Network Working Group
Internet-Draft
Intended status: Informational
Expires: December 4, 2017

L. Geng
China Mobile
J. Dong
S. Bryant
K. Makhijani
Huawei Technologies
A. Galis
University College London
X. de Foy
InterDigital Inc.
S. Kuklinsk
Orange
June 2, 2017

Network Slicing Architecture
draft-geng-netslices-architecture-01

Abstract

This document defines the overall architecture of network slicing. Based on the general architecture, basic concepts of network slicing and examples of network slicing instances are introduced for clarification purposes. Some architectural considerations about the data plane, control plane, management and orchestration of network slicing are described to give a general view of network slicing implementation principles.

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 December 4, 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 3
1.1. Requirements Language 4
1.2. Terminology 4
2. Demand for Network Slicing 7
2.1. Guaranteed Service Performance 7
2.2. End-to-end Customization 8
2.3. Network Slicing as a Service 8
3. Network Slicing Architecture 8
3.1. Requirements 8
3.2. High-Level Functional Components 9
3.2.1. Service Component 11
3.2.2. Network Slicing Management and Orchestration 11
3.2.3. Resource Component 14
3.3. Network Slicing Capabilities 15
3.3.1. Reclusiveness 15
3.3.2. Protection 15
3.3.3. Elasticity 15
3.3.4. Extensibility 16
3.3.5. Safety 16
3.3.6. Isolation 16
3.4. Network Slices Capability Exposure 16
4. Data Plane of Network Slicing 17
4.1. Propagation of Guarantees 17
4.2. The Underlying Physical Layer 17
4.3. Hard vs Soft Slicing in the Data-plane 18
4.4. The Role of Deterministic Networking 18
4.5. The Role of VPNs 19
4.6. Dynamic Reprovisioning 19
4.7. Non-IP Data Plane 19
5. Control Plane of Network Slicing 19
5.1. NS Infrastructure Control Plane 20

5.2. NS Infrastructure Control Operations and Protocols . . . 20
5.3. Programmability of the NS Infrastructure Control Plane . 21
5.4. Intra-Slice Control Plane 21
6. Management Plane of Network Slicing 22
6.1. Network Slice Creation - Reservation / Release Messages Flow 22
6.2. Self- Management Operations 23
6.3. Programmability of the Management Plane 24
6.4. Management plane slicing protocols 24
7. Service Functions and Mappings 24
8. OAM and Telemetry 24
9. IANA Considerations 25
10. Security Considerations 25
11. Acknowledgements 25
12. References 25
12.1. Normative References 25
12.2. Informative References 26
Authors' Addresses 26

1. Introduction

The Internet has always been designed to support a variety of services. The emerging 5G market is expected to bring this diversity of services to a new level [NS_WP]. Typical examples of new bandwidth-hungry services enabled by 5G include high definition (HD) video, virtual reality (VR) and augmented reality (AR). The high bandwidth requirement of these services is not particularly challenging thanks to the continuing advancing technologies. However, the guarantee of high bandwidth performance of these services based-on a spontaneous on-demand pattern is fairly challenging. Moreover, providing high bandwidth with strict packet loss tolerances and high mobility is also difficult for the current networks which are commonly designed for best effort purposes.

Given that most Internet protocols are designed to comply with a best effort, or enhanced best effort paradigm, it is inevitable that the network will suffer from performance degradation in case of congestion. Recent work on deterministic networking (DetNet) [I-D.finn-detnet-architecture] aims to improve this situation by providing a ceiling on latency for a particular traffic flow, which significant improves packet error rate for specific DetNet services. This pioneering work gives a great example that new approaches are investigated to make the Internet aware of certain performance requirement other than the bandwidth.

Taking a look at the network infrastructure, service provider used to build dedicated network and resources for services requiring guaranteed performance. This is simply not cost-effective, neither

is it flexible. The emergence of virtualization and VPN technologies make it possible to set up logically isolated computing and network instances from shared infrastructures. This can be used dedicatedly by specific services for improved performances. However, many questions are still to be answered as different technologies in various domains need to be combined to build network slices, which may require the separation of different resources and various types of performance guarantees.

1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119.

1.2. Terminology

I. Networking Servicing Terms

Service - A piece of software that performs one or more functions and provides one or more APIs to applications or other services of the same or different layers to make use of said functions and returns one or more results. Services can be combined with other services, or called in a certain serialized manner, to create a new service.

Service Instance - An instance of an end-user service or a business service that is realized within or by a network slice. Each service is represented by a service instance. Services and service instances would be provided by the network operator or by third parties.

Administrative domain - A collection of systems and networks operated by a single organization or administrative authority. Infrastructure domain is an administrative domain that provides virtualized infrastructure resources such as compute, network, and storage, or a composition of those resources via a service abstraction to another Administrative Domain, and is responsible for the management and orchestration of those resources.

II. Network Resource Terms

Resource - A physical or virtual (network, compute, storage) component available within a system. Resources can be very simple or fine-grained (e.g., a port or a queue) or complex, comprised of multiple resources (e.g., a network device).

Logical Resource - 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.

Virtual Resource - 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.

Network Function (NF) - A processing function in a network. It includes but is not limited to network nodes functionality, e.g. session management, mobility management, switching, routing functions, which has defined functional behaviour and interfaces. Network functions can be implemented as a network node on a dedicated hardware or as a virtualized software functions. Data, Control, Management, Orchestration planes functions are Network Functions.

Virtual Network Function (VNF) - A network function whose functional software is decoupled from hardware. One or more virtual machines running different software and processes on top of industry-standard high-volume servers, switches and storage, or cloud computing infrastructure, and capable of implementing network functions traditionally implemented via custom hardware appliances and middle-boxes (e.g. router, NAT, firewall, load balancer, etc.)

Network Element - A network element is defined as a manageable logical entity uniting one or more network devices. This allows distributed devices to be managed in a unified way using one management system. It means also a facility or equipment used in the provision of a communication service. Such term also includes features, functions, and capabilities that are provided by means of such facility or equipment, including subscriber numbers, databases, signalling systems, and information sufficient for billing and collection or used in the transmission, routing, or other provision of a telecommunications service.

III. Network Slicing Terms used in this draft

Resource Slice - A grouping of physical or virtual (network, compute, storage) resources. It inherits the characteristics of the resources which are also bound to the capability of the resource. A resource slice could be one of the components of Network Slice, however on its own does not represent fully a Network Slice.

Network Slice - 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).

Network Slice Instance - An activated network slice. It is created based on network template. A set of managed run-time network functions, and resources to run these network functions, forming a complete instantiated logical network to meet certain network characteristics required by the service instance(s). It provides the network characteristics that are required by a service instance. A network slice instance may also be shared across multiple service instances provided by the network operator. The network slice instance may be composed by none, one or more sub-network instances, which may be shared by another network slice instance.

Network Slice Template - A complete description of the structure, configuration and the plans/work flows for how to instantiate and control the Network Slice Instance during its life cycle.

Network Slice Type - Network slices are categorized into different types according to the abstraction of characteristics of the services they facilitate. The methodology used for defining network slice types may be different for the network slice provider. Some typical examples of network slice types according to 5G implementation include eMMB, mMTC and URLLC. Network slice type may be used to map specific network resources, VPNs, QoS categories according to real implementation. It is advised that mutual types should be defined according to existing main-stream service implementation scenarios. Extensions should be allowed for network slicing service provider to make according to new requirements.

Network Slice Provider - A network slicing provider, typically a telecommunication service provider, is the owner or tenant of the network infrastructures from which network slices are created. The network slicing provider takes the responsibilities of managing and orchestrating corresponding resources that the network slicing consists of.

Network Slice Terminal - A terminal that is network-slice-aware, typically subscribed to the service which is hosted within a network slice instance. A network slice terminal may be capable of subscribing to multiple network slice instance simultaneously.

Network Slice Tenant - A network slice tenant is the user of specific NSIs, with which specific services can be provided to end customers. Network slice tenants can make requests of the creation of new network slice instances. Certain level of management capability should be exposed to network slice tenant from network slice service provider.

Network Slice Repository - A repository that in each domain consists of a list of active Network Slices with their identifiers and

description. This description defines also the rules that have to be fulfilled in order to access a slice. Network Slice Repository is updated by slice orchestrator. In case of recursive slicing the Network Slice Repository keeps information about all slices that compose a higher level slice but such slice has its own identifier and descriptors.

Slice Border Control - A functional entity that is used for users to slice attachment and in recursive slicing, in which an end-to-end slice is a horizontal combination of per domain slices. It's role is to expose information about a slice to other slices in order to provide efficient slice connection (i.e. topology information exchange, etc.) and to perform necessary protocol translations.

Slice Selection Function - A functional entity that is used by the end-users in order to attach to a slice. It provides mechanisms related to slice advertisement, on request passes information related to slice attachment to the end-users and optionally authenticate users. In case of lack of an active slice as described in user request it may provide slice matching and also trigger the creation of a slice on-demand.

2. Demand for Network Slicing

It is expected that a diversity of new services will emerge in both mobile/5G and fixed networks.[I-D.qin-netslices-use-cases] describes many of the differentiated services (e.g. smart home, industrial control, remote healthcare, Vehicle-to-Everything (V2X) etc.) and their relevance to the Network Slicing. These use cases are typical examples of service verticals requiring features beyond connectivity such as uRLL, high-bandwidth, and isolation.

2.1. Guaranteed Service Performance

One of the most challenging requirements for future network is to provide guaranteed performance for varieties of new services whilst maintaining the economies of scale that accrue through resource sharing. It has been foreseen that the requirements of different services would be diversified and complex.

Network slicing can deal with these challenges by mapping the performance requirements to physically or logically dedicated resources.

2.2. End-to-end Customization

Customization is another significant feature of future services. Many vertical industries are expected to offer customization capabilities as a service to both internal manufacturing processes and specific end users. Meanwhile, these customized services need to be deployed with short time-to-market. The network needs to adapt to this challenge since customers may frequently adjust and refine their customization requirements.

There is ongoing work such as network orchestration, software defined networks and network function virtualization that aims to address this problem. In principle, these new technologies share a common request for the network to provide the ability to provide agile resource allocation.

2.3. Network Slicing as a Service

It is anticipated that the operation of 5G and future networks will involve new business models. Given that the network is more flexible, elastic, modularized and customized, the shared network infrastructure can be sliced and offered as a service to the customer. For instance, dedicated, isolated, end-to-end network resources with a customized topology can be provided as a network slice service to the tenant of this network slice.The tenants are allowed to have a certain level of provisioning of their network slices.

3. Network Slicing Architecture

This section introduces the general system architecture of network slicing.

3.1. Requirements

To meet the diversified Quality of Experience (QoE) demands of different vertical industries, the gap analysis document has identified the following requirements:

- Req.1 Network Slicing Resource Specification
- Req.2 Cross-Network Segment; Cross-Domain Negotiation
- Req.3 Guaranteed Slice Performance and Isolation
- Req.4 Slice Discovery and Identification
- Req.5 NS Domain-Abstraction

- Req.6 OAM Operations with Customized Granularity

In the following sections, these requirements will be addressed and associated with different aspects of the Network Slicing architecture.

3.2. High-Level Functional Components

End-to-end network slice is a broad area and comprises of several functional components. In the context of distribution of role and responsibilities, a network slice consists of the following components as shown in Figure 1. It can be seen that two network slice instances are created from the shared network infrastructures. In principle, the network slicing subnets (NS Subnets) represent any general physical and logical network resources for demonstration purposes. The two network slice instances created share the computing, connectivity and storage resources, whether they are in physical or virtual forms.

It is fundamental to network slicing that slices may be created, the topology and/or its resources modified, and that the slices may be decommissioned in a timely manner with minimum work by the network slicing provider or the customer. This is not however unique to network slicing, it is a goal of modern classical networks to be able to do this.

The descriptions of functional components are introduced in the following sections.

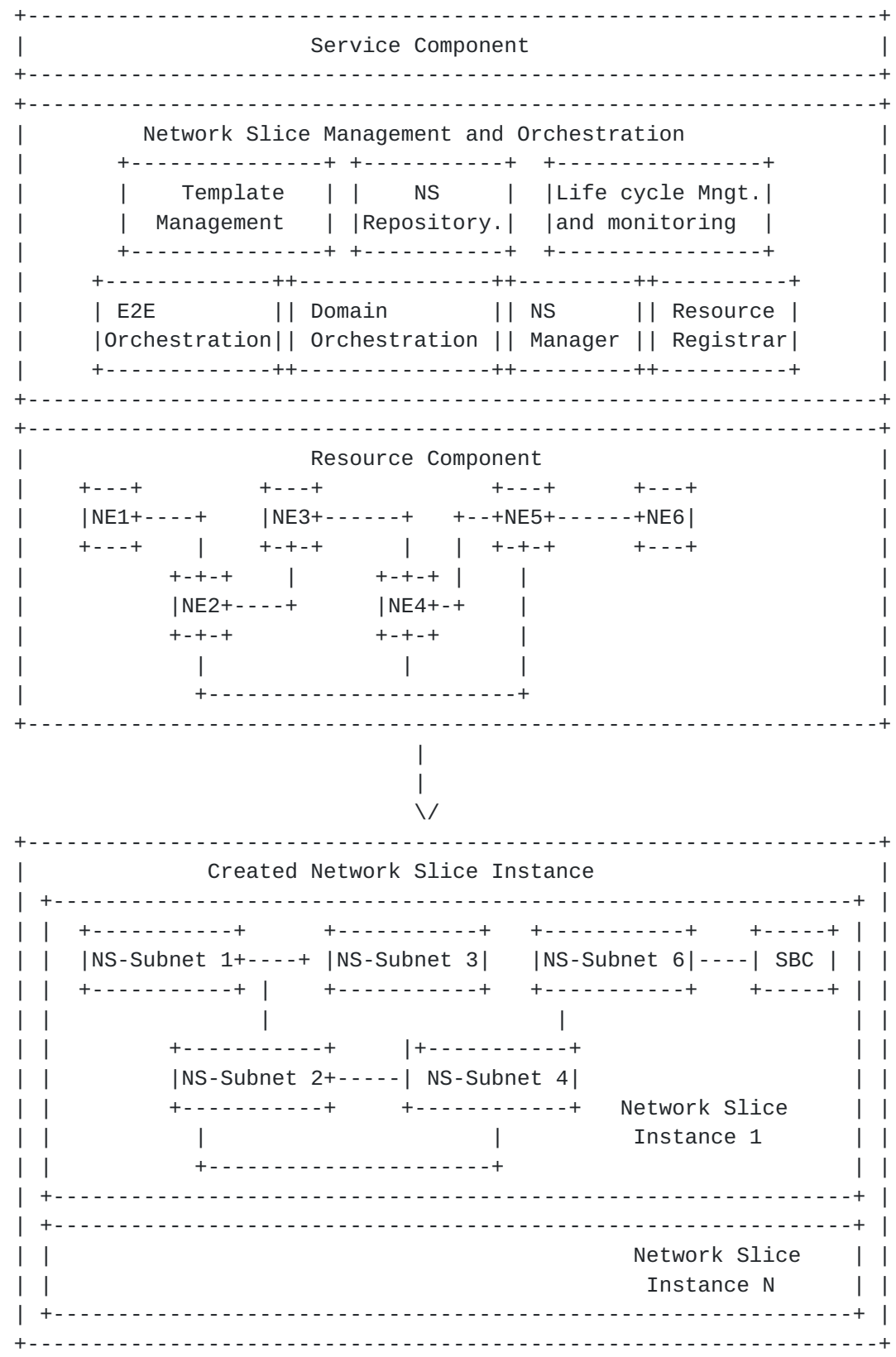


Figure 1: Network Slicing Architecture

3.2.1. Service Component

A service represents an end-user's business logic. It is realized within or by Network slice instance. A service may demand a set of network resources and attributes in form of a network slice. A service is either mapped to a network slice instance or an ordered chain of network slice instances.

3.2.2. Network Slicing Management and Orchestration

As seen in Figure 1, The management and orchestration layer of network slicing system consist of the following functional components.

1. NS Template Management

2.NS Repository

To provide mechanism that will allow the end-user selection and attachment to a slice instance or, if required, to multiple slice instances at the same time, a NS repository (or repositories) is needed, in which there are stored slices with the description of their properties and access rules.

The service component should have an access to such repository in order to check if the required slice exists. If such a slice doesn't exist a matching procedure should allow an attachment of the service to a slice which properties are the most similar ones to the requested slice (under certain policies agreement between network slice provider and tenant). Optionally the service may trigger the deployment of a new slice. During the attachment of the service component to a slice the slice data forwarding mechanisms are configured in way that will redirect a selected part of the end-user traffic to the slice.

3.Lifecycle management and monitoring

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.

A network slice is a dynamic entity therefore its lifecycle has to be managed. The network slice lifecycle management is (creation, update, deletion) is managed by the network slice orchestrator. The slice orchestrator according to requests that can be send by the orchestrator operator, 3rd parties or even by the end-users creates a new slice instance that is based on slice template that is stored is slice template repository however it takes into account slice operator (owner) preferences (policies).

4.E2E Orchestration

This section describes E2E Slices Orchestration and its functionality. Orchestration refers to the system functions in a domain that automate and autonomically co-ordination of network functions in slices autonomically coordinate the slices lifecycle and all the components that are part of the slice (i.e. Service Instances, Network Slice Instances, Resources, Capabilities exposure) to ensure an optimized allocation of the necessary resources across the network. The main functionality of E2E slice orchestration may include the following aspects.

(1) Coordinate a number of interrelated resources, often distributed across a number of subordinate domains, and to assure transactional integrity as part of the process.

(2) Autonomically control of slice life cycle management, including concatenation of slices in each segment of the infrastructure including the data pane, the control plane, and the management plane.

(3) Autonomically coordinate and trigger of slice elasticity and placement of logical resources in slices.

(4) Coordinates and (re)-configure logical resources in the slice by taking over the control of all the virtualized network functions assigned to the slice.

It is the continuous process of allocating resources to satisfy contending demands in an optimal manner. The idea of optimization would include at least prioritized SLA commitments , and factors such as customer endpoint location, geographic or topological proximity, delay, aggregate or fine-grained load, monetary cost, fate- sharing or affinity. The word continuing incorporates recognition that the environment and the service demands constantly change over the course of time, so that orchestration is a continuous, multi-dimensional optimization feedback loop. The E2E slice orchestration should have the following characteristics.

- It protects the infrastructure from instabilities and side effects due to the presence of many slice components running in parallel.

- It ensures the proper triggering sequence of slice functionality and their stable operation.

- It defines conditions/constraints under which service components will be activated, taking into account operator service and network requirements (inclusive of optimize the use of the available network; compute resources and avoid situations that can lead to sub-par performance and even unstable and oscillatory behaviors.

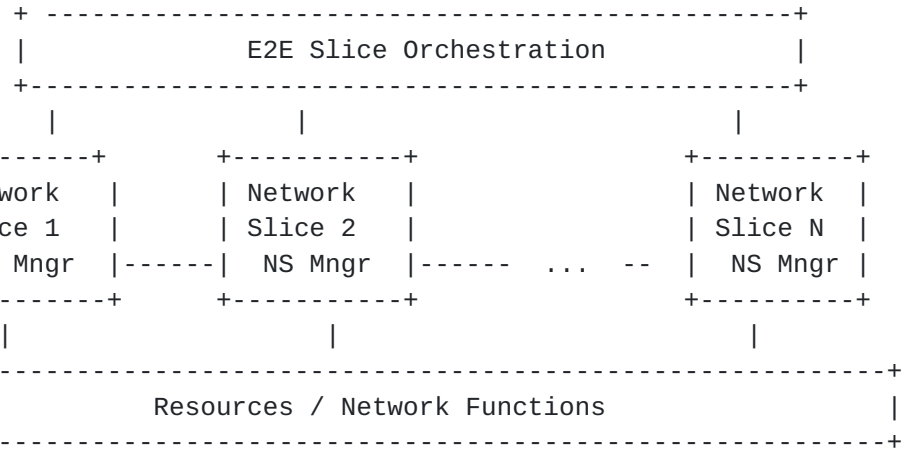


Figure 2: E2E Slice Orchestration

5. Domain Orchestration

Another value that the network slicing brings is fast, automated and dynamic deployment services in end-to-end manner, even in heterogeneous environment. In order to achieve that goal the problem of providing a slice that spans multiple domains has to be solved. There two possible solutions. The first one lies on appropriate allocation of resources in each domain (i.e. creation of the resource slice instance), their aggregation and using of a single orchestrator in order to deploy a slice. Another possibility is to use per domain orchestrators with domain specific template and provide the chaining of domain slices in order to obtain the end-to-end slice. In such a case the orchestration is hierarchical one, i.e. the domain orchestration is driven by a high level orchestrator that interacts with orchestrators of all domains that are involved in end-to-end slice instance creation. The slice that is composed of multiple domain level slices requires specific mechanisms for inter-slice operations like topology information exchange and/or appropriate protocol conversion/adaptation.

The approach may lead to recursive slicing (or sub-slicing) in which higher level slice instances are composed of lower level ones. The creation of end-to-end slice composed of several slices may require specific description of such slice and changes of functions of domain slices. For example the traffic redirection can be implemented only in this domain slice which is an ingress slice.

6. NS Manager

NS Manager mangement entity for a specific network slice instance. it manages all access permissions and all interaction between a Network Slice and external functions (i.e. other Network Slices, Orchestrators, etc). Each NS Manager maps requirements from orchestrator into network resources and manages these resources of a specific network slice instance.

Allow 3rd parties to access via APIs information regarding services provided by the slice (e.g. connectivity information, QoS, mobility, autonomicity, etc.)

Allow dynamical customization of the network characteristics for different diverse use cases within the limits set of functions by the operator. Network slice enables the operator to create networks customized to provide flexible solutions for different market scenarios, which have diverse requirements, with respect to the functionality, performance and resource separation.

It includes a description of the structure (and contained components) and configuration of the slice instance.

7. Resource Registration

Resource registration component manages the exposed capability of the network infrastructure. Details description is TBD.

3.2.3. Resource Component

Resource component includes physical, logical and virtual resources (defined in Section 2). An abstraction of resources is required in order to consistently map the requirements such as latency, reliability, band-width. Resource component may need interfaces with elements in network slice functional component as well as NS manager for that purpose of discovering capabilities.

3.3. Network Slicing Capabilities

3.3.1. Reclusiveness

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 one could summarize the concept of recursive network slice definition 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. If a certain part of a network slice can be replaced by different patterns, this can offer some advantages:

- Each pattern might have its own capabilities in terms of performance.Depending on the required workload, a network function /virtual network function might be replaced by a pattern able to process at higher performance. Similarly, a service or network function /virtual network function can be decomposed so it can be deployed on the available infrastructure.

- From an orchestrating point of view, above way of using recursive network slice templates, can be beneficial for the placement algorithm used by the orchestrator. The success rate, solution quality and/or runtime of such an embedding algorithm benefits from information on both possible scaling or decomposition topologies and available infrastructure.

- Enabling methods for network slice template segmentation allowing a slicing hierarchy with parent - child relationships.

3.3.2. Protection

Protection refers to the related capability and mechanisms so that events within one network slice, such as congestion, do not have a negative impact on another slice.

3.3.3. Elasticity

Elasticity refers to the capability, mechanisms and triggers for the growth /shrinkage of network resources, and/or network and service functions in an Network Slice as function of service needs.

3.3.4. Extensibility

Extensibility refers to the capability and ability to expand a network slice with additional functionality and/or characteristics, or through the modification of existing network function / virtual network function while minimizing impact to existing functions.

3.3.5. Safety

Safety refers to the conditions in within one network slice of being protected against different types and the consequences of failure, error harm or any other event, which could be considered non-desirable in an other network slice.

3.3.6. Isolation

Efficient slice creation is expected to guarantee the isolation and non interference between network slices in the Data /Control /Management planes as well as safety and security for multi-tenancy in slices.

3.4. Network Slices Capability Exposure

An important value of network slicing is the capability of a slice to be tightly coupled with services, i.e. the slice instance can be designed that way that it support a specific service or limited number of services only, but not all of them in the same slice. The property means that not only the slice data plane operations are properly tuned, but also the control plane can be designed according to the requirements of slice specific services. In general it is possible that a single slice instance may support a single service only, however it is more scalable to provide more than a single service per slice. Such approach has important implications. First of all in order to add services to a slice each slice should expose its functions to services/applications. Moreover the service lifecycle management is different than slice lifecycle management. This is similar to the existing networks, however in opposite to them the deployment of a new service may lead to important reconfiguration of a slice to which the service is attached (the slice is programmable what means that we are going beyond the API approach - the services templates are melted with the slice template). The goal is to have tightly coupled services with networks and providing joint optimization of networks and services at the level that is impossible to achieve in present, hardware based solutions.

4. Data Plane of Network Slicing

In the network slicing architecture, the data plane in the edge and core of the network will likely be one or more of the standard IETF data planes: IPv4/IPv6, MPLS or Pseudo-wires (PW). This section assumes that the IETF protocol stack exists as-is, and describes the performance consideration in different layers of the data plane.

4.1. Propagation of Guarantees

Guarantees of delay start at the physical layer and propagate up the stack layer by layer. Any layer can add delay, and can take various steps to minimize the impact of delay on its layer, but no layer can reduce the delay introduced by a lower layer.

Guarantees of loss and jitter can, by contrast be upheld or improved at any layer of the protocol stack, but usually at a cost of increased delay. Where delay is a constrain as it is in some 5G applications the option of trading delay for better loss or jitter characteristics is not an option. In these circumstances it is critical that the quality characteristics start at the physical layer and be maintained at each layer of the protocol stack.

4.2. The Underlying Physical Layer

A point to point dedicated physical channel provides the delay, jitter and loss characteristics limited only by the media itself. This does not fulfill the need for rapid reconfiguration of the network to provision new services.

To address the need to provision a slice of the data-plane one approach that can be deployed is to time-slice access to the physical service. Ignoring many of the classic TDM offering as being too slow, a number of technologies are available that might be applied including OTN and FlexE. Whilst the provisioning of the channel provided by underlays such as FlexE and the interconnection of FlexE channels is within the scope of this architecture the operation of the underlay is outside its scope.

The logical sub-division of a physical channel be that a single channel with the full bandwidth available or a channel multiplexed at the physical layer such as is provided by FlexE we will consider in the following section.

4.3. Hard vs Soft Slicing in the Data-plane

Hard slicing refers to the provision of resources in such a way that they are dedicated to a specific NSI. Data-plane resources are provided in the data-plane through the allocation of a lambda, through the allocation of a time domain multiplexed resource such as a FlexE channel or through a service such as an MPLS hard-pipe. Note that although hard-pipes can be used to allocate dedicated, non-shared resources to an NSI, the using of allocation is bandwidth, which can result in more "lumpiness" in the physical channel that would not be present with a true physical layer multiplexing scheme.

Soft slicing refers to the provision of resources in such a way that whilst the slices are separated such that they cannot statically interfere with each other (one cannot receive the others packets or observe or interfere with the other's storage), they can interact dynamically (one may find the other is sending a packet just when it wants to, or the other may be using CPU cycles just when the other needs to process some information), which means they may compete for some particular resource at some specific time. Soft slicing is achieved through logically multiplexing the data-plane over a physical channel include various types of tunnel (IP or MPLS) or various types of pseudo-wire (again IP or MPLS). Although the design of deterministic networking techniques helps, it is not possible to achieve the same degree of isolation with these techniques as it is possible to achieve with pure physical layer multiplexing techniques. However where such techniques provide sufficient isolation their use leads to a network design that may be deployed on existing equipment designs and which can make unused bandwidth available to best effort traffic.

4.4. The Role of Deterministic Networking

Deterministic networking is a technology under development in the IETF that aims to both minimize congestion loss and set an upper bound on per hop latency. It allows a packet layer to emulate the behaviour of a fully partitioned underlay such might be provided through some physical layer multiplexing system such as FlexE.

Deterministic networking works by policing the ingress rate of a flow to an agreed maximum and then scheduling the transmission time of each flow to reduce the "lumpiness" and hence the possible buildup of queues and hence congestion loss.

Whilst deterministic networking is not as perfect as physical layer multiplexing in terms of latency minimization, because the scheduling is hop by hop and not end to end meaning that at each hop a packet has to wait for the transmission slot allocated to its flow, it has

the advantage that it is able to allocate slots not needed by the allocated traffic to best effort traffic. This reallocation of the unused transmission slots to background traffic significantly improves the efficiency of the network by amortizing the cost between the scheduled high priority users and the best effort users.

4.5. The Role of VPNs

VPNs are considered candidate technologies for network slicing. The existing VPN technologies mainly focus on the isolation of forwarding tables between different tenants and provide a virtual topology for the connectivity between different sites of a tenant. The VPN layer and the underlying network resources are usually loosely coupled, and statistical multiplexing is adopted to improve network utilization.

Although VPNs have been widely used to provide enterprise services in service provide networks, it is unclear that whether VPNs along with existing underlying tunnel technologies can meet the performance and isolation requirements of critical services in the vertical industries.

4.6. Dynamic Reprovisioning

A requirement of the network slicing system is that it can be dynamically and non-disruptively reprovisioned. That is not an unusual requirement of a modern network. However the frequency of reprovisioning with network slicing will be relatively high, such that it in many cases it is not possible to hide any disruption during a "quiet" time.

Physical multiplexing methods such as FlexE have the ability to seamlessly reprovision multiplex slots. At the network layer techniques such as make-before-break, segment routing, and loop-free-convergence can be used to provide uninterrupted operation during a topology change.

4.7. Non-IP Data Plane

Non-IP data plane in support of Information Centric Networking (ICN), some of the IoT services and other similar requirements will be added in a future version.

5. Control Plane of Network Slicing

There are two control plane systems that need to be considered. The first is the control plane of the slicing infrastructure itself (NS Infrastructure Control Plane), the second is the control plane of an individual slice (Intra-Slice Control Plane).

5.1. NS Infrastructure Control Plane

The NS infrastructure control plane receives the instruction of creating a network slice with particular requirements from the orchestration layer. It then creates the network slice by allocating a set of network resources in the corresponding network infrastructure. This set of network resources is associated with the network slice during this operation.

The NS infrastructure control plane is also responsible, with the support of the orchestration layer, for dynamically adjusting the network according to slice change requests (e.g. from slice tenants), and to changes in network infrastructure. As it is critical to meet the service requirements of a network slice independently from activity and changes occurred in other network slices or in infrastructure, appropriate service assurance mechanisms should be deployed in the network. The control plane, with the support of the orchestration layer, MUST be able to react within a pre-determined (possibly system-specific) time to any network events, such as resource addition and failure. The orchestration layer SHOULD be involved, directly or indirectly, to take reactive decisions, e.g. to re-route a flow, to ensure that other network slices are not affected. Indirect involvement includes, for example, reactive programming by the orchestration layer to address foreseeable events or cases where connection to the orchestration layer is lost.

The NS infrastructure control plane can be implemented as an extension of the Virtual Infrastructure Manager (VIM), in cases where the NFV-MANO architecture is used for the management and control architecture of the system. Especially, the VNF Manager is considered part of the management plane and not control plane. From technology standpoint, NS infrastructure control plane can be an extension of Cloud infrastructure technology (e.g. OpenStack), which itself can integrate SDN technology for network control. This logically centralized control can be supplemented or replaced with distributed control protocols, that can provide some benefits in scenarios which require fast reaction, robustness and efficient information distribution. A hybrid architecture is anticipated, where distributed protocols complement and simplify a centralized control system.

5.2. NS Infrastructure Control Operations and Protocols

The following operations should be supported. Different control protocols can be used to control different types of resources. Multiple control protocols can be supported simultaneously.

- Setting up or tearing down network function instances within a slice. Set, increase or decrease compute capacity of NFs.
  - Control protocols can be based on openstack APIs and other Cloud infrastructure control protocols.
- Setting up, tearing down, increase or decrease capacity of connectivity between network function instances within a slice, e.g. as L2-L3 virtual network or software function chain.
  - Control protocols can include NVO3 control protocol, SFC control protocol and NetConf.
- Reservation/release of traffic flows within a slice, possibly with associated QoS and routing requirements.
  - Control protocols can include DETNET, MPLS-TE, etc.
  - Interconnect slices or slice flows, including across domains
  - Control protocols are TBD.

5.3. Programmability of the NS Infrastructure Control Plane

The NS Control Plane exposes a Northbound API, typically for use by the orchestration layer. A higher-than-physical representation level of abstraction can be used, enabling the manipulation of a logical network, that is translated down to physical resource manipulation by the NS infrastructure control plane. The level of this abstraction and of its associated logical network is TBD. Programmability should include programming reactions to events, which reduces the dynamic involvement of the orchestration layer, and therefore reaction time to events.

5.4. Intra-Slice Control Plane

Intra-slice control plane maintains proper connectivity and networking characteristics within the slice. A full range of existing control plane technologies needs to be permissible. Intra-slice control plane technologies can include existing IGP protocols (such as IS-IS or OSPF), BGP, overlay control (such as NVO3 or SFC). Some slices may be controlled by their own SDN controllers. Intra-slice control plane can span across multiple domains (since NS infrastructure control deals with slice interconnection).

6. Management Plane of Network Slicing

It is expected that the management and orchestration layer would use state of the art management technologies to support short time-to-market, and help the operators to build an open ecosystem for new services in vertical industries. In multi-tenant environment the slice tenants can trigger the creation of slice instances for them by interacting with the E2E Orchestrator. After the creation of the slice the slice tenant is able to monitor slice KPIs (performance, faults) and send slice reconfiguration requests to E2E Orchestrator.

The basic functional architecture of management and orchestration layer of network slicing system has been discussed in section 3. This section further introduces some essential characteristics.

6.1. Network Slice Creation - Reservation / Release Messages Flow

The establishment of Network slices is both business-driven (i.e. slices are in support for different types and service characteristics and business cases) and technology-driven as network slice is a grouping of physical or virtual resources (network, compute, storage) and a grouping network functions and virtual network functions (at the data, control and management planes) which can act as a sub network at a given time. A network slice can accommodate service components and network functions (physical or virtual) in all network segments: access, core and edge / enterprise networks.

The management plane creates the grouping of network resources (physical, virtual or a combination thereof), it connects with the physical and virtual network and service functions and it instantiates all of the network and service functions assigned to the slice.

Once a network slice is created, the slice control plane takes over the control, slice operations 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, ingress routers are configured so that appropriate traffic is bound to the relevant slice. Identification means for the traffic may be simple (relying on a subset of the transport coordinate, DSCP/traffic class, or flow label), or identification may be a more sophisticated one. Also, the traffic capacity that is 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 guarantee such needs.

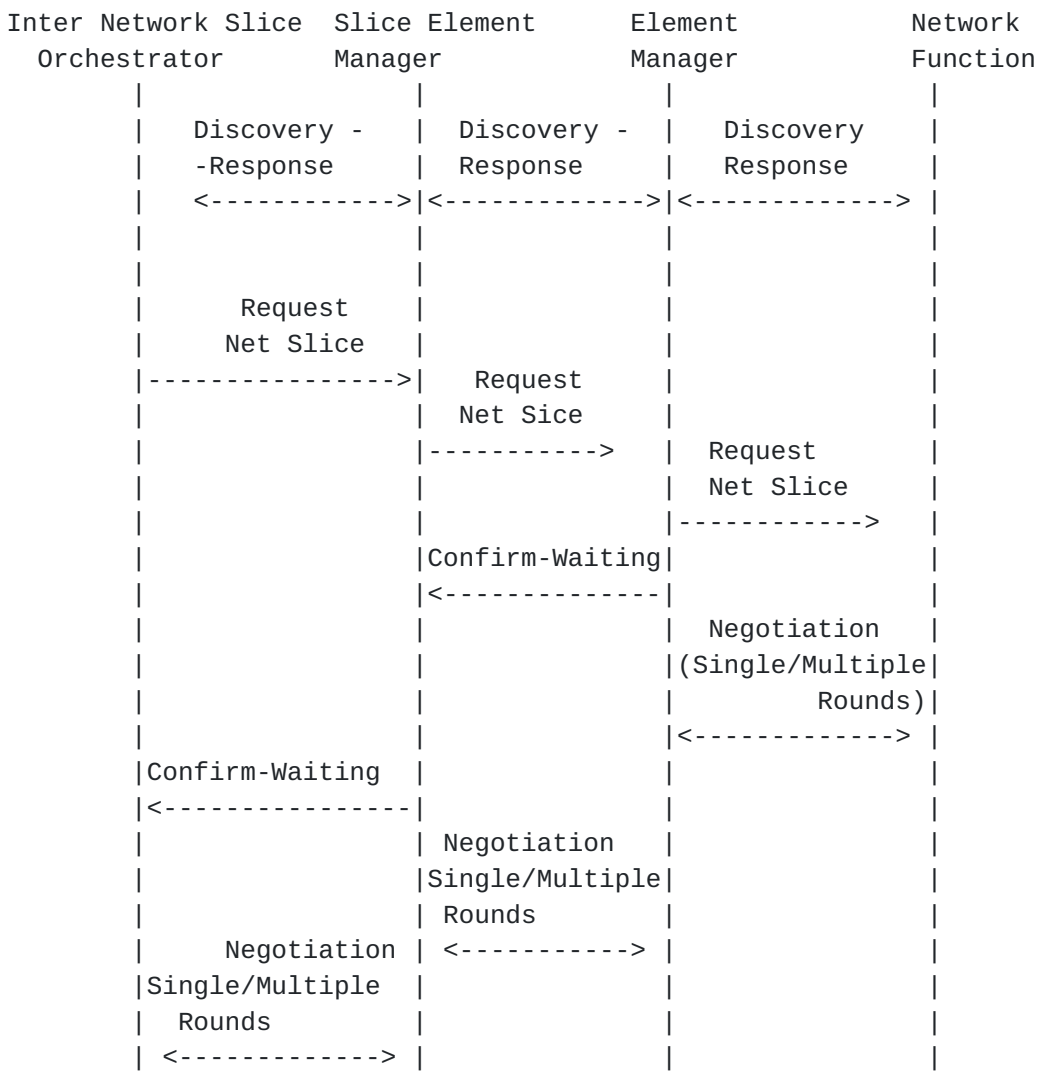


Figure 3: Network Slice Reservation / Release Messages Flow

6.2. Self- Management Operations

Self-management operations are focused on self-optimization and self-healing of network slice instances (including intra-slice functions management), network slice instance services and resources that are used for all slice instances. All these operations are combined with efficient and economical monitoring and reconfigurations at appropriate level. In order to make the management scalable and environment aware the management architecture is composed of many functional entities that follows the feedback loop management paradigm (aka autonomic management). The self-management functions may realize different goals and have to be coordinated according to slice instance and infrastructure operator policies. The self-management deals with dynamic (1) allocation of resources to slice instances in a economical way that provides required slice instances

performance, (2) self-optimization and self-healing of slice instances during their deployment (lifecycle management) and operations (3) self-optimization and self-healing of services of each slice instance. Their lifecycle, that is typically different than slice instance lifecycle should also be managed in the autonomous way. Despite the self-managed functions may have different goals and involved entities the slice instance self-management should be coordinated with self-management of their services and self-management of resources (inter-slice operations) should be aligned with in-slice self-management operations. In the implementation the self-management functionality is split between NS manager (that is a part of slice template) and slice orchestrator in case of slice management and between service specific management and NS manager in case of services that use a specific slice.

6.3. Programmability of the Management Plane

The Management Plane is composed of multiple functional entities and is responsible for resource, slice instance and slice service management. In case of slice instances and services their management comes as a part of appropriate slice or service template respectively. That way slice or service related management functions are instantiated for each slice and/or service. The Management Plane may expose a set of APIs which can be used by additional management services that are added independently on service or slice instance lifecycle. Using these APIs and allocation additional resource the slice or service operator can add advanced and new management functions. That way the Management Plane programmability is provided.

6.4. Management plane slicing protocols

At this stage it is too early do define protocols (IMHO). We have to define the management architecture first with functional entities and reference points/interfaces. Having them we could define which protocol(s) we want to use for each of them. Maybe we can mention some protocols but generally they should be a part of separate specification.

7. Service Functions and Mappings

8. OAM and Telemetry

OAM and telemetry to instrument the system need to be provided for each NSI so that the NSI provider can monitor the health of the NSI and so that the NSI owner can independently verify the health of their NSI.

Running OAM on the NSI from the perspective of its owner can be undertaken by the owner using the native tools for the NSI network type. For example if the NSI is IP, tools like ICMP [RFC792], ICMPv6 [RFC4443], or IPFIX [RFC7011] can be used. Similarly the native OAM tools for MPLS and Ethernet can be used. If the NSI provides a partial emulation of the network type that limits the ability to operate such native instrumentation tools, then this needs to be made clear to the NSI owner.

Similarly running OAM on the underlay will also use the native tools for the network type providing the underlay. Care must be taken that any OAM run by the NS provider does not impinge on the operation of the NSI, and SHOULD be undetectable in the NSI.

Telemetry will need to be provided to both the NS provider and the NSI owner. Telemetry of the underlay will use the NS providers pub-sub system of choice.

Telemetry of the NSI may be provided purely by the NSI owner installing a telemetry collection system. However significant efficiencies may be realised by if the NS provider exports relevant telemetry to the NSI owner's pub-sub system. Where this is done, consideration must be given to the security of the measurement and export system so to no information is leaked between NSIs.

9. IANA Considerations

This document makes no request of IANA.

10. Security Considerations

Each layer of the system has its own security requirements.

11. Acknowledgements

12. References

12.1. Normative References

[I-D.finn-detnet-architecture]
Finn, N. and P. Thubert, "Deterministic Networking Architecture", draft-finn-detnet-architecture-08 (work in progress), August 2016.

[I-D.qin-netslices-use-cases]
Qin, J., kiran.makhijani@huawei.com, k., Dong, J., Qiang, L., and S. Peng, "Network Slicing Use Cases: Network Customization for Different Services", draft-qin-netslices-use-cases-00 (work in progress), March 2017.

12.2. Informative References

[NS_WP] China Mobile Communication Corporation, Huawei Technologies Co. Deutsche Telekom AG,Volkswagen, "5G Service-Guaranteed Network Slicing White Paper", 2016, <http://labs.chinamobile.com/pdf/5GService-GuaranteedNetworkSlicingWhitePaper.pdf>.

Authors' Addresses

Liang Geng
China Mobile
Beijing
China

Email: gengliang@chinamobile.com

Jie Dong
Huawei Technologies
Huawei Campus, No. 156 Beiqing Rd.
Beijing 100095

Email: jie.dong@huawei.com

Stewart Bryant
Huawei Technologies
U.K.

Email: stewart.bryant@gmail.com

Kiran Makhijani
Huawei Technologies
2890 Central Expressway
Santa Clara CA 95050

Email: kiran.makhijani@huawei.com

Alex Galis
University College London
London
U.K.

Email: a.galis@ucl.ac.uk

Xavier de Foy
InterDigital Inc.
1000 Sherbrooke West
Montreal
Canada

Email: Xavier.Defoy@InterDigital.com

Slawomir Kuklinski
Orange

Email: slawomir.kuklinski@gmail.com