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.

Copyright Notice

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.

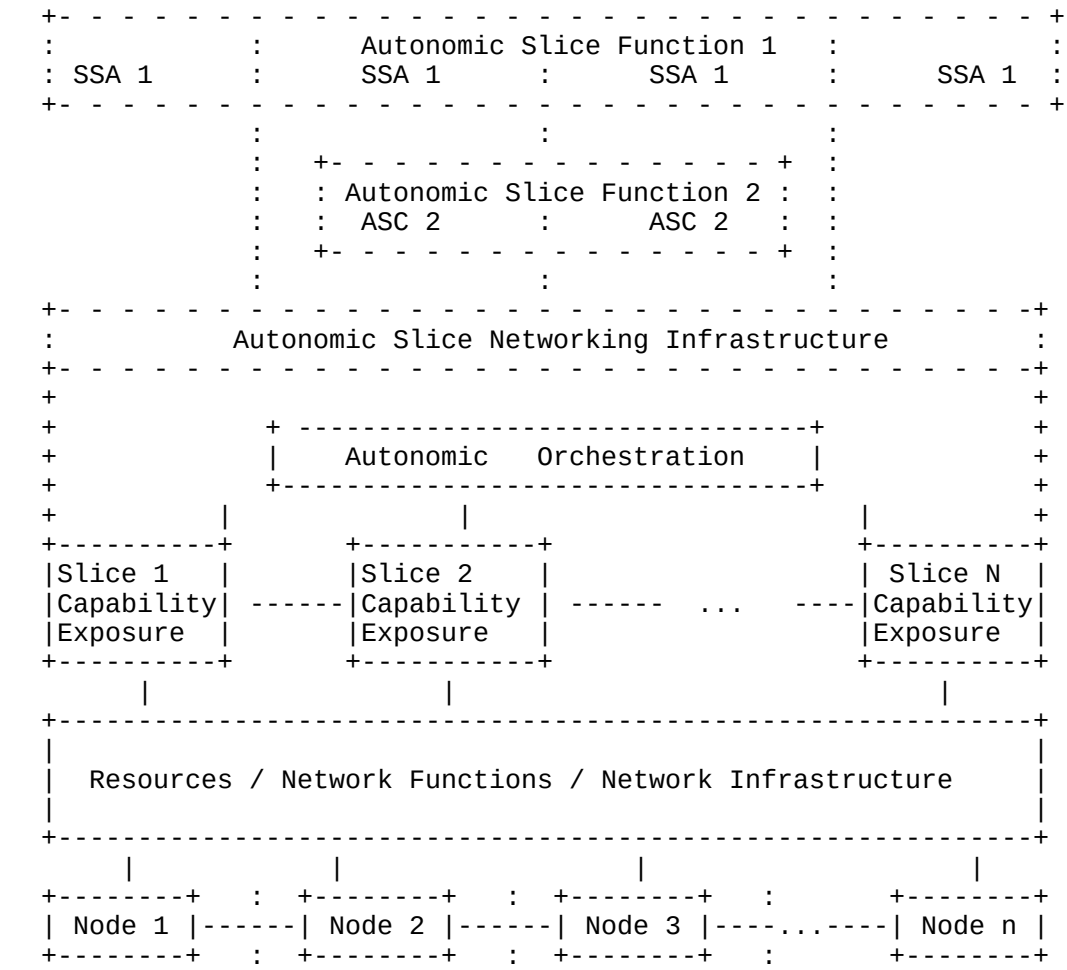


Figure 1: High level view of Autonomic Slice Networking

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.

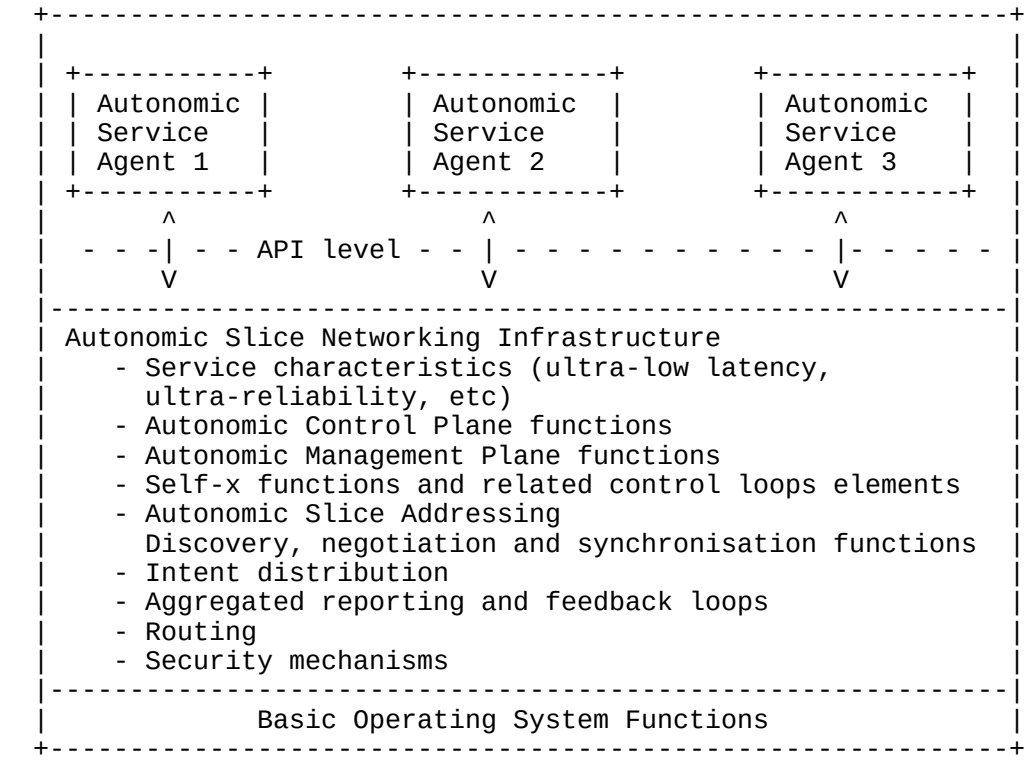


Figure 2: Model of an autonomic element

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

No Working Group
Internet-Draft
Intended Status: Standards Track
Expires: March 30, 2019

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

September 26, 2018

Autonomic Slice Networking
draft-galis-anima-autonomic-slice-networking-05

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), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

The list of current Internet-Drafts can be accessed at https://www.ietf.org/1id-abstracts.html

The list of Internet-Draft Shadow Directories can be accessed at https://www.ietf.org/shadow.html

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 September 6, 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. It enhances [ASN].

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. Key Terms and Context

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], [NS-ETSI]. Such definitions converge towards NS as group of components: Service Instance, Network Slice Instance, Resources and Slice Element Manager

In this draft we are using the following terms:

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 (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 Slicing (NS) refers to 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 resources include connectivity, compute, and storage resources.

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

Network slicing represents logically or physically isolated groups of network resources and network function/virtual network functions configurations separating its behavior from the underlying physical network.

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.

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 [SLICING]. 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) capability exposure of abstract resources towards service and management clients that are needed for the operation of slices

The capability exposure creates an abstraction of physical network devices that would provide information and information models allowing operators to manipulate the network resources. By utilizing open programmable network interfaces, it would enable access to control layer by customer interfaces and applications.

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 that 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, which is a subnetworks.

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 that can be assigned to the slice at radio access/transport network. Different future businesses [5GNS], [PER-NS] 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.

(1) Separation from partition of the physical network: Network slicing represents logically or physically isolated groups of network resources and network function/virtual network functions configurations separating its behavior from the underlying physical network.

(2) 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.

(3) 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.

(4) 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.

(5) Stringent Resource Characteristics: A Network Slicing aware infrastructure allows operators to use part of the network resources to meet stringent resource characteristics.

(6) Type of resources: Network Slice instance is a dedicated network

that is build and activated on an infrastructure mainly composed of, but not limited to, connectivity, storage and computing.

(7) 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.

(8) 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.

(9) Network slice Self-management: Network slices will need to be self-managed by automated, autonomic and autonomous systems in order to cope with dynamic requirements, such as flexible scalability, extensibility, elasticity, residency and reliability of an infrastructure. Network slices will need to be self-managed by automated, autonomic and autonomous systems in order to cope with dynamic requirements, such as scalability or extensibility of an infrastructure. A common information model describing uniformly the NS in a single and/or multiple domain would support such self-managed.

(10) 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.

(11) Network Slice elasticity: A Network Slice instance has the mechanisms and triggers for the growth/shrinkage of all resources, and/or network and service functions as enabled by a common information model that explicitly provides for elasticity policies for scaling up/down resources.

(12) Multiple domains activation: Network slice instances are concurrently activated as multiple logical, self-contained and independent, partitioned network functions and resources on a specific infrastructure domain.

(13) Resource Exposure: Each network slice has the ability to dynamically expose and possibly negotiate the parameters that characterize an NS as enabled by a common information model that explicitly provides monitoring policies for all model descriptors.

(14) Network Tenants: Network slicing support tenants that are strongly independent on infrastructure as enabled by a common information model that explicitly provides for a level of tenants management for the resources dedicated to an instance of network slice.

(15) End-to-end Orchestration of Network Slicing: Coordinating underlay network infrastructure and service function resources. In the process of orchestration of network slice, resource registration and templates for network slice repository are needed.

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.

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

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

From the tenant point of view, network slice provides different capabilities, specifically in terms of their management and control capabilities, and how much of them the network service provider hands over to the slice tenant. As such there are two kinds of slices: (A) Inner slices, understood as the partitions used for internal services of the provider, retaining full control and management of them. (B)

Outer slices, being those partitions hosting customer services, appearing to the customer as dedicated networks.

Network Slicing lifecycle includes the management plane selecting a group 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 as appropriate, and it instantiates all of the network and service functions assigned to the slice. For slice operations, the control plane takes over governing of all the network resources, network 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.

One expected autonomic Slice Networking function is the capability and resource Usability for a slice. Applications or services requiring information of available slice capabilities and resources are satisfied by abstracted resource view and control. Usability of capabilities and resources can be enabled either by resource publishing or by discovery. In the latter case, the service performs resource collection directly from the provider of the slice by using discovery mechanisms to get total information about the available resources to be consumed. In the former, the network provider exposes available resources to services (e.g., through a resource catalog) reducing the amount of detail of the underlying network.

Slice Element Manager (SEM) is installed for each control domain. Control domain is defined according to geographic location and control functions. Each SEM converts requirements from orchestrator into virtual resources and manages virtual resources of a slice. SEM also exchanges information of virtual resources with other slice element managers via a dedicated resource interface. SEM provides also capability exposure facilities by 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.

Physical Element Manager (PEM) is installed for each control domain. Control domain is defined according to geographic location and control functions. PEM exchanges information of virtual resource with SEM via virtual resource interface and interconverts between virtual resource and physical resource. The PEM orders physical functions (ex. switches) to allocate physical resource via physical resource interface.

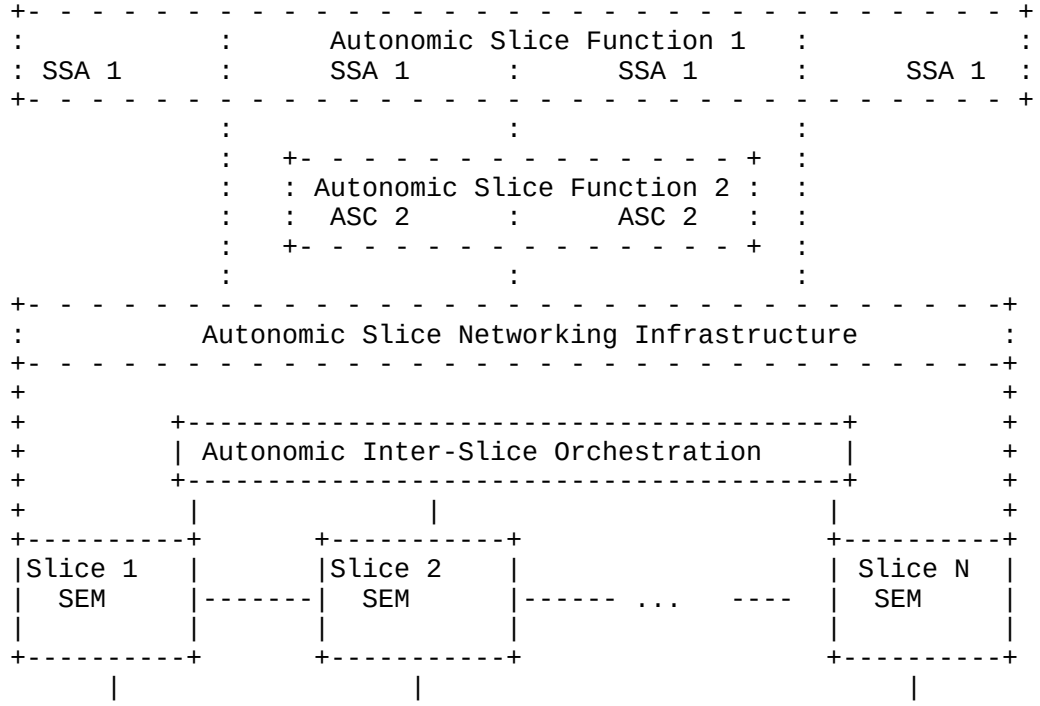
Figure 1 shows the high level view of an Autonomic Slice Networking.

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 ASN 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.

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.



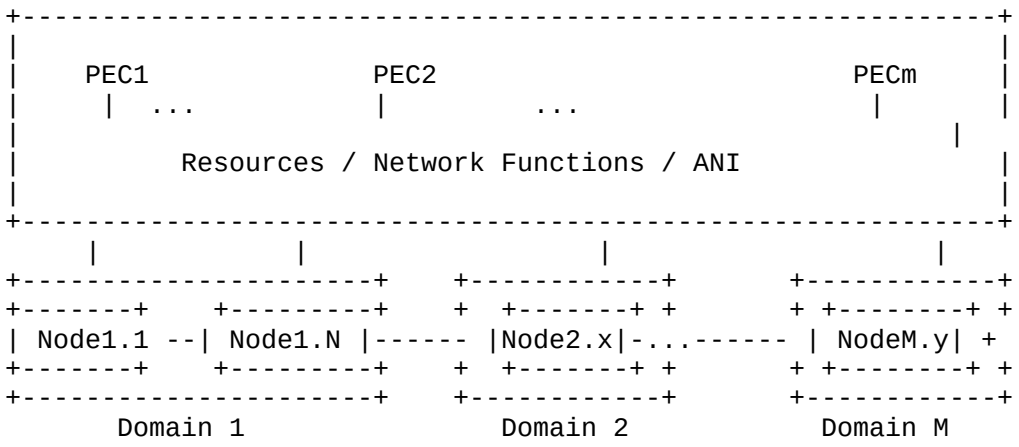


Figure 1: High level view of Autonomic Slice Networking

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 Inter-Slice Orchestration

This section describes an autonomic orchestration and its functionality.

Orchestration refers to the system functions that:

* automated 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.

* 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].

* 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.

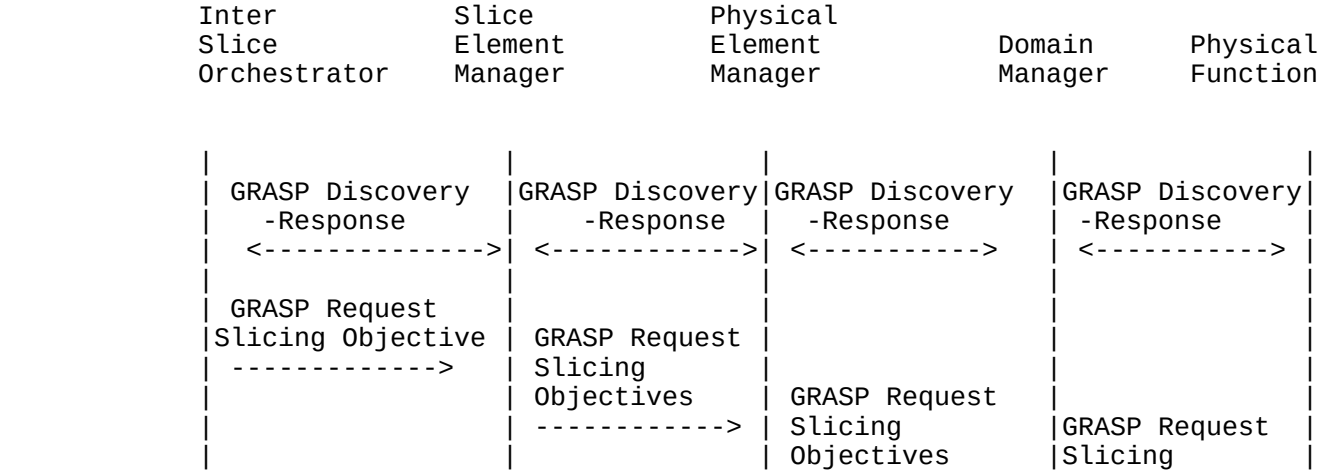
* autonomically coordinate and trigger of slice elasticity and placement of logical resources in slices.

* 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 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 [SERMODEL], 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 [I-D.strassner-anima-control-loops].

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. GRASP Resource Reservation / Release Messages flow



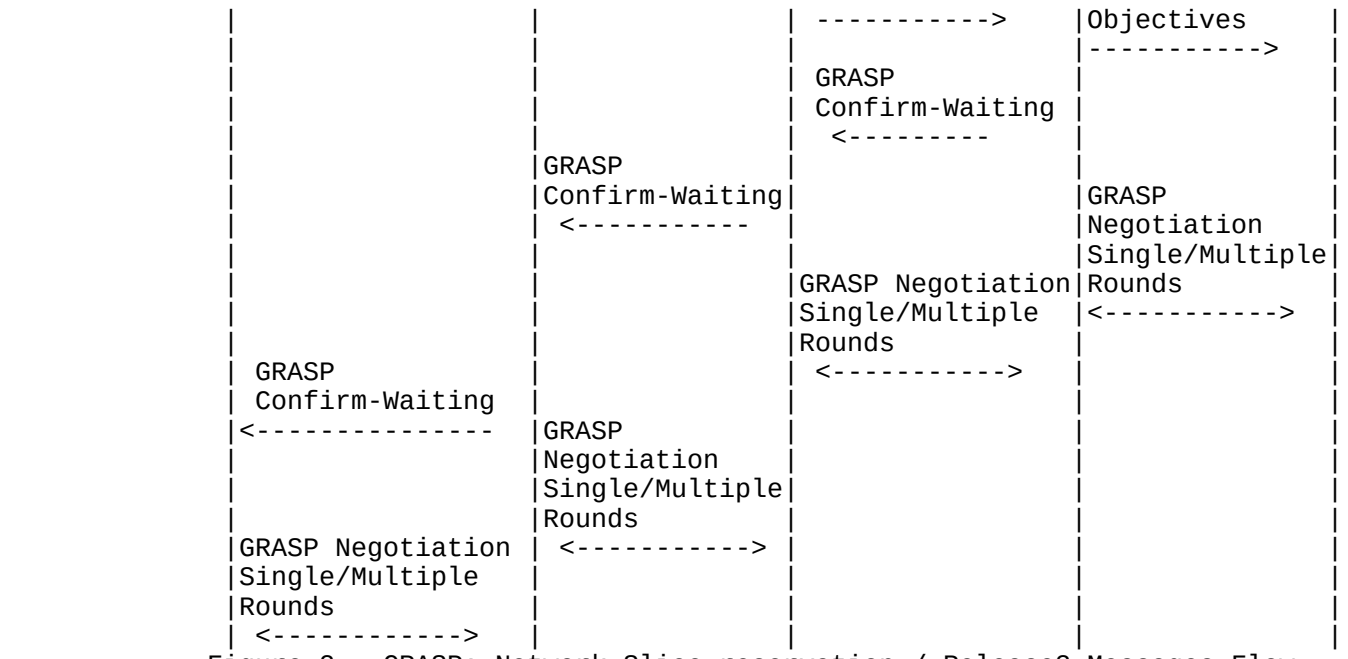


Figure 2 - GRASP: Network Slice reservation / Release3 Messages Flow

The above message sequence figure shows the message flows of the interactions between Inter-Slice Orchestrator, Slice Element Manager, Physical Element Manager, Domain Manager and Physical Network functions.

6. 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-management, 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. The self management functionality (self-configuration, self-optimisation, self- healing) could be implemented across the Inter Slice Orchestrator, Slice Element Manager and Physical Element Manager. Such functionality deals with dynamic

* coordination the life cycle of slices

* allocation of resources to slice instances in an efficient way that provides required slice instances performance,

* self-configuration, self-optimization and self-healing of slice instances during their lifecycle management including deployment and operations

* self-configuration, self-optimization and self-healing of services of each slice instance. Service lifecycle, that is typically different than slice instance lifecycle should also be managed in the autonomous way.

Figure 3 illustrates this concept.

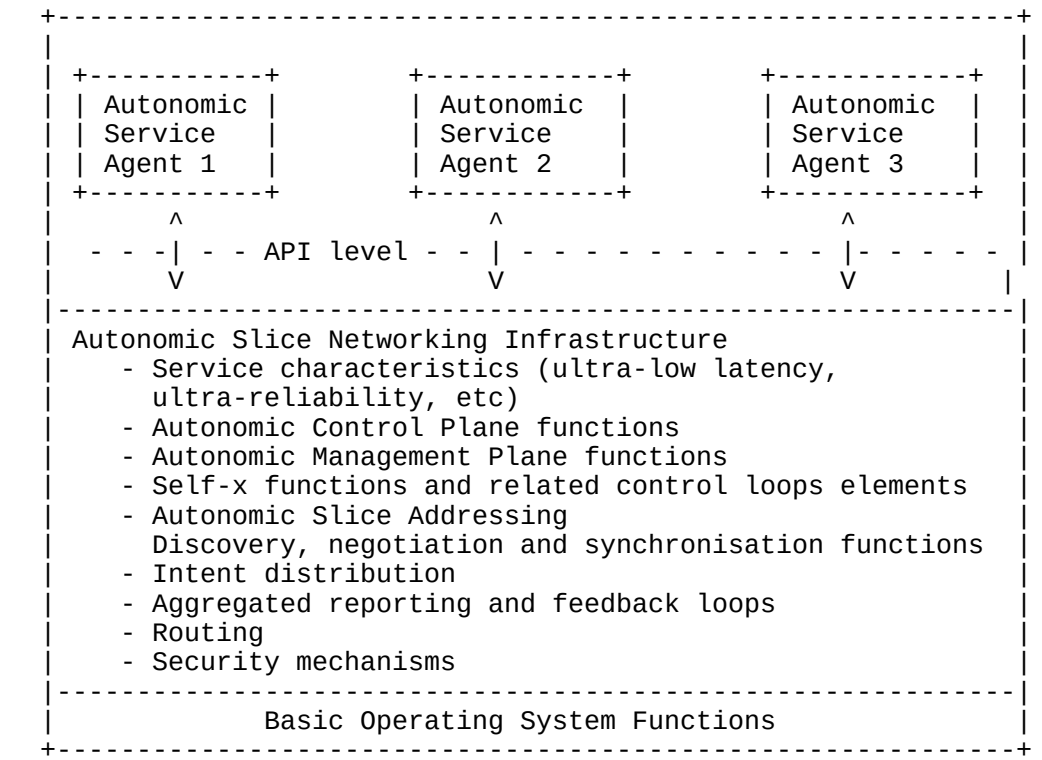


Figure 3: Model of an autonomic element

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.

7. The Autonomic Slice Networking Ianfrastructure

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.

7.1. Signaling Between Autonomic Slice Element Managers

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 extensions signaling [I-D.liu-anima-grasp-distribution] [I-D.liu-anima-grasp-api] [I-D.ietf-anima-grasp] may be used for

* Discovery of SEMs - a process by which an one SEM discovers peers according to a specific discovery objective. The discovered SEMs peers may later be used as negotiation counterparts or as sources of other coordination activities.

* Negotiation between SEMs - a process by which two SEMs interact to agree on slice logical resource settings that best satisfy the objectives of both SEMs.

* The Synchronization between SEMs - a process by which Orchestrator and SEMs interact to receive the current state of capability exposure values used at a given time in other SEM. This is a special case of negotiation in which information is sent but the SEM or Orchestrator do not request their peers to change configuration settings.

* Self configuration of SEMs - a process by which Orchestrator and SEMs interact to receive the current state of capability exposure values used at a given time in other SEM. This is a special case of synchronization in which information is sent and the SEM is requesting their peers to change configuration settings.

* Self optimization of SEMs - a process by which Orchestrator and SEMs interact to receive the current state of capability exposure values used at a given time in other SEMs. This is a special case of configuration in which information is sent and the SEM is requesting their peers to change logical resource settings in a slice based on an optimisation criteria.

* Mediation for slice resources - a process by which two SEMs interact to agree to logically move resources between slices that best satisfy the objectives of both SEMs triggering of slice elasticity and placement of logical resources in slices. Th???is is a special case of negotiation in which information is sent Orchestrator do request SEMs to change logical resource configuration settings.

* Triggering and governing of elasticity ? a process for autonomic scaling intent configuration mechanisms and resources on the slice level; it allows rapid provisioning, automatic scaling out, or in, of resources. Scale in/out criteria might be used for network autonomics in order the controller to react to a certain set of variations in monitored slices.

* Providing on-demand a self-service network slicing.

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.

7.2. The Autonomic Control Plane

TBD.

7.3. 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.

7.4. 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.

7.5. Routing

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

8. 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.

8.1. 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.

8.2. Domain Certificate

TBD.

9. Cross-Domain Functionality

TBD.

10. 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.

11. Management and Programmability

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

11.1. How a Slice Network Is Managed

Slice autonomic management is driven by Slice Element Managers, there are five 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.

5. Self-X slice operation: namely self-configuration, self-composition, self-monitoring, self-optimisation, self-elasticity would be carried out as part of new slice protocols.

11.2. Autonomic Resource Information Model

TBD.

The proposed autonomic resource information model is presented as a tree structure of attributes including the following elements: connectivity resources, storage resources, compute resources, service instances, network slice level attributes, etc. The Yang language would be used to represent the autonomic resource information model.

11.3. Control Loops

TBD.

11.4. APIs

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

11.4.1. 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.

11.4.2. 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.

11.4.3. 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.

11.4.4. 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.

11.5. Relationship with MANO

Please refer to [MANO] for MANO introduction.

12. Security Considerations

12.1. Threat Analysis

TBD.

12.2. Security Mechanisms

TBD.

13. IANA Considerations

This document requests no action by IANA.

14. Acknowledgements

This document was converted to nroff by Stuart Clayman (UCL) to comply with RFC format [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-10 (work in progress), March 2017.

[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>.

[RFC7665] Halpern, J., Pignataro, C., "Service Function Chaining (SFC) Architecture", October 2015 <https://tools.ietf.org/html/rfc7665>.

[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>.

[NS1] L. Geng, J. Dong, S. Bryant, K., Makhijani, A., Galis, X. de Foy, S. Kuklinski, - "Network Slicing Architecture", July 2017. <https://tools.ietf.org/html/draft-geng-netslices- architecture-02>.

[NS2] L. Geng, L. Wang, S. Kuklinski, L. Qiang, S. Matsushima, A., Galis, L. Contreras - "Problem Statement of Supervised Heterogeneous Network Slicing", October 2017 <https://datatracker.ietf.org/doc/draft-geng-coms-problem-statement/>.

[ASN] A., Galis, K., Makhijani, D. Yu, B. Liu - "Autonomic Slice Networking-Requirements and Reference Model" - May 2017 < https://datatracker.ietf.org/doc/draft-galis-anima-autonomic-slice-networking/>.

[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, July 2016, <http://www.rfc-editor.org/info/rfc7576>.

[TETT1] Guerzoni, R., Vaishnavi, I., Pares-Caparros, D., Galis, A., et al, "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., Draxler, 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>.

[SERMODEL] C., Borman, B. Carpenter, B., Liu, "Service Models Explained " draft-wu-opsawg-service-model-explained-05 <https://datatracker.ietf.org/doc/draft-wu-opsawg-service-model- explained/>.

[5GNS] Galis, A. (UCL), Chih-Lin I (China Mobile) - "Towards 5G Network Slicing - Motivations and Challenges" March 2017, IEEE 5G Tech Focus, Volume 1, Number 1, March 2017- <http://5g.ieee.org/tech-focus/march-2017#networkslicing>.

[PER-NS] Galis, A. - " Perspectives on Network Slicing - Towards the New ’Bread and Butter’ of Networking and Servicing", IEEE SDN Initiative - January 2018 <https://sdn.ieee.org/newsletter/january-2018/perspectives-on- network-slicing-towards-the-new-bread-and-butter-of-networking-and-servicing>.

[NS-ETSI] "Network Functions Virtualisation (NFV) Release 3; Evolution and Ecosystem; Report on Network Slicing Support with ETSI NFV Architecture Framework- ETSI GR NFV-EVE 012 V3.1.1 (2017-12)" <http://www.etsi.org/deliver/etsi_gr/NFV-

EVE/001_099/012/03.01.01_60/gr_NFV-EVE012v030101p.pdf>

Authors' Addresses

Alex Galis (editor)
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

Bing Liu
Huawei Technologies Co., Ltd
Q14, Huawei Campus
No.156 Beiqing Road
Hai-Dian District, Beijing 100095
P.R. China

Email: leo.liubing@huawei.com

Network Management Research Group J. Nobre
Internet-Draft University of Vale do Rio dos Sinos
Intended status: Informational L. Granville
Expires: April 10, 2017 Federal University of Rio Grande do Sul
A. Clemm
Sympotech
A. Prieto
Cisco Systems
October 7, 2016

Autonomic Networking Use Case for Distributed Detection of SLA Violations
draft-irtf-nmrg-autonomic-sla-violation-detection-04

Abstract

This document describes a use case for autonomic networking in distributed detection of Service Level Agreement (SLA) violations. It is one of a series of use cases intended to illustrate requirements for autonomic networking.

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 April 10, 2017.

Copyright Notice

Table of Contents

1. Introduction

The Internet has been growing dramatically in terms of size and capacity, and accessibility in the last years. Communication requirements of distributed services and applications running on top of the Internet have become increasingly demanding. Some examples are real-time interactive video or financial trading. Providing such services involves stringent requirements in terms of acceptable latency, loss, or jitter. Those requirements lead to the articulation of Service Level Objectives (SLOs) which are to be met. Those SLOs become part of Service Level Agreements (SLAs) that articulate a contract between the provider and the consumer of a service. To fulfill a service, it needs to be ensured that the SLOs are met. Examples of service fulfillment clauses can be found on [RFC7297]). Violations of SLOs can be associated with significant financial loss, which can by divided in two types. First, there is the loss incurred by the service users (e.g., the trader whose orders are not executed in a timely manner) and the loss incurred by the service provider in terms of penalties for not meeting the service and loss of revenues due to reduced customer satisfaction. Thus, the service level requirements of critical network services have become a

key concern for network administrators. To ensure that SLAs are not being violated, service levels need to be constantly monitored at the network infrastructure layer. To that end, network measurements must take place.

Network measurement mechanisms are performed through either active or passive measurement techniques. In passive measurements, production traffic is observed. Network conditions are checked in a non intrusive way because no monitoring traffic is created by the measurement process itself. In the context of IP Flow Information EXport (IPFIX) WG, several documents were produced to define passive measurement mechanisms (e.g., flow records specification [RFC3954]). Active measurement, on the other hand, is intrusive because it injects synthetic traffic into the network to measure the network performance. The IP Performance Metrics (IPPM) WG produced documents that describe active measurement mechanisms, such as: One-Way Active Measurement Protocol (OWAMP) [RFC4656], Two-Way Active Measurement Protocol (TWAMP) [RFC5357], and Cisco Service Level Assurance Protocol (SLA) [RFC6812]. Besides that, there are some mechanisms that do not fit into either active or passive categories, such as Performance and Diagnostic Metrics Destination Option (PDM) techniques [draft-ietf-ippm-6man-pdm-option].

Active measurement mechanisms offer a high level of control of what and how to measure. It also does not require inspecting production traffic. Because of this, it usually offers better accuracy and privacy than passive measurement mechanisms. Traffic encryption and regulations that limit the amount of payload inspection that can occur are non-issues. Furthermore, active measurement mechanisms are able to detect end-to-end network performance problems in a fine-grained way (e.g., simulating the traffic that must be handled considering specific Service Level Objectives - SLOs). As a result, active measurements are often preferred over passive measurement for SLA monitoring. Measurement probes must be hosted in network devices and measurement sessions must be activated to compute the current network metrics (e.g., considering those described in [RFC4148]). This activation should be dynamic in order to follow changes in network conditions, such as those related with routes being added or new customer demands.

The activation of active measurement sessions (hosted in senders and responders considering the architecture described by Cisco [RFC6812]) is expensive in terms of the resource consumption, e.g., CPU cycle and memory footprint, and monitoring functions compete for resources with other functions, including routing and switching. Besides that, the activated sessions also increase the network load because of the injected traffic. The resources required and traffic generated by the active measurement sessions are a function of the number of

measured network destinations, i.e., with more destinations the larger will be the resources and the traffic needed to deploy the sessions. Thus, to have a better monitoring coverage it is necessary to deploy more sessions what consequently turns increases consumed resources. Otherwise, enabling the observation of just a small subset of all network flows can lead to an insufficient coverage. Hence, the decision how to place measurement probes becomes an important management activity, so that with a limited amount of measurement overhead the maximum benefits in terms of service level monitoring are obtained.

2. Definitions and Acronyms

Active Measurements: Techniques to measure service levels that involves generating and observing synthetic test traffic

Passive Measurements: Techniques used to measure levels based on observation of production traffic

SLA: Service Level Parameter

SLO: Service Level Objective

P2P: Peer-to-Peer

3. Current Approaches

The current best practice in feasible deployments of active measurement solutions to distribute the available measurement sessions along the network consists in relying entirely on the human administrator expertise to infer which would be the best location to activate such sessions. This is done through several steps. First, it is necessary to collect traffic information in order to grasp the traffic matrix. Then, the administrator uses this information to infer which are the best destinations for measurement sessions. After that, the administrator activates sessions on the chosen subset of destinations considering the available resources. This practice, however, does not scale well because it is still labor intensive and error-prone for the administrator to compute which sessions should be activated given the set of critical flows that needs to be measured. Even worse, this practice completely fails in networks whose critical flows are too short in time and dynamic in terms of traversing network path, like in modern cloud environments. That is so because fast reactions are necessary to reconfigure the sessions and administrators are not just enough in computing and activating the new set of required sessions every time the network traffic pattern changes. Finally, the current active measurements practice usually covers only a fraction of the network flows that should be observed,

which invariably leads to the damaging consequence of undetected SLA violations.

4. Problem Statement

The problem to solve involves automating the placement of active measurement probes in the most effective manner possible. Specifically, assuming a bounded resource budget that is available for measurements, the problem becomes how to place those measurement probes such that the likelihood of detecting service level violations is maximized, and subsequently performing the required configurations. The method should be embeddable as management software inside network devices that controls the deployment of active measurement mechanisms. The method shall furthermore be dynamic and be able to adapt to changing network conditions.

5. Benefits of an Autonomic Solution

The use case considered here is the distributed autonomic detection of SLA violations. The use of Autonomic Networking (AN) properties can help such detection through an efficient activation of measurement sessions [P2PBNM-Nobre-2012]. The problem to be solved by AN in the present use case is how to steer the process of measurement session activation by a complete solution that sets all necessary parameters for this activation to operate efficiently, reliably and securely, with no required human intervention, while allowing for their input.

We advocate for embedding Peer-to-Peer (P2P) technology in network devices in order to improve the measurement session activation decisions using autonomic control loops. The provisioning of the P2P management overlay should be transparent for the network administrator. It would be possible to control the measurement session activation using local data and logic and to share measurement results among different network devices.

An autonomic solution for the distributed detection of SLA violations can provide several benefits. First, efficiency: this solution could optimize the resource consumption and avoid resource starvation on the network devices. In practice, the solution should maximize the benefits of SLA monitoring (i.e., maximize the likelihood of SLA violations being detected) by operating within a given resource budget. This optimization comes from different sources: taking into account past measurement results, taking into account other observations (such as, observations of link utilizations and passive measurements, where available) sharing of measurement results between network devices, better efficiency in the probe activation decisions, etc. Second, effectiveness: the number of detected SLA violations

could be increased. This increase is related with a better coverage of the network. Third, the solution could decrease the time necessary to detect SLA violations. Adaptivity features of an autonomic loop could capture faster the network dynamics than an human administrator. Finally, the solution could help to reduce the workload of human administrator, or, at least, to avoid their need to perform operational tasks.

6. Intended User and Administrator Experience

The autonomic solution should not require the human intervention in the distributed detection of SLA violations. Besides that, it could enable the control of SLA monitoring by less experienced human administrators. However, some information may be provided from the human administrator. For example, the human administrator may provide the SLOs regarding the SLA being monitored. The configuration and bootstrapping of network devices using the autonomic solution should be minimal for the human administrator. Probably it would be necessary just to inform the address of a device which is already using the solution and the devices themselves could exchange configuration data.

7. Analysis of Parameters and Information Involved

The active measurement model assumes that a typical infrastructure will have multiple network segments and Autonomous Systems (ASs), and a reasonably large number of several of routers and hosts. It also considers that multiple SLOs can be in place in a given time. Since interoperability in a heterogenous network is a goal, features found on different active measurement mechanisms (e.g. OWAMP, TWAMP, and IPSLA) and programability interfaces (e.g., Cisco’s EEM and onePK) could be used for the implementation. The autonomic solution should include and/or reference specific algorithms, protocols, metrics and technologies for the implementation of distributed detection of SLA violations as a whole.

7.1. Device Based Self-Knowledge and Decisions

Each device has self-knowledge about the local SLA monitoring. This could be in the form of historical measurement data and SLOs. Besides that, the devices would have algorithms that could decide which probes should be activated in a given time. The choice of which algorithm is better for a specific situation would be also autonomic.

7.2. Interaction with other devices

Network devices should share information about service level measurement results. This information can speed up the detection of SLA violations and increase the number of detected SLA violations. In any case, it is necessary to assure that the results from remote devices have local relevancy. The definition of network devices that exchange measurement data, i.e., management peers, creates a new topology. Different approaches could be used to define this topology (e.g., correlated peers [P2PBNM-Nobre-2012]). To bootstrap peer selection, each device should use its known endpoints neighbors (e.g., FIB and RIB tables) as the initial seed to get possible peers.

8. Comparison with current solutions

There is no standartized solution for distributed autonomic detection of SLA violations. Current solutions are restricted to ad hoc scripts running on a per node fashion to automate some administrator’s actions. There some proposals for passive probe activation (e.g., DECON and CSAMP), but without the focus on autonomic features. It is also mentioning a proposal from Barford et al. to detect and localize links which cause anomalies along a network path.

9. Related IETF Work

The following paragraphs discuss related IETF work and are provided for reference. This section is not exhaustive, rather it provides an overview of the various initiatives and how they relate to autonomic distributed detection of SLA violations. 1. [LMAP]: The Large-Scale Measurement of Broadband Performance Working Group aims at the standards for performance management. Since their mechanisms also consist in deploying measurement probes the autonomic solution could be relevant for LMAP specially considering SLA violation screening. Besides that, a solution to decrease the workload of human administrators in service providers is probably highly desirable. 2. [IPFIX]: IP Flow Information EXport (IPFIX) aims at the process of standardization of IP flows (i.e., netflows). IPFIX uses measurement probes (i.e., metering exporters) to gather flow data. In this context, the autonomic solution for the activation of active measurement probes could be possibly extended to address also passive measurement probes. Besides that, flow information could be used in the decision making of probe activation. 3. [ALTO]: The Application Layer Traffic Optimization Working Group aims to provide topological information at a higher abstraction layer, which can be based upon network policy, and with application-relevant service functions located in it. Their work could be leveraged for the definition of

the topology regarding the network devices which exchange measurement data.

10. Acknowledgements

We wish to acknowledge the helpful contributions, comments, and suggestions that were received from Mohamed Boucadair, Bruno Klauser, Eric Voit, and Hanlin Fang.

11. IANA Considerations

This memo includes no request to IANA.

12. Security Considerations

The bootstrapping of a new device follows the approach proposed on anima wg [draft-anima-boot], thus in order to exchange data a device should register first. This registration could be performed by a "Registrar" device or a cloud service provided by the organization to facilitate autonomic mechanisms. The new device sends its own credentials to the Registrar, and after successful authentication, receives domain information, to enable subsequent enrolment to the domain. The Registrar sends all required information: a device name, domain name, plus some parameters for the operation. Measurement data should be exchanged signed and encripted among devices since these data could carry sensible information about network infrastructures. Some attacks should be considering when analyzing the security of the autonomic solution. Denial of service (DoS) attacks could be performed if the solution be tempered to active more local probe than the available resources allow. Besides that, results could be forged by a device (attacker) in order to this device be considered peer of a specific device (target). This could be done to gain information about a network.

13. References

13.1. Normative References

[draft-anima-boot]
Pritikin, M., Richardson, M., Behringer, M., and S. Bjarnason, "draft-ietf-anima-bootstrapping-keyinfra", draft-ietf-anima-bootstrapping-keyinfra-03 (work in progress), June 2016.

[draft-ietf-ippm-6man-pdm-option]
Elkins, N., Hamilton, R., and M. Ackermann, "draft-ietf-ippm-6man-pdm-option", draft-ietf-ippm-6man-pdm-option-06 (work in progress), September 2016.

[P2PBNM-Nobre-2012]
Nobre, J., Granville, L., Clemm, A., and A. Prieto, "Decentralized Detection of SLA Violations Using P2P Technology, 8th International Conference Network and Service Management (CNSM)", 2012, <http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6379997>.

[RFC4656] Shalunov, S., Teitelbaum, B., Karp, A., Boote, J., and M. Zekauskas, "A One-way Active Measurement Protocol (OWAMP)", RFC 4656, DOI 10.17487/RFC4656, September 2006, <http://www.rfc-editor.org/info/rfc4656>.

[RFC5357] Hedayat, K., Krzanowski, R., Morton, A., Yum, K., and J. Babiarz, "A Two-Way Active Measurement Protocol (TWAMP)", RFC 5357, DOI 10.17487/RFC5357, October 2008, <http://www.rfc-editor.org/info/rfc5357>.

[RFC6812] Chiba, M., Clemm, A., Medley, S., Salowey, J., Thombare, S., and E. Yedavalli, "Cisco Service-Level Assurance Protocol", RFC 6812, DOI 10.17487/RFC6812, January 2013, <http://www.rfc-editor.org/info/rfc6812>.

[RFC7297] Boucadair, M., Jacquenet, C., and N. Wang, "IP Connectivity Provisioning Profile (CPP)", RFC 7297, DOI 10.17487/RFC7297, July 2014, <http://www.rfc-editor.org/info/rfc7297>.

13.2. Informative References

[RFC3954] Claise, B., Ed., "Cisco Systems NetFlow Services Export Version 9", RFC 3954, DOI 10.17487/RFC3954, October 2004, <http://www.rfc-editor.org/info/rfc3954>.

[RFC4148] Stephan, E., "IP Performance Metrics (IPPM) Metrics Registry", BCP 108, RFC 4148, DOI 10.17487/RFC4148, August 2005, <http://www.rfc-editor.org/info/rfc4148>.

Authors' Addresses

Jeferson Campos Nobre
University of Vale do Rio dos Sinos
Porto Alegre
Brazil

Email: jcnobre@unisinos.br

Lisandro Zambenedetti Granville
Federal University of Rio Grande do Sul
Porto Alegre
Brazil

Email: granville@inf.ufrgs.br

Alexander Clemm
Sympotech
Los Gatos
USA

Email: alex@sympotech.com

Alberto Gonzalez Prieto
Cisco Systems
San Jose
USA

Email: albertgo@cisco.com

Network Management Research Group J. Nobre
Internet-Draft University of Vale do Rio dos Sinos
Intended status: Informational L. Granville
Expires: May 19, 2018 Federal University of Rio Grande do Sul
A. Clemm
Huawei
A. Gonzalez Prieto
VMware
November 15, 2017

Autonomic Networking Use Case for Distributed Detection of SLA Violations
draft-irtf-nmrg-autonomic-sla-violation-detection-13

Abstract

This document describes an experimental use case for autonomic networking concerning monitoring of Service Level Agreements (SLAs). The use case aims to detect violations of SLAs in a distributed fashion, striving to optimize and dynamically adapt the autonomic deployment of active measurement probes in a way that maximizes the likelihood of detecting service level violations with a given resource budget to perform active measurements, and is able to do so without any outside guidance or intervention.

This document is a product of the IRTF Network Management Research Group (NMRG). It is published for informational purposes.

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 https://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 19, 2018.

Table of Contents

1. Introduction

The Internet has been growing dramatically in terms of size, capacity, and accessibility in the last years. Communication requirements of distributed services and applications running on top of the Internet have become increasingly demanding. Some examples are real-time interactive video or financial trading. Providing such services involves stringent requirements in terms of acceptable latency, loss, or jitter.

Performance requirements lead to the articulation of Service Level Objectives (SLOs) which must be met. Those SLOs are part of Service Level Agreements (SLAs) that define a contract between the provider and the consumer of a service. SLOs, in effect, constitute a service

level guarantee that the consumer of the service can expect to receive (and often has to pay for). Likewise, the provider of a service needs to ensure that the service level guarantee and associated SLOs are met. Some examples of clauses that relate to service level objectives can be found in [RFC7297]).

Violations of SLOs can be associated with significant financial loss, which can by divided into two categories. For one, there is the loss that can be incurred by the user of a service when the agreed service levels are not provided. For example, a financial brokerage's stock orders might suffer losses when it is unable to execute stock transactions in a timely manner. An electronic retailer may lose customers when their online presence is perceived by customers as sluggish. An online gaming provider may not be able to provide fair access to online players, resulting in frustrated players who are lost as customers. In each case, the failure of a service provider to meet promised service level guarantees can have a substantial financial impact on users of the service. By the same token, there is the loss that is incurred by the provider of a service who is unable to meet promised service level objectives. Those losses can take several forms, such as penalties for not meeting the service and, in many cases more important, loss of revenue due to reduced customer satisfaction. Hence, service level objectives are a key concern for the service provider. In order to ensure that SLOs are not being violated, service levels need to be continuously monitored at the network infrastructure layer in order to know, for example, when mitigating actions need to be taken. To that end, service level measurements must take place.

Network measurements can be performed using active or passive measurement techniques. In passive measurements, production traffic is observed and no monitoring traffic is created by the measurement process itself. That is, network conditions are checked in a non intrusive way. In the context of IP Flow Information eXport (IPFIX), several documents were produced that define how to export data associated with flow records, i.e. data that is collected as part of passive measurement mechanisms, generally applied against flows of production traffic (e.g., [RFC7011]). In addition, it would be possible to collect real data traffic (not just summarized flow records) with time-stamped packets, possibly sampled (e.g., per [RFC5474], as a means of measuring and inferring service levels. Active measurements, on the other hand, are more intrusive to the network in the sense that it involves injecting synthetic test traffic into the network to measure network service levels, as opposed to simply observing production traffic. The IP Performance Metrics (IPPM) WG produced documents that describe active measurement mechanisms, such as: One-Way Active Measurement Protocol (OWAMP) [RFC4656], Two-Way Active Measurement Protocol (TWAMP) [RFC5357], and

Cisco Service Level Assurance Protocol (SLA) [RFC6812]. In addition, there are some mechanisms that do not cleanly fit into either active or passive categories, such as Performance and Diagnostic Metrics Destination Option (PDM) techniques [RFC8250].

Active measurement mechanisms offer a high level of control of what and how to measure. They do not require inspecting production traffic. Because of this, active measurements usually offer better accuracy and privacy than passive measurement mechanisms. Traffic encryption and regulations that limit the amount of payload inspection that can occur are non-issues. Furthermore, active measurement mechanisms are able to detect end-to-end network performance problems in a fine-grained way (e.g., simulating the traffic that must be handled considering specific Service Level Objectives - SLOs). As a result, active measurements are often preferred over passive measurement for SLA monitoring. Measurement probes must be hosted in network devices and measurement sessions must be activated to compute the current network metrics (e.g., considering those described in [RFC4148]). This activation should be dynamic in order to follow changes in network conditions, such as those related with routes being added or new customer demands.

While offering many advantages, active measurements are expensive in terms of network resource consumption. Active measurements generally involve measurement probes that generate synthetic test traffic that is directed at a responder. The responder needs to timestamp test traffic it receives and reflect it back to the originating measurement probe. The measurement probe subsequently processes the returned packets along with time stamping information in order to compute service levels. Accordingly, active measurements consume substantial CPU cycles as well as memory of network devices to generate and process test traffic. In addition, synthetic traffic increases network load. Active measurements thus compete for resources with other functions, including routing and switching.

The resources required and traffic generated by the active measurement sessions are to a large part a function of the number of measured network destinations. (In addition, the amount of traffic generated for each measurement plays a role, which in turn influences the accuracy of the measurement.) The more destinations are being measured, the larger the amount of resources consumed and traffic needed to perform the measurements. Thus, to have a better monitoring coverage it is necessary to deploy more sessions which consequently increases consumed resources. Otherwise, enabling the observation of just a small subset of all network flows can lead to an insufficient coverage.

Furthermore, while some end-to-end service levels can be determined by adding up the service levels observed across different path segments, the same is not true for all service levels. For example, the end-to-end delay or packet loss from a node A to a node C routed via a node B can often be computed simply by adding delays (or loss) from A to B, and B to C. This allows to decompose a large set of end-to-end measurements into a much smaller set of segment measurements. However, end-to-end jitter and (for example) Mean Opinion Scores cannot be decomposed as easily and, for higher accuracy, must be measured end-to-end.

Hence, the decision how to place measurement probes becomes an important management activity. The goal is to obtain maximum benefits of service level monitoring with a limited amount of measurement overhead. Specifically, the goal is to maximize the number of service level violations that are detected with a limited amount of resources.

The use case and the solution approach described in this document address an important practical issue. They are intended to provide a basis for further experimentation to lead into solutions for wider deployment. This document represents the consensus of the IRTF's Network Management Research Group (NMRG). It was discussed extensively and received three separate in-depth reviews.

2. Definitions and Acronyms

Active Measurements: Techniques to measure service levels that involve generating and observing synthetic test traffic

Passive Measurements: Techniques used to measure service levels based on observation of production traffic

AN: Autonomic Network; a network containing exclusively autonomic nodes, requiring no configuration and deriving all required information through self-knowledge, discovery, or intent.

Autonomic Service Agent (ASA): An agent implemented on an autonomic node that implements an autonomic function, either in part (in the case of a distributed function, as in the context of this document), or whole.

Measurement Session: A communications association between a Probe and a Responder used to send and reflect synthetic test traffic for active measurements

Probe: The source of synthetic test traffic in an active measurement

Responder: The destination for synthetic test traffic in an active measurement

SLA: Service Level Agreement

SLO: Service Level Objective

P2P: Peer-to-Peer

(Note: definitions of AN and ASA are borrowed from [RFC7575]).

3. Current Approaches

The current best practice in feasible deployments of active measurement solutions to distribute the available measurement sessions along the network consists in relying entirely on the human administrator expertise to infer which would be the best location to activate such sessions. This is done through several steps. First, it is necessary to collect traffic information in order to grasp the traffic matrix. Then, the administrator uses this information to infer which are the best destinations for measurement sessions. After that, the administrator activates sessions on the chosen subset of destinations considering the available resources. This practice, however, does not scale well because it is still labor intensive and error-prone for the administrator to determine which sessions should be activated given the set of critical flows that needs to be measured. Even worse, this practice completely fails in networks whose critical flows are too short in time and dynamic in terms of traversing network path, like in modern cloud environments. That is so because fast reactions are necessary to reconfigure the sessions and administrators are just not quick enough in computing and activating the new set of required sessions every time the network traffic pattern changes. Finally, the current active measurements practice usually covers only a fraction of the network flows that should be observed, which invariably leads to the damaging consequence of undetected SLA violations.

4. Use Case Description

The use case involves a service level provider who needs to monitor the network to detect service level violations using active service level measurements, and wants to be able to do so with minimal human intervention. The goal is to conduct the measurements in an effective manner maximizing the percentage of detected service level violations. The service level provider has a bounded resource budget with regards to measurements that can be performed, specifically, with regards to the number of measurements that can be conducted concurrently from any one network device, and possibly with regards

to the total amount of measurement traffic on the network. However, while at any one point in time the number of measurements conducted is limited, it is possible for a device to change which destinations to measure over time. This can be exploited to achieve a balance of eventually covering all possible destinations using a reasonable amount of "sampling" where measurement coverage of a destination cannot be continuous. The solution needs to be dynamic and be able to cope with network conditions which may change over time. The solution should also be embeddable inside network devices that control the deployment of active measurement mechanisms.

The goal is to conduct the measurements in a smart manner that ensures that the network is broadly covered and the likelihood of detecting service level violations is maximized. In order to maximize that likelihood, it is reasonable to focus measurement resources on destinations that are more likely to incur a violation, while spending less resources on destinations that are more likely to be in compliance. In order to do so, there are various aspects that can be exploited, including past measurements (destinations close to a service level threshold requiring more focus than destinations further from it), complementation with passive measurements such as flow data (to identify network destinations that are currently popular and critical), and observations from other parts of the network. In addition, measurements can be coordinated among different network devices to avoid hitting the same destination at the same time and to be able to share results that may be useful in future probe placement.

Clearly, static solutions will have severe limitations. At the same time, human administrators cannot be in the loop for continuous dynamic measurement probe reconfigurations. Accordingly, an automated or, ideally, autonomic solution is needed in which network measurements are automatically orchestrated and dynamically reconfigured from within the network. This can be accomplished using an autonomic solution that is distributed, using Autonomic Service Agents that are implemented on nodes in the network.

5. A Distributed Autonomic Solution

The use of Autonomic Networking (AN) [RFC7575] can help such detection through an efficient activation of measurement sessions. Such an approach, along with a detailed assessment confirming its viability, has been described [P2PBNM-Nobre-2012]. The problem to be solved by AN in the present use case is how to steer the process of Measurement Session activation by a complete solution that sets all necessary parameters for this activation to operate efficiently, reliably and securely, with no required human intervention other than setting overall policy.

When a node first comes online, it has no information about which measurements are more critical than others. In the absence of information about past measurements and information from measurement peers, it may start with an initial set of measurement sessions, possibly randomly seeding a set of starter measurements, perhaps taking a round robin approach for subsequent measurement rounds. However, as measurements are collected, a node will gain increasing information that it can utilize to refine its strategy of selecting measurement targets going forward. For one, it may take note of which targets returned measurement results very close to service level thresholds that may therefore require closer scrutiny compared to others. Second, it may utilize observations that are made by its measurement peers in order to conclude which measurement targets may be more critical than others, and in order to ensure that proper overall measurement coverage is obtained (so that not every node incidentally measure the same targets, while other targets are not measured at all).

We advocate for embedding Peer-to-Peer (P2P) technology in network devices in order to conduct the Measurement Session activation decisions using autonomic control loops. Specifically, we advocate for network devices to implement an autonomic function to monitor service levels for violations of service level objectives, determining which Measurement Sessions to set up at any given point in time based on current and past observations of the node, and of other peer nodes.

By performing these functions locally and autonomically on the device itself, which measurements to conduct can be modified quickly based on local observations while taking local resource availability into account. This allows a solution to be more robust and react more dynamically to rapidly changing service levels than a solution that has to rely on central coordination. However, in order to optimize decisions which measurements to conduct, a node will need to communicate with other nodes. This allows a node to take into account other nodes' observations in addition to its own in its decisions.

For example, remote destinations whose observed service levels are on the verge of violating stated objectives may require closer monitoring than remote destinations that are comfortably within a range of tolerance. It also allows nodes to coordinate their probing decisions to collectively achieve the best possible measurement coverage. As the amount of resources available for monitoring and for exchange of measurement data and coordination with other nodes are limited, a node may further be interested in identifying other nodes whose observations are most similar to and correlated with its own. This helps a node prioritize and guide with which other nodes

to primarily coordinate and exchange data with. All of this requires the use of a P2P overlay.

A P2P overlay is essential for several reasons:

- It makes it possible for nodes (respectively Autonomic Service Agents that are deployed on those nodes) in the network to autonomically set up Measurement Sessions, without having to rely on central management system or controller to perform configuration operations associated with configuring measurement probes and responders.

- It facilitates the exchange of data between different nodes to share measurement results so that each node can refine its measurement strategy based not just its own observations, but observations from its peers.

- It allows nodes to coordinate their measurements to obtain the best possible test coverage and avoid measurements that have a very low likelihood of detecting service level violations.

The provisioning of the P2P overlay should be transparent for the network administrator. An Autonomic Control Plane such as defined in [I-D.anima-autonomic-control-plane] provides an ideal candidate for the P2P overlay to run on.

An autonomic solution for the distributed detection of SLA violations provide several benefits. First, efficiency: this solution should optimize the resource consumption and avoid resource starvation on the network devices. A device that is "self-aware" of its available resources will be able to adjust measurement activities rapidly as needed, without requiring a separate control loop involving resource monitoring by an external system. Secondly, placing logic where to conduct measurements in the node enables rapid control loops in which devices are able to react instantly to observations and adjust their measurement strategy. For example, a device could decide to adjust the amount of synthetic test traffic being sent during the measurement itself depending on results observed so far on this and on other concurrent measurement sessions. As a result, the solution could decrease the time necessary to detect SLA violations. Adaptivity features of an autonomic loop could capture faster the network dynamics than an human administrator and even a central controller. Finally, the solution could help to reduce the workload of human administrator, or, at least, to avoid their need to perform operational tasks.

In practice, these factors combine to maximize the likelihood of SLA violations being detected while operating within a given resource

budget, allowing to conduct a continuous measurement strategy that takes into account past measurement results, observations of other measures such as link utilization or flow data, sharing of measurement results between network devices, and coordinating future measurement activities among nodes. Combined this can result in efficient measurement decisions that achieve a golden balance between broad network coverage and honing in on service level "hot spots".

6. Intended User Experience

The autonomic solution should not require any human intervention in the distributed detection of SLA violations. By virtue of the solution being autonomic, human users will not have to plan which measurements to conduct in a network, often a very labor intensive task today that requires detailed analysis of traffic matrices and network topologies and is not prone to easy dynamic adjustment. Likewise, they will not have to configure measurement probes and responders.

There are some ways in which a human administrator may still interact with the solution. For one, the human administrator will of course be notified and obtain reports about service level violations that are observed. Second, a human administrator may set a policies regarding how closely to monitor the network for service level violations and how many resources to spend. For example, an administrator may set a resource budget that is assigned to network devices for measurement operations. With that given budget, the number of SLO violations that are detected will be maximized. Alternatively, an administrator may set a target for the percentage of SLO violations that must be detected, i.e. a target for the ratio between the number of detected SLO violations, and the number of total SLO violations that are actually occurring (some of which might go undetected). In that case, the solution will aim to minimize the resources spent (i.e. the amount of test traffic and Measurement Sessions) that are required to achieve that target.

7. Implementation Considerations

The active measurement model assumes that a typical infrastructure will have multiple network segments and Autonomous Systems (ASs), and a reasonably large number of routers. It also considers that multiple SLOs can be in place at a given time. Since interoperability in a heterogenous network is a goal, features found on different active measurement mechanisms (e.g. OWAMP, TWAMP, and IPSLA) and device programability interfaces (such as Juniper’s Junos API or Cisco’s Embedded Event Manager) could be used for the implementation. The autonomic solution should include and/or reference specific algorithms, protocols, metrics and technologies

for the implementation of distributed detection of SLA violations as a whole.

Finally, it should be noted that there are multiple deployment scenarios, including deployment scenarios that involve physical devices hosting autonomic functions, or virtualized infrastructure hosting the same. Co-deployment in conjunction with Virtual Network Functions (VNF) is a possibility for further study.

7.1. Device Based Self-Knowledge and Decisions

Each device has self-knowledge about the local SLA monitoring. This could be in the form of historical measurement data and SLOs. Besides that, the devices would have algorithms that could decide which probes should be activated in a given time. The choice of which algorithm is better for a specific situation would be also autonomic.

7.2. Interaction with other devices

Network devices should share information about service level measurement results. This information can speed up the detection of SLA violations and increase the number of detected SLA violations. For example, if one device detects that a remote destination is in danger of violating an SLO, other devices may conduct additional measurements to the same destination or other destinations in its proximity. For any given network device, the exchange of data may be more important with some devices (for example, devices in the same network neighborhood, or devices that are "correlated" by some other means) than with others. The definition of network devices that exchange measurement data, i.e., management peers, creates a new topology. Different approaches could be used to define this topology (e.g., correlated peers [P2PBNM-Nobre-2012]). To bootstrap peer selection, each device should use its known endpoints neighbors (e.g., FIB and RIB tables) as the initial seed to get possible peers. It should be noted that a solution will benefit if topology information and network discovery functions are provided by the underlying autonomic framework. A solution will need to be able to discover measurement peers as well as measurement targets, specifically measurement targets that support active measurement responders and which will be able to respond to measurement requests and reflect measurement traffic as needed.

8. Comparison with current solutions

There is no standardized solution for distributed autonomic detection of SLA violations. Current solutions are restricted to ad hoc scripts running on a per node fashion to automate some

administrator's actions. There are some proposals for passive probe activation (e.g., DECON and CSAMP), but without the focus on autonomic features.

9. Related IETF Work

The following paragraphs discuss related IETF work and are provided for reference. This section is not exhaustive, rather it provides an overview of the various initiatives and how they relate to autonomic distributed detection of SLA violations.

1. [LMAP]: The Large-Scale Measurement of Broadband Performance Working Group aims at the standards for performance management. Since their mechanisms also consist in deploying measurement probes the autonomic solution could be relevant for LMAP specially considering SLA violation screening. Besides that, a solution to decrease the workload of human administrators in service providers is probably highly desirable.

2. [IPFIX]: IP Flow Information EXport (IPFIX) aims at the process of standardization of IP flows (i.e., netflows). IPFIX uses measurement probes (i.e., metering exporters) to gather flow data. In this context, the autonomic solution for the activation of active measurement probes could be possibly extended to address also passive measurement probes. Besides that, flow information could be used in the decision making of probe activation.

3. [ALTO]: The Application Layer Traffic Optimization Working Group aims to provide topological information at a higher abstraction layer, which can be based upon network policy, and with application-relevant service functions located in it. Their work could be leveraged for the definition of the topology regarding the network devices which exchange measurement data.

10. Acknowledgements

We wish to acknowledge the helpful contributions, comments, and suggestions that were received from Mohamed Boucadair, Brian Carpenter, Hanlin Fang, Bruno Klauser, Diego Lopez, Vincent Roca, and Eric Voit. In addition, we thank Diego Lopez, Vincent Roca, and Brian Carpenter for their detailed reviews.

11. IANA Considerations

This memo includes no request to IANA.

12. Security Considerations

Security of the solution hinges on the security of the network underlay, i.e. the Autonomic Control Plane. If the Autonomic Control Plane were to be compromised, an attacker could undermine the effectiveness of measurement coordination by reporting fraudulent measurement results to peers. This would cause measurement probes to be deployed in an ineffective manner that would increase the likelihood that violations of service level objectives go undetected.

Likewise, security of the solution hinges on the security of the deployment mechanism for autonomic functions, in this case, the autonomic function that conducts the service level measurements. If an attacker were able to hijack an autonomic function, it could try to exhaust or exceed the resources that should be spent on autonomic measurements in order to deplete network resources, including network bandwidth due to higher-than-necessary volumes of synthetic test traffic generated by measurement probes. Again, it could also lead to reporting of misleading results, among other things resulting in non-optimal selection of measurement targets and in turn an increase in the likelihood that service level violations go undetected.

13. Informative References

[draft-anima-boot]
Pritikin, M., Richardson, M., Behringer, M., Bjarnason, S., and K. Watsen, "draft-ietf-anima-bootstrapping-keyinfra", draft-ietf-anima-bootstrapping-keyinfra-08 (work in progress), October 2017.

[I-D.anima-autonomic-control-plane]
Behringer, M., Eckert, T., and S. Bjarnason, "An Autonomic Control Plane", draft-ietf-anima-autonomic-control-plane-12 (work in progress), October 2017.

[P2PBNM-Nobre-2012]
Nobre, J., Granville, L., Clemm, A., and A. Gonzalez Prieto, "Decentralized Detection of SLA Violations Using P2P Technology, 8th International Conference Network and Service Management (CNSM)", 2012, <http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6379997>.

[RFC4148] Stephan, E., "IP Performance Metrics (IPPM) Metrics Registry", BCP 108, RFC 4148, DOI 10.17487/RFC4148, August 2005, <https://www.rfc-editor.org/info/rfc4148>.

[RFC4656] Shalunov, S., Teitelbaum, B., Karp, A., Boote, J., and M. Zekauskas, "A One-way Active Measurement Protocol (OWAMP)", RFC 4656, DOI 10.17487/RFC4656, September 2006, <https://www.rfc-editor.org/info/rfc4656>.

[RFC5357] Hedayat, K., Krzanowski, R., Morton, A., Yum, K., and J. Babiarz, "A Two-Way Active Measurement Protocol (TWAMP)", RFC 5357, DOI 10.17487/RFC5357, October 2008, <https://www.rfc-editor.org/info/rfc5357>.

[RFC5474] Duffield, N., Ed., Chiou, D., Claise, B., Greenberg, A., Grossglauser, M., and J. Rexford, "A Framework for Packet Selection and Reporting", RFC 5474, DOI 10.17487/RFC5474, March 2009, <https://www.rfc-editor.org/info/rfc5474>.

[RFC6812] Chiba, M., Clemm, A., Medley, S., Salowey, J., Thombare, S., and E. Yedavalli, "Cisco Service-Level Assurance Protocol", RFC 6812, DOI 10.17487/RFC6812, January 2013, <https://www.rfc-editor.org/info/rfc6812>.

[RFC7011] Claise, B., Ed., Trammell, B., Ed., and P. Aitken, "Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information", STD 77, RFC 7011, DOI 10.17487/RFC7011, September 2013, <https://www.rfc-editor.org/info/rfc7011>.

[RFC7297] Boucadair, M., Jacquenet, C., and N. Wang, "IP Connectivity Provisioning Profile (CPP)", RFC 7297, DOI 10.17487/RFC7297, July 2014, <https://www.rfc-editor.org/info/rfc7297>.

[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, <https://www.rfc-editor.org/info/rfc7575>.

[RFC8250] Elkins, N., Hamilton, R., and M. Ackermann, "IPv6 Performance and Diagnostics Metrics (PDM) Destination Option", RFC 8250, October 2017.

Authors' Addresses

Jeferson Campos Nobre
University of Vale do Rio dos Sinos
Porto Alegre
Brazil

Email: jcnobre@unisinos.br

Lisandro Zambenedetti Granville
Federal University of Rio Grande do Sul
Porto Alegre
Brazil

Email: granville@inf.ufrgs.br

Alexander Clemm
Huawei
Santa Clara, California
USA

Email: ludwig@clemm.org

Alberto Gonzalez Prieto
VMware
Palo Alto, California
USA

Email: agonzalezpri@vmware.com