ToC for Network Slicing - Revised Problem Statement

Draft V0.7 -21stth May 20167

Edited by Alex Galis

(**added expected contributors**)

Abstract

This document introduces Network Slicing problems and the motivation

for new work areas. It represents an initial revision of the Network

Slicing problem statement derived from the analysis of the technical

gaps in IETF protocols ecosystem. It complements and brings together

the silo efforts being carried out in several other IETF working groups

to achieve certain aspects of Network Slicing functions and operations.

Status of This Memo

This Internet-Draft is submitted in full conformance with the

provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering

Task Force (IETF). Note that other groups may also distribute

working documents as Internet-Drafts. The list of current Internet-

Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months

and may be updated, replaced, or obsoleted by other documents at any

time. It is inappropriate to use Internet-Drafts as reference

material or to cite them other than as "work in progress."

This Internet-Draft will expire on July 23, 2017.

Copyright Notice

Copyright (c) 2017 IETF Trust and the persons identified as the

document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal

Provisions Relating to IETF Documents

(http://trustee.ietf.org/license-info) in effect on the date of

publication of this document. Please review these documents carefully,

as they describe your rights and restrictions with respect

to this document. Code Components extracted from this document must

include Simplified BSD License text as described in Section 4.e of

the Trust Legal Provisions and are provided without warranty as

described in the Simplified BSD License.

Table of Contents

1. Introduction 2

1.1. Notes 4

2. Suggested Problems and Work Areas 4

2.1. Notes 6

4. Security Considerations 7

5. IANA Considerations 7

6. Acknowledgements 7

7. References 7

7.1. IETF References 7

7.2. Informative References 9

Authors' Addresses 10

# 1. Introduction

Network Slicing (NS) refers to the end-to-end or domain managed partitions

of physical and/or virtual network resources, network physical/virtual and

service functions [RFC7665] that can act as an independent instance

of a connectivity network and/or as a cloud network. Network resources

include connectivity, compute, and storage resources.

Network Slices considerably transform the networking perspective by

abstracting, isolating, orchestrating, softwarizing, and separating

logical network components from the underlying physical network

resources and as such they enhance Internet architecture principles

([RFC1958], [RFC3439], [RFC3234]), through the potential of introducing new

network architectures for service delivery [5G-ICN].

The management plane allocates the grouping of network resources (whereby

network resources can be physical, virtual or a combination thereof),

it connects with the physical and virtual network and service functions

([SFC WG]) as appropriate, and it instantiates all of the network and

service functions assigned to the slice. On the other hand, for slice

operations, the slice control plane functionality that may be operated

by slice tenant, takes over the control and governing of all the network

resources, network functions, and service functions assigned to the slice.

It (re-) configures them as appropriate and as per elasticity needs, in

order to provide an end-to-end service. In particular, slice ingress routers

are configured so that appropriate traffic is bound to the relevant slice.

Allocation of traffic flows to slices as may be based on simple rules

(relying on a subset of the transport coordinate, DSCP/traffic class, or

flow label), or may be a more sophisticated one (to be further defined)such

as enabling new slice specific constructs in the data

plane. Also, the flows to slice allocation rules that are specified for a

slice can be changed dynamically, based on some events (e.g. triggered by a

service request). The slice control plane is responsible for instructing

the involved elements to honour such needs.

Network operators can use NS to enable different services to receive

different treatment and to allow the allocation and release of network

resources according to the context and contention policy of the operators.

Such an approach using NS would allow significant reduction of the operations

expenditure. In addition NS makes possible softwarization, programmability

([RFC7149]), and the innovation necessary to enrich the offered services.

Network softwarization techniques [IMT2020-2015], [IMT2020-2016] may be

used to realise and manage [MANO-2014] network slicing. NS provides the

means for the network operators to provide network programmable

capabilities to both service providers and other market players without

changing their physical infrastructure. NS enables the concurrent

deployment of multiple logical, self-contained and independent,

shared or partitioned networks on a common infrastructure. Slices

may support dynamic multiple services, multi-tenancy, and the

integration means for vertical market players (e.g. automotive

industry, energy industry, healthcare industry, media and

entertainment industry, etc.)

The purpose of the NS work in IETF is to develop a set of protocols and/

or protocol extensions that enable efficient slice creation,

activation / deactivation, composition, elasticity, coordination /

orchestration, management, isolation, guaranteed SLA, and safe and secure

operations within a connectivity network or network cloud / data centre

environment that assumes an IP and/or MPLS-based underlay.

While there are isolated efforts being carried out in several IETF

working groups Network WG [I-D.leeking-actn-problem-statement 03], TEAS WG

[I-D.teas-actn-requirements-04], [I-D.dong-network-slicing-problem-statement],

ANIMA WG [I-D.galis-anima-autonomic-slice-networking], [IETF-Slicing1],

[IETF-Slicing2], [IETF-Slicing3], [IETF-Slicing4], [IETF-Slicing5],[IETF-

Mobility], [IETF-Virtualization], [IETF-Coding], [IETF-Anchoring] to

achieve certain aspects of network slice functions and operations,

there is a clear need to look at the complete life-cycle management

characteristics of Network Slicing solutions though the discussions

based on the following architectural tenets:

o Underlay tenet: support for an IP/MPLS-based underlay data plane the

transport network used to carry that underlay.

o Governance tenet: a logically centralized authority for network

slices in a domain.

o Separation tenet: slices may be virtually or physically independent of

each other and have an appropriate degree of isolation (note 1) from

each other.

o Capability exposure tenet: each slice allows third parties to

access via dedicated interfaces and /or APIs and /or programming methods

information regarding services provided by the slice (e.g., connectivity

information, mobility, autonomicity, etc.) within the limits set by the

operator or the slice owner.

NS approaches that do not adhere to these tenets are explicitly

outside of the scope of the proposed work at IETF.

In pursuit of the solutions described above, there is a need to

document an architecture for network slicing within both wide area

network and edge/central data center environments.

Elicitation of requirements (examples are [RFC2119], [RFC4364])

for both Network Slice control and management planes will be needed,

Facilitating the selection, extension, and/or development of the protocols

for each of the functional interfaces identified to support the architecture.

Additionally, documentation on the common use-cases for slice

validation for 5G is needed, such as mission-critical ultra-low latency

communication services; massive-connectivity machine communication

services (e.g. smart metering, smart grid and sensor networks); extreme

QoS; independent operations and management; independent cost and/or

energy optimisation; independent multi-topology routing; multi-tenant

operations; new network architecture enablement, etc.

The proposed NS work would be coordinated with other IETF WGs (e.g.

TEAS WG, DETNET WG, ANIMA WG, SFC WG, NETCONF WG, SUPA WG, NVO3 WG,

DMM WG, Routing Area WG (RTGWG), Network Management Research Group

(NMRG)and NFV Research Group (NFVRG)) to ensure that the commonalities

and differences in solutions are properly considered. Where suitable

protocols, models or methods exist, they will be preferred over

creating new ones.

## 1.1. Notes

(1) This issue requires efficient interaction between an upper layer

in the hierarchy and a lower layer for QoS guarantees and for most

of the operations on slicing.

# 2. Suggested Problems and Work Areas

The goal of this proposed work is to develop one or more protocol

specifications (or extensions to existing protocols) to address specific

slicing problems that are not met by the existing tools. The following

problems were selected according to the analysis of the technical gaps in

IETF protocols ecosystem.

I. Global Issues Problems (GP)

\* Problem GI1: Uniform Reference Model for Network Slicing (Architecture

document):

- Describes all of the functional elements and instances of a network

slice.

- Describes shared non-sliced network parts. Establishes the boundaries to

the basic network slice operations (creation, management, exposure,

consumption).

- Describes the minimum functional roles derived from basic

network slice operations including infrastructure owner (creation,

exposure, management), slice operator (exposure, management, consumption),

slice customer (management, consumption). Describe the interactions

between infrastructure owner <🡪 slice operator, slice operator <🡪 slice

operator, slice operator <🡪slice customer.

Additionally, this working area will normalize nomenclature and definitions

for Network Slicing.

**Alex**

Short explanation: A uniform definition and architecture of Network slicing is

presented in the NS Architecture draft.

A Network slice is a managed group of subsets of resources, network functions

/ network virtual functions at the data, control, management/orchestration

planes and services at a given time. Network slice is programmable and has

the ability to expose its capabilities. The behaviour of the network slice

realized via network slice instance(s).

(1) The Service Instance component

* Represents the end-user service or business services.
* An instance of an end-user service or a business service that is realized within or by a NS.
* Would be provided by the network operator or by 3rd parties.

(2) A Network Slice Instance component

• Represented by a set of network functions, virtual network functions and resources at a given time

• Forms a complete instantiated logical network to meet certain network characteristics required by the Service Instance(s).

• Provides network characteristics which are required by a Service Instance.

• May also be shared across multiple Service Instances

(3) Resources component – it includes: Physical, Logical & Virtual resources

• Physical & Logical resources - An independently manageable partition of a physical resource, which inherits the same characteristics as the physical resource and whose capability is bound to the capability of the physical resource. It is dedicated to a Network Function or shared between a set of Network Functions.

• Virtual resources - An abstraction of a physical or logical resource, which may have different characteristics from that resource, and whose capability may not be bound to the capability of that resource.

(4) Slice Element Manager (SEM) and Capability exposure component

• Slice Element Manager (SEM) is instantiated in each Network Slice and it manages all access permissions and all interaction between a Network Slice and external functions (i.e. other Network Slices, Orchestrators, etc). Each SEM converts requirements from orchestrator into virtual resources and manages

New protocols are needed for the creation, for discovery and for orchestrating network slicing

\* Problem GI2: Review common scenarios from the requirements for operations

and interactions point of view. Describes the roles (owner, operator, user)

which are played by entities with single /multiple entities playing different

roles.

Short explanation:

**Ravi – TO DO –- approx 0.5 page: Description of relevant**

**types of scenario/ UCs (Links with gap analysis & UCs documents) &**

**new/revised protocols needed**

\* Problem GI3: Slice Templates: Design the slices to different scenarios

([ChinaCom-2009], [GENI-2009], [IMT2020-2016bis], [NGMN-2016],

[NGS-3GPP-2016], [ONF-2016]); Outlines an appropriate slice template

definition that may include capability exposure of managed partitions

of network resources (i.e. connectivity ([CPP]), compute and storage

resources), physical and/or virtual network and service functions that can

act as an independent connectivity network and/or as a network cloud.

Short explanation:

**Liang - TODO – approx 0.5 page: Description & links**

**wit the Framework document & new/revised protocols needed**

**(Links with UCs & framework documents)**

II. Network Slice Capabilities Problems (NSC)

\* Problem NSC1: Four-dimensional efficient slice creation with guarantees for isolation in each of the Data /Control /Management /Service planes. Enablers for safe, secure and efficient multi-tenancy in slices.

**Carlos**

Short explanation: Network slices MUST support multi-tenancy, ensuring that

isolation and performance guarantees are provided at the data, control,

management and service planes. This involves the following:

* A network slice SHOULD provide a guaranteed level of service, according to a negotiated SLA between the customer and the slice provider.
* Slices MUST be isolated at service level (e.g., one slice must not impact on the level of service of the other slides, even if sharing resources).
* Slices MUST be isolated at data level, even if sharing resources. Security and privacy mechanisms should be in place to ensure this.
* A network slice SHOULD be provided with exclusive control and/or management interfaces (depending on the type of network slice), enabling the deployment of different logical network slices over shared resources**.**

\* Problem NSC2: Methods to enable diverse requirements for NS including

guarantee for the end-to-end QoS of service in a slice.

Short explanation:

**Ravi – TODO - approx 0.5 page: Description &**

**new/revised protocols needed (Links with UCs & framework documents)**

\*Problem NSC3: Efficiency in slicing: specifying policies and methods

to realize diverse requirements without re-engineering the infrastructure.

**Med**

Short explanation:This item is deployment-specific and cannot be promised as a problem to be solved by protocols. An underlying infrastructure will always be needed to be reengineered and maintained to support up-to-date technologies and emerging requirements (including instantiating new service functions or withdrawing service functions, adding new nodes to absorb for traffic, ...). It is a local decision to figure out whether many services will be bound to the same slice, how many slices are to be instantiated and so on. Exposing standard interfaces to capture requirements will help to rationale the use of resources and how the requirements are fulfilled, however it is a challenge to guarantee in an absolute manner that slicing allows "diverse requirements without re-engineering the infrastructure".

\* Problem NSC4: Recursion namely methods for NS segmentation allowing a

slicing hierarchy with parent - child relationships.

**Alex / Liang**

Short explanation:Recursion is a property of some functional blocks: a larger functional block can be created by aggregating a number of a smaller functional block and interconnecting them with a specific topology. As such recursive network slice definition is defined as the ability to build a new network slice out of existing network slice (s). A certain resource or network function /virtual network function could scale recursively, meaning that a certain pattern could replace part of itself. This leads to a more elastic network slice definition, where a network slice template, describing the functionality, can be filled by a specific pattern or implementation, depending on the required performance, required QoS or available infrastructure.

New protocols are needed for use of network slice template segmentation allowing a slicing hierarchy with parent - child relationships.

\* Problem NSC5: Customized security mechanisms per slice.

**Med**

Short explanation:Customized securing mechanisms will be needed on a per slice basis:

* + This may be provided by enabling dedicated service functions. For such cases, SFC techniques can be used here. Soliciting distinct SFs per slice can be provided with existing tools. I don't see a new problem out there.
  + This may be provided by configuring dedicated policies in a given security service function. In such case, I2NSF techniques can be used to interact with a given service function.
  + Traffic isolation may be needed for some services. Legacy tools can be used. I'm not sure if there is specific work specific to slicing other than making sure that appropriate flows are grafted to the appropriate slice and no data leaking between slices is to happen.

\* Problem NSC6: Methods and policies to manage the trade-offs between

flexibility and efficiency in slicing.

**Carlos**

Short explanation: Mechanisms SHOULD be in place to allow different

levels of flexibility when providing network slices:

* from the ones that provided greater levels of flexibility in the provided resources and services that compound the slice, allowing to dynamically change/scale/migrate it over time within a negotiated range,
* to the ones that ensure the efficiency of the use of the resources at the cost of a smaller degree of flexibility.

\* Problem NSC7:Optimisation: namely methods for network resources automatic

selection for NS; global resource view formed; global energy view

formed; Network Slice deployed based on global resource and energy

efficiency; Mapping algorithms.

Short explanation:

**Luis – Todo approx 0.5 page: Description & new/revised**

**protocols needed (Links with UCs & framework documents)**

\* Problem NSC8: Monitoring status and behaviour of NS in a single and/or

muti-domain environment; NS interconnection.

**Alex**

Short explanation: A Network slice is a managed group of subsets of resources, network functions / network virtual functions at the data, control, management/orchestration planes and services at a given time. Monitoring of slices interacts with and it is part of the NS Lifecycle management to aiming at reporting the performance of the running NS. As input, the Monitoring Subsystem receives the detailed service monitoring requests with references to resource allocation and Network functions instances in a NS. The Monitoring Subsystem is responsible for the monitoring continuously the state all 4 components of a NS (Service Instance component, Network Slice Instance component, Resources component).

New protocols are needed for discovery and monitoring probes of all NS

components.

\* Problem NSC8: Capability exposure for NS; plus APIs for slices.

**Pedro**

Short explanation: To exploit the flexibility offered by network slices they must be structures in a way their users (customers, overlying operators) are able to know the features offered by both individual resources and complete slices. This means that there must be interfaces to deliver such information to the entity that needs it, but that will be also transitively delivered to the following chains of the slicing structure towards the final users.

To this sense, there are two specific interfaces that must be defined to address such function:

- The bottom-up interface, offered by underlying resource providers to resource consumers (operators) of any layer.

- The top-down interface, offered by overlying operators to lower level providers.

On the one hand, the first interface will, obviously, enable slice operators to access the slices owned by underlying providers and manage the resources they have been assigned in them. On the other hand, the second interface will enable lower layers to know details about the resources managed by overlying operators and the requirements they impose to the overlying network slices.

To this respect, both interfaces will emphasize the relation among the original resources, as well as the links from them to the resulting resources. This forms the main key of their management operations.

\* Problem NSC9: Programmability and control of Network Slices.

**Slawomir**

Short explanation: Network slice operations consist of all operations related to life cycle management of a slice and its optimized operation. Slice instance lifecycle management includes all operations related to slice instance creation, activation, update and deactivation. All these operations are automated and driven by appropriate policies. A slice instance is created according to a slice template and related policies. A unique identifier is assigned to each slice after its creation and a list of active slice instances are stored in slice repository. Several slice types are predefined which describe their functions as access, core, transport, data center and edge cloud slices. To each slice instance a Slice Priority parameter is assigned which describe the way of handling of slice degradation in case of lack of resources that can be allocated to slices. The parameter is also used in emergency situations in which there is a need to release resources from existing slices and to allocate them to newly created slices that are used for emergency situation handling.

The end-to-end slice can be a composition of per administrative or technological domain slices that are created according to their local templates. The process of slice creation can be recursive. The slice level are split between slice operator and slice tenant. The slice tenant obtains information about slices related KPIs and is expressing his reconfiguration wills as intents (high level policies), which are implemented in an automated manner by slice control and management planes. The slice operator is responsible for slice lifecycle and slice FCAPS handling. During operations of slice the slice resources are allocated in a dynamic way in order to provide required performance but in an economical way.

Each network slice exhibits following features: protection (note 2), elasticity (note 3), extensibility (note 4) and safety (note 5).

III. Network Slice Operations Problems (NSO)

\* Problem NSO1: Slice life cycle management including creation,

activation / deactivation, protection (note 2), elasticity (note 3),

extensibility (note 4), safety (note 5), sizing and scalability of the

slicing model per network and per network cloud: slices in access, core

and transport networks; slices in data centres, slices in edge clouds.

**Alex**

Short explanation: Network slicing enables the operator to create logically partitioned networks at a given time customized to provide optimized services for different market scenarios. These scenarios demand diverse requirements in terms of service characteristics, required customized network and virtual network functionality (at the data, control, management planes), required network resources, performance, isolation, elasticity and QoS issues. A network slice is created only with the necessary network functions and network resources at a given time. They are gathered from a complete set of resources and network /virtual network functions and orchestrated for the particular services and purposes.

New protocols are needed for realising full Slice life cycle management at two distinct levels:

(1) “network slice life-cycle management level” (i.e. the series of state of functional activities through which a network slice passes: creation, operation, deletion) and

(2) “network slice instances level” (activated network slice level) as shown in next figure.

Functions for creating and managing network slice instances and the functions instantiated in the network slice instance are mapped to respective framework level.

\* Problem NSO2: Autonomic slice management and operation: namely self-configuration, self-composition, self-monitoring, self-optimisation,

self-elasticity are carried as part of the slice protocols.

**Slawomir**

**TODO approx 0.5 page: Description & new/revised**

**protocols needed (Links with UCs & framework documents)**

\* Problem NSO3 - Slice stitching / composition: having enablers and methods for efficient stitching /composition/ decomposition of slices:

- vertically (service + management + control planes) and/or

- horizontally (between different domains part of access, core, edge segments) and /or

- vertically + horizontally.

**Slawomir**

Short explanation: Slice stitching. The network slice has to provide end-to-end communication and services. In some cases such end-to-end network slice instance can be created directly but in multi-domain environment the end-to slice will be a composition of slices of different domains (access, core, transport, etc.). In such a case the domain slices will be created and maintained using domain specific slice templates and use domain specific operations and all the domain slicing will be stitched together horizontally. The operation is supported by appropriate descriptions of domain slices, exchange of slice related policies between domains. Slice stitching operations are supported by uniform slice descriptors and appropriate matching of them. Each slices has appropriate set of mechanisms (slice border control functions) that support horizontal stitching of slices.

The vertical stitching of slices is an operation that modifies functionality of existing slice by adding and merging of functions of another slice (i.e. enhancing control plane properties be functions defined in another slice template). In general the vertical stitching of slices is used to enrich slice services.

* Problem NSO4- End-to-end network segments and network clouds orchestration

of slices ([GUERZONI-2016], [KARL-2016]).

**Pedro**

Short explanation: Network service composition has demonstrated to be highly beneficial for both operators and final users [GRAMMATIKOU-2012]. It allows the formation of large number of different services, which will be specialized to the particular needs of a user or a specific situation. However, the current network architecture is far for being ideal to implement such function.

One of the keys of network slicing is the flexibility it adds to the network and the resulting “de-ossification” of network resources. Thus, this environment is much more optimal to allow the proliferation of network service composition, but it means that some sort of specific requirements must be pushed towards the architecture that supports the general slicing.

First, a proper composable network service model needs network resources to be compatible, regardless of the domain to which they pertain. Then they must be homogeneously described, so a user can actually understand their individual capabilities and “draw” the service they want to build by combining them. Finally, the resources living among separated network slices must be “connectable” to each other. This means that they must cross the domain of their providers/owners in order to reach their destination.

\* Problem NSO5: End-to-end network segments and network clouds orchestration

of slices ([GUERZONI-2016], [KARL-2016]).

**Luis – TODO - approx 0.5 page: Description & new/revised protocols needed**

**(Links with UCs & framework documents)**

\* Problem NSO6: Service Mapping: having dynamic and Automatic Mapping of Services to slices; YANG models for slices.

**Susan / Satoru – TODO - approx 0.5 page: Description & new/revised protocols needed (Links with UCs & framework documents)**

\* Problem NSO7: Describe the enablers and methods for the above mentioned

capabilities and operations from different viewpoints on slices (note 6).

**Pedro**

Short explanation: The viewpoints that emerge from the global and wide interaction among the Network Slice Provider (NSP), Network Slice Operator (NSO), and Application Service Provider (ASP) must be treated by network slicing to ensure their use cases are correctly covered. They are:

NSO <=> ASP:

– ASP requests to CRUD on Network Slices.

NSP <=> NSO:

– Allows the NSO to manage the “slice” of network resources from a provider.

+ Vertical interaction to request and instantiate (embed) composite network services onto the underlying physical infrastructures.

– Possibly recursive when a NSO also acts as NSP.

NSO <=> NSO:

– Allows NSOs to coordinate:

+ Inter-operator tasks (e.g. resource migration) requested by ASPs.

+ Interconnection and interoperability among NSs of different operators.

– Horizontal (non-recursive) communication between NSOs.

NSP <=> NSP:

– Horizontal communication between providers to coordinate the interaction among physical infrastructure resources, and/or the migration of slices among NSPs.

\* Problem NSO8: Efficient enablers and methods for integration of above capabilities and operations.

**Carlos**

Short explanation: In order to enable the above required capabilities and operation for network slicing, well defined reference points among the involved actors and entities are required, as well as the proper interface definitions ensuring interoperability of all involved pieces. Some examples of the required reference points/interfaces include:

* Customer/Vertical (user of the slice) – Network Slice Provider. The user of the slide SHOULD be able to specify the characteristics of the slide and provide it in a suitable/understandable format to the NSP. A proper information model might be needed to convey the customer slice requirements. And the model might need to support different levels of abstraction, to support different use cases.
* Network Slice Provider – Network Slice Provider / Network Slice Operator / Network Service Provider. The slice provider MUST be able to request resources to compose a slice to other slice providers, slice operator or service operators. The interface needs to support recursiveness and different levels of abstraction (the request might involve resources or services).
* Inter-domain interactions at different levels. Another way of composing a slice is by interaction of players at the same level (peering, instead of recursive), by delegating the request to other provides/operators. This type of interaction can take place at different levels (resource, network service, etc), and therefore would impose different requirements. In all cases, security issues are key due to the inter-operator nature.

## 2.1. Notes

(2) Protection refers to the related mechanisms so that events within

one slice, such as congestion, do not have a negative impact on

another slice.

(3) Elasticity refers to the mechanisms and triggers for the growth

/shrinkage of network resources, and/or network and service functions.

(4) Extensibility refers to the ability to expand a NS with

additional functionality and/or characteristics, or through the

modification of existing functionality/characteristics, while

minimizing impact to existing functions.

(5) Safety refers to the conditions of being protected against

different types and the consequences of failure, error harm or any

other event, which could be considered non-desirable.

(6) Multiple viewpoints on slices: I) viewpoint of the slice's owner

towards user: from this viewpoint a slice is defined as a means to

"split" physical or virtual infrastructure elements to "service" smaller

portions. This action would be recursively done from the owner of the

initial and physical infrastructure element to the users. II) viewpoint of

from the user towards the physical infrastructure owner. From this viewpoint

a slice is viewed just as a set of resources that must be managed (requests

to a provider, listed, changed, returned to the provider, etc.). This

viewpoint emphasizes those issues that would be used in the SLA definition

of a slice.

4. OAM Operation with Customized Granularity

Med & Christian

In accordance with [RFC6291], OAM is used to denote the following:

o Operations: refer to activities that are undertaken to keep the

network and the services it deliver up and running. It includes

monitoring the underlying resources and identifying problems.

o Administration: refer to activities to keep track of resources

within the network and how they are used.

o Maintenance: refer to activities to facilitate repairs and

upgrades. Maintenance also involves corrective and preventive

measures to make the managed network run more effectively, e.g.,

adjusting configuration and parameters.

As per [RFC6291], NetSlices provisioning operations are not

considered as part of OAM. Provisioning operations are discussed in

other sections.

Maintaining automatically-provisioned slices within a network raises

the following requirements:

o Ability to run OAM activities on a provider's customized

granularity level. In other words, ability to run OAM activities

at any level of granularity that a service provider see fit. In

particular:

\* An operator must be able to execute OAM tasks on a per slice

basis.

\* These tasks can cover the "whole" slice within a domain or a

portion of that slice (for troubleshooting purposes, for

example).

\* For example, OAM tasks can consist in tracing resources that

are bound to a given slice, tracing resources that are invoked

when forwarding a given flow bound to a given network slice,

assessing whether flow isolation characteristics are in

conformance with the NS Resource Specification, or assessing

the compliance of the allocated slice resource against flow/

customer requirements.

\* An operator must be able to enable differentiated failure

detect and repair features for a specific/subset of network

slices. For example, a given slice may require fast detect and

repair mechanisms (e.g., as a function of the nature of the

traffic (pattern) forwarded through the NS), while others may

not be engineered with such means.

\* When a given slice is shared among multiple services/customers,

an operator must be able to execute (per-slice) OAM tasks for a

particular service or customer.

o Ability to automatically discover the underlying service functions

and the slices they are involved in or they belong to.

o Ability to dynamically discover the set of NetSlices that are

enabled within a network. Such dynamic discovery capability

facilitates the detection of any mismatch between the view

maintained by the control plane and the actual network

configuration. When mismatches are detected, corrective actions

must be undertaken accordingly.

# 5. Security Considerations

Security will be a major part of the design of network slicing.

# 6. IANA Considerations

This document requests no IANA actions.

# 7. Acknowledgements

…………………..

# 8. References

## 8.1. IETF References

[I-D.dong-network-slicing-problem-statement]

Dong, J. and S. Bryant, "Problem Statement of Network

Slicing in IP/MPLS Networks", draft-dong-network-slicing-

problem-statement-00 (work in progress), October 2016.

[I-D.galis-anima-autonomic-slice-networking]

Galis, A., Makhijani, K., and D. Yu, "Autonomic Slice

Networking-Requirements and Reference Model", draft-galis-

anima-autonomic-slice-networking-01 (work in progress),

October 2016.

[RFC7665]

Halpern, J., Pignataro, C.,

"Service Function Chaining (SFC) Architecture",

https://tools.ietf.org/html/rfc7665, October 2015.

[I-D.leeking-actn-problem-statement 03]

Ceccarelli, D., Lee, Y.,

"Framework for Abstraction and Control of Traffic Engineered

Networks", draft-leeking-actn-problem-statement-03

(work in progress), September 2014.

[I-D.teas-actn-requirements-04]

Lee, Y., Dhody, D., Belotti, S., Pithewan, K., Ceccarelli, D.,

"Requirements for Abstraction and Control of TE Networks",

draft-ietf-teas-actn-requirements-04.txt

January 2017.

[IETF-Slicing1]

"Presentations - Network Slicing meeting at IETF 97 of

15th November 2016", n.d., <https://www.dropbox.com/s/ax2o

fdwygjema8z/0-Network%20Slicing%20Side%20Meeting%20Introdu

ction\_IETF97.pdf>.

[IETF-Slicing2]

"Presentations - Network Slicing meeting at IETF 97 of

15th November 2016", n.d., <https://www.dropbox.com/s/k2or

6sd0ddzrc6c/1-Network%20Slicing%20Problem%20Statement\_IETF

97.pdf>.

[IETF-Slicing3]

"Presentations - Network Slicing meeting at IETF 97 of

15th November 2016", n.d., <https://www.dropbox.com/s/g8zv

fvbrtkysjs1/2-Autonomic%20Slice%20Networking\_IETF97.pdf>.

[IETF-Slicing4]

"Presentations - Network Slicing meeting at IETF 97 of

15th November 2016", n.d., <https://www.dropbox.com/s/d3rk

4pjeg552ilv/3-Architecture%20for%20delivering%20multicast%

20mobility%20services%20using%20network%20slicing\_IETF97.p

df>.

[IETF-Slicing5]

"Presentations - Network Slicing meeting at IETF 97 of

15th November 2016", n.d., <https://www.dropbox.com/s/e3is

n1bxwwhaw8g/4-ACTN%20and%20network%20slicing\_IETF97.pdf>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate

Requirement Levels", BCP 14, RFC 2119,

DOI 10.17487/RFC2119, March 1997,

<http://www.rfc-editor.org/info/rfc2119>.

[RFC4364] Rosen, E. and Y. Rekhter, "BGP/MPLS IP Virtual Private

Networks (VPNs)", RFC 4364, DOI 10.17487/RFC4364, February

2006, <http://www.rfc-editor.org/info/rfc4364>.

[RFC1958] Carpenter, B., "Architectural Principles of the Internet",

RFC 1958,

<https://www.ietf.org/rfc/rfc1958.txt>.

[RFC3439] Bush, R., Meyer, D., "Some Internet Architectural Guidelines and

Philosophy", RFC 3439,

<https://www.ietf.org/rfc/rfc3439.txt>.

[RFC3234] Carpenter, B., Brim S., "Middleboxes: Taxonomy and Issues",

RFC3439,

<https://tools.ietf.org/html/rfc3234>.

[RFC7149] Boucadair, M., Jacquenet, C. , " Software-Defined Networking: A

Perspective from within a Service Provider Environment", RFC 7149,

March 2014

<https://tools.ietf.org/html/rfc7149>.

[SFG WG]

"Service Function Chaining WG"

<https://datatracker.ietf.org/doc/charter-ietf-sfc/>.

[CPP]

Boucadair M., Jacquenet, C., Wang, N., "IP Connectivity

Provisioning Profile (CPP)"

<https://tools.ietf.org/html/rfc7297>

[IETF-Mobility]Truong-Xuan Do, Young-Han Kim, "Architecture

for delivering multicast mobility services using network

slicing" 2016-10-31

<draft-xuan-dmm-multicast-mobility-slicing-00.txt>

[IETF-Virtualization] Carlos Bernardos, Akbar Rahman, Juan Zuniga, Luis Contreras,

Pedro Aranda, " Network Virtualization Research Challenges" 2016-10-31

<draft-irtf-nfvrg-gaps-network-virtualization-03.txt>

[IETF-Coding] M.A. Vazquez-Castro, Tan Do-Duy, Paresh Saxena, Magnus Vikstrom,

"Network Coding Function Virtualization" 2016-11-14

<draft-vazquez-nfvrg-netcod-function-virtualization-00.txt>

[IETF-Anchoring] Anthony Chan, Xinpeng Wei, Jong-Hyouk Lee, Seil Jeon,

Alexandre Petrescu, Fred Templin "Distributed Mobility Anchoring"

2016-12-15

<draft-ietf-dmm-distributed-mobility-anchoring-03.txt,.pdf>

[RFC6291] L. Andersson, H. van Helvoort, R. Bonica, D. Romascanu,

S. Mansfield “Guidelines for the Use of the "OAM" Acronym

in the IETF” – June 2011

<https://tools.ietf.org/html/rfc6291>

,

## 8.2. Informative References

[ChinaCom-2009]

"A. Galis et al - Management and Service-aware Networking

Architectures (MANA) for Future Internet - Invited paper

IEEE 2009 Fourth International Conference on

Communications and Networking in China (ChinaCom09) 26-28

August 2009, Xi'an, China", n.d.,

<http://www.chinacom.org/2009/index.html>.

[GENI-2009]

"GENI Key Concepts - Global Environment for Network

Innovations (GENI)", n.d.,

<http://groups.geni.net/geni/wiki/GENIConcepts>.

[GUERZONI-2016]

"Guerzoni, R., Vaishnavi, I., Perez-Caparros, D., Galis, A.,

et al Analysis of End-to-End Multi Domain Management and

Orchestration Frameworks for Software Defined

Infrastructures - an Architectural Survey", June 2016,

<onlinelibrary.eiley.com/10.1002/ett.3084/pdf>.

[IMT2020-2015]

"Report on Gap Analysis", ITU-T FG IMT2020, December

2015, <http://www.itu.int/en/ITU-T/focusgroups/imt-

2020/Pages/default.aspx>.

[IMT2020-2016]

"Draft Technical Report Application of network

softwarization to IMT-2020 (O-041)", ITU-T FG IMT2020,

December 2016, <http://www.itu.int/en/ITU-T/focusgroups/

imt-2020/Pages/default.aspx>.

[IMT2020-2016bis]

"Draft Terms and definitions for IMT-2020 in ITU-T

(O-040)", ITU-T FG IMT2020, December 2016,

<http://www.itu.int/en/ITU-T/focusgroups/imt-2020/Pages/

default.aspx>.

[KARL-2016]

"Karl, H., Peuster, M, Galis, A., et al DevOps for Network

Function Virtualization - An Architectural Approach", July

2016,

<http://onlinelibrary.wiley.com/doi/10.1002/ett.3084/full>.

[MANO-2014]

"Network Functions Virtualisation (NFV); Management and

Orchestration v1.1.1.", ETSI European Telecommunications

Standards Institute., December 2014,

<http://www.etsi.org/deliver/etsi\_gs/NFV-

MAN/001\_099/001/01.01.01\_60/gs\_nfv-man001v010101p.pdf>.

[NGMN-2016]

"Hedmar,P., Mschner, K., et al - Description of Network

Slicing Concept", NGMN Alliance NGS-3GPP-2016, January

2016, <https://www.ngmn.org/uploads/

media/160113\_Network\_Slicing\_v1\_0.pdf>.

[NGS-3GPP-2016]

"Study on Architecture for Next Generation System - latest

version v1.0.2", September 2016,

<http://www.3gpp.org/ftp/tsg\_sa/WG2\_Arch/Latest\_SA2\_Specs/

Latest\_draft\_S2\_Specs>.

[ONF-2016]

"Paul, M, Schallen, S., Betts, M., Hood, D., Shirazipor,

M., Lopes, D., Kaippallimalit, J., - Open Network

Fundation document "Applying SDN Architecture to 5G

Slicing", Open Network Fundation, April 2016,

<https://www.opennetworking.org/images/stories/downloads/

sdn-resources/technical-reports/

Applying\_SDN\_Architecture\_to\_5G\_Slicing\_TR-526.pdf>.

[5G-ICN]

Ravi Ravindran, Asit Chakraborti, Syed Obaid Amin, Aytac Azgin,

G.Q.Wang, “5G-ICN: Delivering ICN Services in 5G using Network

Slicing”, IEEE Communication Magazine, May, 2017.

[GRAMMATIKOU-2012]

Grammatikou, M; Marinos, C; Martinez-Julia, P; Jofre, J;

Gheorghiu, S; et al. Proceedings of the International Conference on

Parallel and Distributed Processing Techniques and Applications (PDPTA); Athens: 1-5. Athens: The Steering Committee of The World

Congress in Computer Science, Computer Engineering and Applied

Computing (WorldComp). (2012)

[GAPS]

Gap Analysis for Network Slicing

draft-qiang-netslices-gap-analysis-00

[NS UseCases]

Network Slicing Use Cases: Network Customization for

different services

draft-makhijani-netslices-usecase-customization-02

[NS ARCH]

Network Slicing Architecture

draft-geng-netslices-architecture-01

# Authors' Addresses

Alex, Kiran, Stewart, Jimmy, Liang, Ravi, Pedro, Med, Christian, Slawomir, Luis, Susan, Satoru, Carlos,