Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: May 4, 2017

A. Galis
University College London
K. Makhijani
D. Yu
Huawei Technologies

October 31, 2016

Autonomic Slice Networking-Requirements and Reference Model draft-galis-anima-autonomic-slice-networking-01

Abstract

This document describes the technical requirements and the related reference model for the intercommunication and coordination among devices in Autonomic Slicing Networking. The goal is to define how the various elements in a network slicing context work and orchestrate together, to describe their interfaces and relations. While the document is written as generally as possible, the initial solutions are limited to the chartered scope of the WG.

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 May 4, 2017.

Table of Contents

1. Introduction

The document "Autonomic Networking - Definitions and Design Goals" [RFC7575] explains the fundamental concepts behind Autonomic Networking, and defines the relevant terms in this space, as well as a high level reference model. This document defines this reference model with more detail, to allow for functional and protocol specifications to be developed in an architecturally consistent, non-overlapping manner. While the document is written as generally as possible, the initial solutions are limited to the chartered scope of the WG.

Most networks will run with some autonomic functions for the full networks or for a group of nodes [RFC7576] or for a group of slice networks while the rest of the network is traditionally managed.

The goal of this document is to focus on the autonomic slicing networking. [RFC7575] is focusing on fully or partially autonomic nodes or networks.

The proposed revised ANIMA reference model allows for this hybrid approach across all such capabilities.

This is a living document and will evolve with the technical solutions developed in the ANIMA WG. Sections marked with (*) do not represent current charter items.

While this document must give a long term architectural view, not all functions will be standardized at the same time.

2. The Network Slicing Overall View

2.1. Context

Network Slicing is end-to-end concept covering the radio and non-radio networks inclusive of access, core and edge / enterprise networks. It enables the concurrent deployment of multiple logical, self-contained and independent shared or partitioned networks on a common infrastructure platform.

From a business point of view, a slice includes combination of all relevant network resources / functions / assets required to fulfill a specific business case or service, including OSS, BSS and DevOps processes.

From the network infrastructure point of view, slicing instances require the partitioning and assignment of a set of resources that can be used in an isolated, disjunctive or non- disjunctive manner.

Examples of physical or virtual resources to be shared or partitioned would include: bandwidth on a network link, forwarding tables in a network element (switch, router), processing capacity of servers, processing capacity of network or network clouds elements. As such slice instances would contain:

(i) a combination/group of the above resources which can act as a network,

(ii) appropriate resource abstractions,

(iii) exposure of abstract resources towards service and management clients that are needed for the operation of slices

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

A complete slice is composed of not only various network functions which are based on virtual machines at C-RAN and C-Core, but also transport network resources which can be assigned to the slice at radio access/transport network. Different future businesses require different throughput, delay and mobility, and some businesses need very high throughput or/and low delay.

2.2. High Level Requirements

Slice creation: management plane create virtual or physical network functions and connects them as appropriate and instantiate them in the slice.

The instance of slice management then takes over the management and operations of all the (virtualised) network functions and network programmability functions assigned to the slice, and (re-)configure them as appropriate to provide the end-to-end service.

A complete slice is composed of not only various network functions which are based on virtual machines at C-RAN and C-Core, but also transport network resources which can be assigned to the slice at radio access/transport network. Different future businesses require different throughput, delay and mobility, and some businesses need very high throughput or/and low delay. Transport network shall provide QoS isolation, flexible network operation and management, and improve network utilization among different business.

QoS Isolation: Although traditional VPN technology can provide physical network resource isolation across multiple network segments, it is deemed far less capable of supporting QoS hard isolation, Which means QoS isolation on forwarding plane requires better coordination with management plane.

Independent Management Plane: Like above, network isolation is not sufficient, a flexible and more importantly a management plane per instance is required to operate on a slice independently and autonomously within the constraints of resources allocated to the slice.

Another flexibility requirement is that an operator can deploy their new business application or a service in network slice with low cost and high speed, and ensure that it does not affect existing of business applications adversely.

Programmability: Operator not only can slice a common physical infrastructure into different logical networks to meet all kinds of new business requirements, but also can use SDN based technology to improve the overall network utilization. By providing a flexible programmable interface; the 3rd party can develop and deploy new network business rapidly. Further, if a network slicing can run with its own slice controller, this network slicing will get more granular control capability [I-D.ietf-anima-autonomic-control-plane] to retrieve slice status, and issuing slicing flow table, statistics fetch etc.

Life cycle self-management: It includes creation, operations, re-configuration, composition, decomposition, deletion of slices. It would be performed automatically, without human intervention and based on a governance configurable model of the operators. As such protocols for slice set-up /operations /(de)composition / deletion must also work completely automatically. Self-management (i.e. self-configuration, self-composition, self-monitoring, self-optimisation, self-elasticity) is carried as part of the slice protocol characterization.

Extensibility: Since the Autonomic Slice Networking Infrastructure is a relatively new concept, it is likely that changes in the way of operation will happen over time. As such new networking functions will be introduced later, which allow changes to the way the slices operate.

Transport network shall provide QoS isolation, flexible network operation and management, and improve network utilization among different business.

The flexibility behind the slice concept needs to address QoS guarantee on the transport network and enable network openness.

Other considerations and requirements: TBD.

2.3. Key Terms and Definitions

A number of slice definitions were used in the last 10 years in distributed and federated testbed research [GENI], future internet research [ChinaCom09] and more recently in the context of 5G research [NGMN], [ONF], [IMT2020], [NGS-3GPP].

A unified Slice definition usable in the context of Autonomic Networking consist of 4 components:

- Service Instance component,
- Network Slice Instance component,
- Resources component and
- Slice Capability exposure component

The Service Instance component represents the end-user service or business services which are to be supported. It is 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 3rd parties.

A Network Slice Instance component is represented by a set of 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 which 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.

Slice Capability exposure component is allowing 3rd parties to access / use via APIs information regarding services provided by the slice (e.g. connectivity information, QoS, mobility, autonomicity, etc.) and to dynamically customize the network characteristics for different diverse use cases (e.g. ultra-low latency, ultra-reliability, value-added services for enterprises, etc.) within the limits set of functions by the operator. It includes a description

of the structure (and contained components) and configuration of the slice instance.

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. It is dedicated to a Network Function or shared between a set of Network Functions.

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 - It refers to processing functions in a network. This includes but is not limited to telecom nodes functionality, as well as switching functions e.g. switching function, IP routing functions.

Virtual Network Function - One or more virtual machines running different software and processes on top of high-volume servers, switches and storage, or cloud computing infrastructure, and capable of implementing network functions traditionally implemented via custom hardware appliances and middleboxes (e.g. router, NAT, firewall, load balancer, etc.).

3. Autonomic Slice Networking

This section describes the various elements in a network with autonomic functions, and how these entities work together, on a high level. Subsequent sections explain the detailed inside view for each of the autonomic network elements, as well as the network functions (or interfaces) between those elements.

Figure 1 shows the high level view of an Autonomic Slice Networking.

It consists of a number of autonomic nodes resources, which interact directly with each other. Those autonomic nodes resources provide a common set of capabilities across a network slice, called the "Autonomic Slice Networking Infrastructure" (ASNI). The ASNI provides functions like naming, addressing, negotiation, synchronization, discovery and messaging.

Autonomic network functions typically span several slices in the network. The atomic entities of an autonomic function are called the "Autonomic Service Agents" (ASA), which are instantiated on slices.

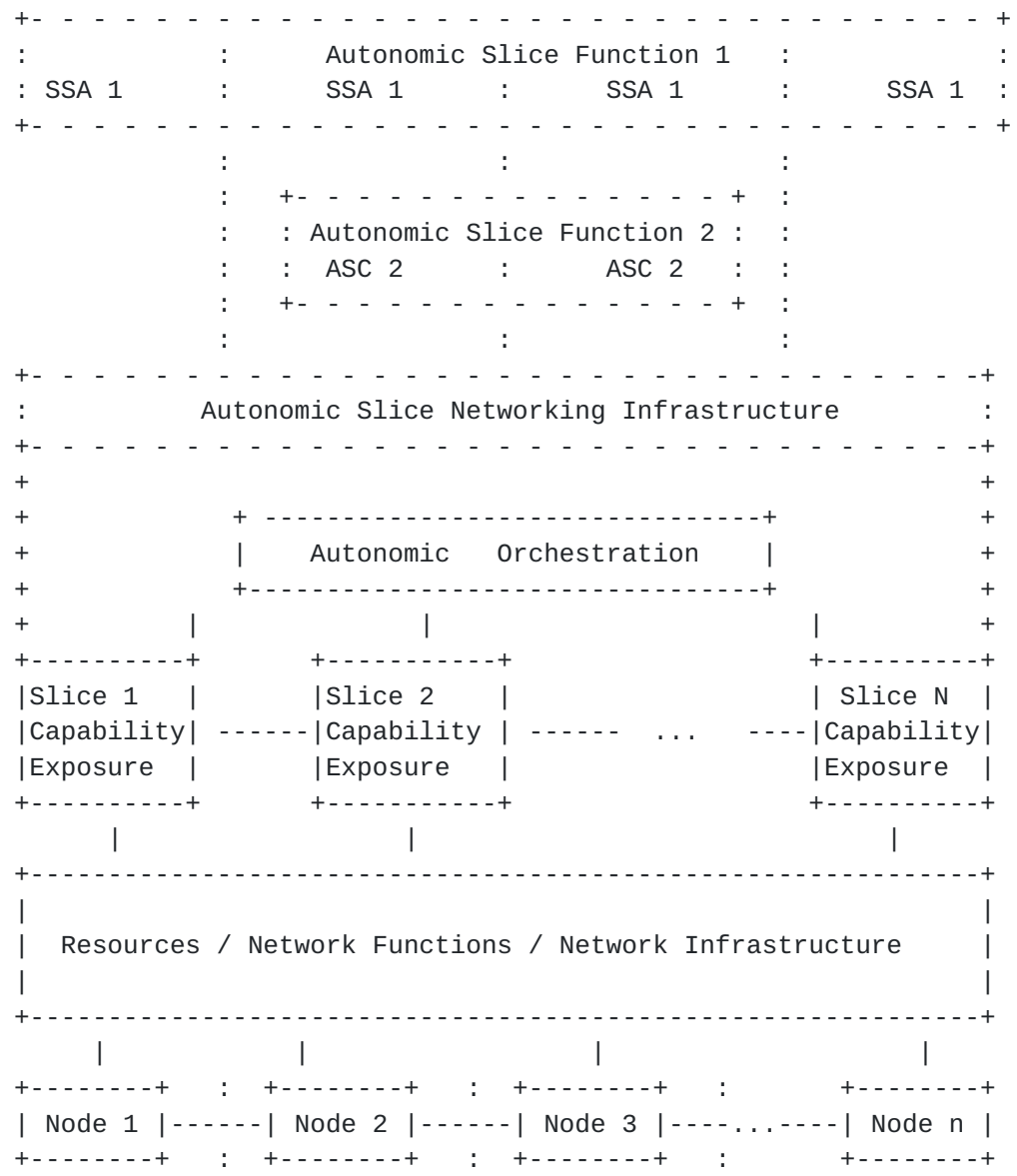

Figure 1: High level view of Autonomic Slice Networking

In a horizontal view, autonomic functions span across the network, as well as the Autonomic Slice Networking Infrastructure. In a vertical view, a slice always implements the ASNI, plus it may have one or several Autonomic Service Agents as part of slice capability exposure.

The Autonomic Networking Infrastructure (ASNI) therefore is the foundation for autonomic functions. The current charter of the ANIMA WG includes the specification of the ASNI, using a few autonomic functions as use cases. ASNI would represent a customized and an approach [I-D.ietf-anima-reference-model] for implementing a general purposed ASI.

Additionally, at least 2 autonomous functions are envisioned - Autonomous Slice control (ASC) and Slice Service agent (SSA). These are explained in sections below.

4. Autonomic Orchestration (*)

This section describes an autonomic orchestration and its functionality.

Orchestration refers to the functions that 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. It is expected to coordinate a number of interrelated resources, often distributed across a number of subordinate domains, and to assure transactional integrity as part of the process [TETT1] .

It is also the continuing process of allocating resources to satisfy contending demands in an optimal manner [TETT2] . The idea of optimal 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.

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.

5. The Autonomic Network Slicing Element

This section describes an autonomic slice network element and its internal architecture. The reference model explained in the document "Autonomic Networking - Definitions and Design Goals" [RFC7575] shows the sources of information that an autonomic service agent can leverage: Self-knowledge, network knowledge (through discovery), Intent [I-D.du-anima-an-intent] , and feedback loops. Fundamentally, there are two levels inside an autonomic node: the level of Autonomic

Service Agents, and the level of the Autonomic Slice Networking Infrastructure, with the former using the services of the latter.

Figure 2 illustrates this concept.

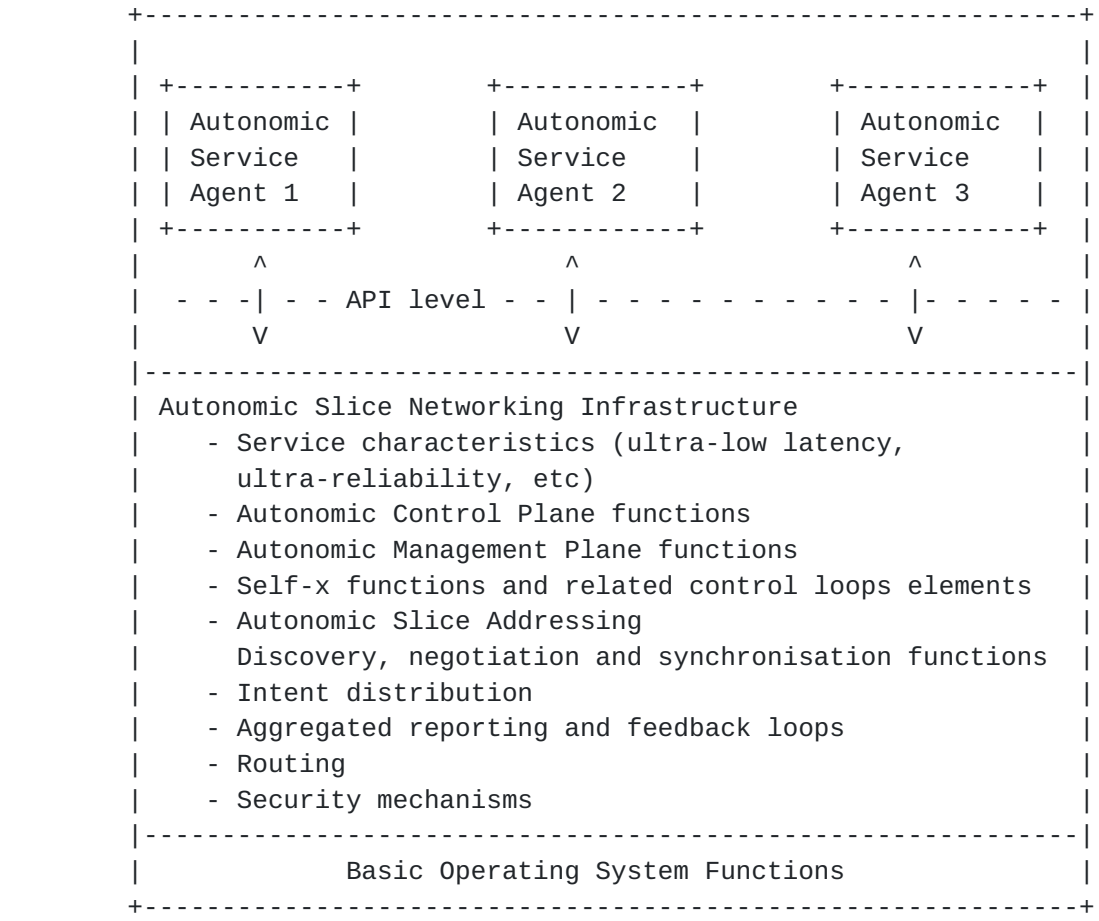

Figure 2: Model of an autonomic element

The Autonomic Slice Networking Infrastructure (lower part of Figure 2) contains slice specific data structures, for example trust information about itself and its peers, as well as a generic set of functions, independent of a particular usage. This infrastructure should be generic, and support a variety of Autonomic Service Agents (upper part of Figure 2). The Autonomic Control Plane is the summary of all interactions of the Autonomic Slice Networking Infrastructure with other services.

The use cases of "Autonomics" such as self-management, self-optimisation, etc, are implemented as Autonomic Service Agents. They use the services and data structures of the underlying autonomic networking infrastructure. The Autonomic Slice Networking Infrastructure should itself be self-managing.

The "Basic Operating System Functions" include the "normal OS", including the network stack, security functions, etc. Autonomic Network Slicing Element is a composition of autonomic slice service agents and autonomic slice control. Autonomic slice service agents obtain specific network resources and provide self-managing and self-controlling functions. An autonomic slice control is a higher-level autonomic function that takes the role of life-cycle management of a or many slice instances. There can be many slice control functions based on different types or attributes of slice.

6. The Autonomic Slice Networking Infrastructure

The Autonomic Networking Infrastructure provides a layer of common functionality across an Autonomic Network. It comprises "must implement" functions and services, as well as extensions. The Autonomic Slice Networking Infrastructure (ASNI) resides on top of an abstraction layer of resource, network function and network infrastructure as shown in figure 1. The document assumes abstraction layer enables different autonomous service agents to communicate with the underlying disaggregated and distributed network infrastructure, which itself maybe an autonomous networking (AN) domain or combination of multiple AN domain. The goal of ASNI is to provide autonomic life-cycle management of network slices.

6.1. Signaling Between Autonomic Slice Capability Exposures

The basic network capabilities are autonomically or through traditional techniques are learnt by slice agents. This depends on the fact that physical infrastructure is an autonomic network or not. The GASP signaling [I-D.ietf-anima-grasp] [I-D.liu-anima-grasp-distribution] [I-D.liu-anima-grasp-api] may be used to expose capabilities among SSAs or slice control. Optionally, SSA capabilities are more interesting to slice control autonomic functions for slice creation and install. The slice control must have the independent intelligence to process and filter capabilities to meet a network slice specification and have low level resources allocated for a slice through SSAs. 6.2 The Autonomic Control Plane.

6.2. The Autonomic Control Plane

TBD.

6.3. Naming & Addressing

A slice can be instantiated on demand, represents a logical network and therefore, must be assigned a unique identifier. A Slice Service Agent (SSA) may support functions of a single or multiple slices and communicate with each other, using the addressing of the Autonomic or

traditional (non-autonomic) Networking Infrastructure reside on. An SSA complies with ACP addressing mechanisms and in a domain, i.e., As part of the enrolment process the registrar assigns a number to the device, which is unique for slicing registrar and in ASNI domain.

6.4. Discovery

Slices themselves are not discovered but are instantiated through slice control autonomic function. However, both slice service agents and slice control functions must be discovered. Even though autonomic control plane will support discovery of all the SSAs and slice control, it may not be necessary.

6.5. Routing

Autonomic network slicing follows single routing protocol as described in [I-D.ietf-anima-autonomic-control-plane].

6.6. Intent

TBD.

7. Security and Trust Infrastructure

An Autonomic Slice Network is self-protecting. All protocols are secure by default, without the requirement for the administrator to explicitly configure security.

TBD.

7.1. Public Key Infrastructure

An autonomic domain uses a PKI model. The root of trust is a certification authority (CA). A registrar acts as a registration authority (RA).

A minimum implementation of an autonomic domain contains one CA, one Registrar, and network elements.

7.2. Domain Certificate

TBD.

8. Cross-Domain Functionality

TBD.

9. Autonomic Service Agents (ASA)

This section describes how autonomic services run on top of the Autonomic Slice Networking Infrastructure. There are at least two different types of autonomic functions are known:

1. Slice Service Agents are low level functions that learn capabilities of underlying infrastructure in terms of interfaces and available resources. They coordinate with Slice control to associate these resources with specific slice instances in effect performing full life cycle management of these resources.

2. Slice Control Autonomic Function: Slice control is responsible for high-level life-cycle management of a slice itself. This function will hold slice instances and their attributes related data structures in autonomic network slice infrastructure. As an example, a slice is defined for high bandwidth, highly secure transactional application. A slice control must be capable of negotiating resources required across different SSAs.

Out of scope are details of the mechanisms how the information is represented and exchanged between the two autonomic functions.

10. Management and Programmability

This section describes how an Autonomic Network is managed, and programmed.

10.1. How a Slice Network Is Managed

Slice network management is driven by Slice control, there are four categories operation:

1. Creating a network slice: Receive a network slice resource description request, upon successful negotiation with SSA allocate resource for it.

2. Shrink/Expand slice network: Dynamically alter resource requirements for a running slice network according service load.

3. (Re-)Configure slice network: The slice management user deploys a user level service into the slice. The slice control takes over the control of all the virtualized network functions and network programmability functions assigned to the slice, and (re-)configure them as appropriate to provide the end-to-end service.

4. Destroy slice network: Recycle all resource from the infrastructure.

10.2. Intent

TBD.

10.3. Control Loops

TBD.

10.4. APIs

The API model of for autonomic slicing semantically, is grouped into the following APIs to be defined.

10.4.1. Slice Control APIs

1. Create a slice network on user request. The request includes resource description. A unique identify a slice network, group all the resource.

2. Destroy a slice network identified by it's id.

3. Query a slice network slicing state by it's uuid.

4. Modify a slice network.

10.4.2. Service Agent - Device APIs

A service agent will interface with the physical infrastructure either through an autonomic network or traditional infrastructure. Depending upon which a device can either have autonomic or non-autonomic addressing. Service agents are required to perform life cycle management of network elements participating in a network slice and the following APIs are needed for addition, removal or update of a specific device. A device may be a logical or physical network element. Optionally, it may be a network function.

10.4.3. Service Agent - Port APIs

A port may be a physical or logical network port in a slice depending upon whether underlying infrastructure is an autonomic or traditional network. Service agents must be able to control the operational state of these ports. APIs are needed for addition, removal, update and operational state retrieval of a specific port.

10.4.4. Service Agent - Link APIs

A link connects two or more ports of devices described in above section. Service agents must be able to control the operational and connection status of these links through APIs for addition, removal, update and state retrieval for each link.

10.5. Relationship with MANO

Please refer to [MANO] for MANO introduction.

TBD.

11. Security Considerations

11.1. Threat Analysis

TBD.

11.2. Security Mechanisms

TBD.

12. IANA Considerations

This document requests no action by IANA.

13. Acknowledgements

Thanks Bing Liu for helping editing the draft.

This document was produced using the xml2rfc tool [RFC2629].

14. References

14.1. Normative References

[I-D.ietf-anima-grasp]
Bormann, C., Carpenter, B., and B. Liu, "A Generic Autonomic Signaling Protocol (GRASP)", draft-ietf-anima-grasp-07 (work in progress), September 2016.

[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>.

[RFC2629] Rose, M., "Writing I-Ds and RFCs using XML", RFC 2629, DOI 10.17487/RFC2629, June 1999, <http://www.rfc-editor.org/info/rfc2629>.

14.2. Informative References

[ChinaCom09]
"A. Galis et all - "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, <http://www.chinacom.org/2009/index.html>.".

[GENI] ""GENI Key Concepts" - Global Environment for Network Innovations (GENI) <http://groups.geni.net/geni/wiki/GENIConcepts>.".

[I-D.du-anima-an-intent]
Du, Z., Jiang, S., Nobre, J., Ciavaglia, L., and M. Behringer, "ANIMA Intent Policy and Format", draft-du-anima-an-intent-04 (work in progress), July 2016.

[I-D.ietf-anima-autonomic-control-plane]
Behringer, M., Eckert, T., and S. Bjarnason, "An Autonomic Control Plane", draft-ietf-anima-autonomic-control-plane-03 (work in progress), July 2016.

[I-D.ietf-anima-reference-model]
Behringer, M., Carpenter, B., Eckert, T., Ciavaglia, L., Pierre, P., Liu, B., Nobre, J., and J. Strassner, "A Reference Model for Autonomic Networking", draft-ietf-anima-reference-model-02 (work in progress), July 2016.

[I-D.liu-anima-grasp-api]
Carpenter, B., Liu, B., Wang, W., and X. Gong, "Generic Autonomic Signaling Protocol Application Program Interface (GRASP API)", draft-liu-anima-grasp-api-02 (work in progress), September 2016.

[I-D.liu-anima-grasp-distribution]
Liu, B. and S. Jiang, "Information Distribution over GRASP", draft-liu-anima-grasp-distribution-02 (work in progress), September 2016.

[I-D.strassner-anima-control-loops]
Strassner, J., Halpern, J., and M. Behringer, "The Use of Control Loops in Autonomic Networking", draft-strassner-anima-control-loops-01 (work in progress), April 2016.

[IMT2020] "ITU-T IMT2020 document "Report on Gap Analysis" - ITU-T IMT2020 ITU- Dec 2015 Published by ITU-T IMT2020. <http://www.itu.int/en/ITU-T/focusgroups/imt-2020/Pages/default.aspx>.".

[MANO] "ETSI European Telecommunications Standards Institute. Network Functions Virtualisation (NFV); Management and Orchestration v1.1.1. Website, December 2014. <http://www.etsi.org/deliver/etsi_gs/NFV-MAN/001_099/001/01.01.01_60/gs_ nfv-man001v010101p.pdf>.".

[NGMN] "Hedmar,P., Mschner, K., et all - NGMN Alliance document "Description of Network Slicing Concept", January 2016. <https://www.ngmn.org/uploads/media/160113_Network_Slicing_v1_0.pdf>.".

[NGS-3GPP]
""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] "Paul, M, Schallen, S., Betts, M., Hood, D., Shirazipor, M., Lopes, D., Kaippallimalit, J., - Open Network Fundation document "Applying SDN Architecture to 5G Slicing", April 2016. <https://www.opennetworking.org/images/stories/downloads/sdn-resources/technical-reports/Applying_SDN_Architecture_to_5G_Slicing_TR-526.pdf>.".

[RFC7575] Behringer, M., Pritikin, M., Bjarnason, S., Clemm, A., Carpenter, B., Jiang, S., and L. Ciavaglia, "Autonomic Networking: Definitions and Design Goals", RFC 7575, DOI 10.17487/RFC7575, June 2015, <http://www.rfc-editor.org/info/rfc7575>.

[RFC7576] Jiang, S., Carpenter, B., and M. Behringer, "General Gap Analysis for Autonomic Networking", RFC 7576, DOI 10.17487/RFC7576, June 2015, <http://www.rfc-editor.org/info/rfc7576>.

[TETT1] "Guerzoni,R., Vaishnavi,I.,Perez-Caparros, D., Galis,A., et all "Analysis of End-to-End Multi Domain Management and Orchestration Frameworks for Software Defined Infrastructures: an Architectural Survey", Transactions on Emerging Telecommunications Technologies, Wiley Online Library, DOI: 10.1002/ett.3103, June 2016, <onlinelibrary.wiley.com/doi/10.1002/ett.3103/pdf>.".

[TETT2] "Karl,H., Draexler,S., Peuster, M, Galis, A., et all "DevOps for Network Function Virtualization: An Architectural Approach", Transactions on Emerging Telecommunications Technologies, Wiley Online Library, DOI: 10.1002/ett.3084, July 2016, <http://onlinelibrary.wiley.com/doi/10.1002/ett.3084/full>.".

Authors' Addresses

Alex Galis
University College London
Department of Electronic and Electrical Engineering
Torrington Place
London WC1E 7JE
United Kingdom

Email: a.galis@ucl.ac.uk

Kiran Makhijani
Huawei Technologies
2890, Central Expressway
Santa Clara CA 95032
USA

Email: USA Email: kiran.makhijani@huawei.com

Delei Yu
Huawei Technologies
Q22, Huawei Campus
No.156 Beiqing Road
Hai-Dian District, Beijing 100095
P.R. China

Email: yudelei@huawei.com