information from the virtualized service overlay and IP/MPLS WAN underlay. Segment routing is used to dynamically engineer inter-domain service tunnels between network functions that can meet deterministic bandwidth, latency and path diversity constraints end-to-end.

To learn more about the Nokia NF-IX architecture, please read our white paper "Network Functions Interconnect Architecture: A dynamic and smart network fabric for mobile broadband evolution, the Internet of Things and 5G", and watch our Nokia SReXperts demo to see NF-IX in action.

EMS/NMS, controller integration APIs

These APIs enable communications with pre-integrated EMS/NMS and controller solutions. This enables support for the TSC to support southbound integrations with third-party software across various domains of the E2E 5G network. For example, a TSC may use an MDM framework to support third-party IP/MPLS equipment natively, or instead integrate with an existing third-party EMS/NMS or controller that already provides support for the third-party devices. For multi-layer, cross-domain support, a TSC's open, programmable integration capability to support a third-party vendor's equipment may be the most feasible approach (e.g., to integrate third-party vendors for the optical, PON and microwave domains). It is expected that E2E 5G network vendors will natively support TSC integrations across all domains (e.g., Nokia NSP integrates with the access domain through the Nokia Altiplano Access Controller, NetAct, EdenNet, etc). However, it is important to note that support for integrating third-party equipment is mandatory to enable the flexibility of choice to use whichever vendors are determined to be the best fit for each individual operator's needs and network environment.

EMS/NMS, controller integration APIs may also provide unique capabilities for the TSC to improve E2E automation by enabling cross-domain support between 5G RAN and core controllers. This would be in support of aiding an E2E network orchestrator within 3GPP 5G standards.

Open-source APIs

These APIs provide communications with various open-source frameworks and applications. Examples of customer and industry initiatives driving network and service automation include those from:

- **Standards definition organizations**: for example, TMForum, MEF, O-RAN, ETSI NFV, 3GPP SAS, ETSI Zero-touch Service Management (ZSM)
- **Open source projects**: for example, ONAP, OpenDaylight (ODL), OpenStack, ONOS, OPNFV, ONF, Open Source MANO, The Linux Foundation's ACUMOS, TensorFlow, Apache Software Foundation Spark/SystemML.

Nokia believes that ETSI ZSM as well as ONAP have significant importance for network orchestration and control, while O-RAN is an important project for RAN optimization and automation.

ZSM is an ETSI standardization project aiming to define an architecture spanning both physical and virtualized networks and resources. It will provide a common foundation enabling a diverse ecosystem of open source groups, including ONAP, to produce interoperable solutions. Founded in December 2017, the ZSM group members include DT, NTT, Telefonica, Telstra, Ericsson, Nokia, Huawei and ZTE. Nokia has a leading role in the project, including vice chair and architecture lead positions.

Open Network Automation Platform (ONAP) is an open source project offering a complete platform for management and orchestration of software-defined networks. It was created in 2017 as a merger of AT&T's ECOMP platform and China Mobile's Open-O project. Both operators and their vendor ecosystems (Amdocs for AT&T, Huawei and ZTE for China Mobile) are still the dominant contributors, having a combined 60% share of contributions. Nokia has been a leading network equipment vendor contributor to the ONAP project in the past several years, and we are working with CSPs on collaborative development projects for integration with the Nokia NSP where needed.

ONAP has similarities that complement Nokia's vision for E2E 5G networks and best practices for a TSC, especially in the areas of model-driven resource definition and resource assurance, which support a vendor-independent architecture.

Appendix C: Associating transport slices with a cloud RAN or core

In the case of a 5G cloud RAN or core, the DC-GW will be the delineation point between the transport and the cloud. For such cases, an NF-IX architecture provides enhanced cloud interconnect automation benefits to establish dynamic connectivity for specific VNFs (such as RUs, DUs, and CUs that will be a part of a RAN slice, or core functions like the AMF, SMF, UPF, etc. that will be a part of a core slice). It will connect these VNFs through the transport network (using the appropriate transport slice).

Nokia's demo on NF-IX in action describes this cloud interconnect automation in an example use case that shows newly created core VNFs being associated to a particular VLAN by a Virtual Infrastructure Manager (VIM), such as Nuage Networks Virtualized Services Controller (VSC). In this way, the new VNFs will automatically (and transparently without DC-transport software integration) be associated to a pre-created VLAN.

The NF-IX will have mapped to a specific transport network policy color that will implement adherence to a pre-determined SLA designed by the operator to support a specific type of slice. In the Nokia NF-IX demo, the policy color is mapped to a Segment Routing Traffic Engineering (SR-TE) policy to achieve the needed transport path implementation for meeting the SLA. With cloud interconnect automation from the 5G core to establish transport connectivity, it will be significantly more efficient for the TSC to be able to associate this NF-IX created transport connectivity to the required transport network slice (and transport S-NSSAI/sub-slice ID) once the request is received by its TSC API.

For the 5G core, the deployment of microservices architectures and software components will be more prevalent to achieve modularity, portability, separation of concerns of individual components, scalability, exposure of APIs toward third-party entities and adherence to standardized interface definition. For similar advantages, there are specific 5G core network functions that are shared or dedicated per 5G E2E network slice (specifically the candidates are AMF, SMF, UPF, NRF, NSSF and N3IWF), which have LCM events (e.g., slice configuration/re-configuration, scale up/down), as well as performance KPIs and threshold crossing alarms, where these triggers will be able to drive automation use cases for adapting transport slices to better meet SLAs. More advanced implementations of a TSC may be able to support a tight integration or be deployed as a SIS, with the 5G domain controller for the 5G core or Cloud Packet Core (CPC). Such a domain controller would consist of several microservices that pertain to the specific core network functions that will be candidates for improving cloud interconnect automation use cases. This is especially beneficial for the CPC functions, such as the AMF, SMF and UPF, which in the context of a new core slice will be spun up, configured, updated, upgraded, and terminated in a very dynamic way. Cloud interconnect use cases for core and transport control are not only centered around 5G network slice creation, but also include advantages for seamless operations and assurance for transport and core network slices, as well as for optimization and remediation.

Abbreviations

3GPP	3rd Generation Partnership Project
AMF	Access and Mobility Management Function
API	application programming interface
BBF	Broadband Forum
BGP	Border Gateway Protocol
CAPEX	capital expenses
CPC	Cloud Packet Core
CSMF	Communications Service Management Function
CSP	communications service provider
CU	central unit
DC	data center
DC-GW	DC Gateway
DU	distributed unit
E2E	end-to-end
ECOMP	Enhanced Control, Orchestration, Management & Policy
EPC	Evolved Packet Core
EMS	Element Management System
ETSI	European Telecommunications Standards Institute
IBN	intent-based networking
IBNS	intent-based networking system
IETF	Internet Engineering Task Force
KPI	key performance indicator
LCM	life-cycle management
MEF	Metro Ethernet Forum
MDM	model-driven mediation
MANO	Management and Orchestration
MPLS	Multiprotocol Label Switching
MTTR	mean time to repair
NF	Network Function
NF-IX	Network Functions Interconnect
NFV	Network Functions Virtualization
NFVO	NFV orchestrator

NMS	Network Management System
NRF	NF Repository Function
NRT	non-real-time
NSMF	Network Slice Management Function
NSP	Network Services Platform
NSSF	Network Slice Selection Function
NSSI	Network Sub-Slice Instance
NSSMF	Network Slice Subnet Management Function
N3IWF	Non-3GPP Inter Working Function
OCH	optical channel
ODU	optical data unit
ONAP	Open Network Automation Platform
ONF	Open Networking Foundation
ONOS	Open Network Operating System
Open-O	Open Orchestrator
OPEX	operating expenses
OPNFV	Open Platform for NFV
O-RAN	Open RAN
OSS	operations support system
PNF	physical network function
PON	passive optical network
RAC	radio admission control
RAN	radio access network
RAU	radio aggregation unit
RESTCONF	Representational State Transfer Configuration
RT	real-time
RU	radio unit
SAS	Statistical Analysis System
SBA	Service-Based Architecture
SIS	single integrated solution
SLA	service level agreement
SMF	Session Management Function
S-NSSAI	Single Network Slice Selection Assistance Information

SR-IC	Segment Routed Interconnect Controller
SR-TE	Segment Routing-Traffic Engineering
TCO	total cost of ownership
TMForum	TeleManagement Forum
TSC	transport slice controller
TTM	time to market
UPF	User Plane Function
UNI	user network interface
VIM	Virtual Infrastructure Manager
VLAN	virtual LAN
VPN	virtual private network
VNF	virtual network function
VSC	Virtualized Services Controller
vRAN	Virtual RAN
YANG	Yet Another Network Generation
ZSM	Zero-touch Service Management

Related resources

- White paper: "Transport slicing in end-to-end 5G networks: Maximize operational flexibility and efficiency to meet the demands of 5G"
- IETF "5G impact on networks – edge cloud and slicing"
  - Video
  - Presentation slides
- Reza Rokui (Nokia), "IETF Draft for Transport Slice Definition", https://tools.ietf.org/html/draft-nsdt-teas-transport-slice-definition-00, November 2019
- Reza Rokui (Nokia), "IETF Draft for 5G Transport Slice Connectivity Interface", https://tools.ietf.org/html/draft-rokui-5g-transport-slice-00, July 2019
- IETF Draft for "Network Slice Provisioning Models", Nov 2019, https://tools.ietf.org/html/draft-homma-slice-provision-models-02
- White paper: "Network Functions Interconnect Architecture: A dynamic and smart network fabric for mobile broadband evolution, the Internet of Things and 5G"

Nokia Oyj
Karaportti 3
FI-02610 Espoo, Finland
Tel. +358 (0) 10 44 88 000

Document code: SR2001040939EN (February) CID207124