ETSI GS NGP 002 V1.1.1 (2017-05)

Next Generation Protocols (NGP); Self-Organizing Control and Management Planes

Disclaimer

The present document has been produced and approved by the Next Generation Protocols (NGP) ETSI Industry Specification Group (ISG) and represents the views of those members who participated in this ISG.
It does not necessarily represent the views of the entire ETSI membership.

Reference
DGS/NGP-002

Keywords
management plane, next generation protocol,
self-management, self-organizing

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Contents

Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (https://ipr.etsi.org/).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Foreword

This Group Specification (GS) has been produced by ETSI Industry Specification Group (ISG) Next Generation Protocols (NGP).

Modal verbs terminology

In the present document "**shall**", "**shall not**", "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the ETSI Drafting Rules (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

1 Scope

The scope of the present document is to specify the self-organizing control and management planes for the Next Generation Protocols (NGP), Industry Specific Group (ISG).

2 References

2.1 Normative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

Referenced documents which are not found to be publicly available in the expected location might be found at https://docbox.etsi.org/Reference.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are necessary for the application of the present document.

[1] IETF RFC 7575: "Autonomic Networking: Definitions and Design Goals".

2.2 Informative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

[i.1] ETSI GS AFI 002 (V1.1.1): "Autonomic network engineering for the self-managing Future Internet (AFI); Generic Autonomic Network Architecture (An Architectural Reference Model for Autonomic Networking, Cognitive Networking and Self-Management)".

[i.2] IETF draft-ietf-anima-reference-model: "Reference Model for Autonomic Networking", April 2016.

[i.3] IETF draft-pritikin-anima-bootstrapping-keyinfra: "Bootstrapping Key Infrastructures", October 2016.

[i.4] IETF draft-ietf-anima-grasp: "Generic Autonomic Signaling Protocol (GRASP)", December 2016.

[i.5] ETSI TR 121 905: "Digital cellular telecommunications system (Phase 2+) (GSM); Universal Mobile Telecommunications System (UMTS); LTE; Vocabulary for 3GPP Specifications (3GPP TR 21.905)".

[i.6] ETSI TS 132 501: "Universal Mobile Telecommunications System (UMTS); LTE; Telecommunication management; Self-configuration of network elements; Concepts and requirements (3GPP TS 32.501)".

[i.7] ETSI GS NGP 006: "Next Generation Protocol (NGP); Intelligence-defined Network".

[i.8] NTECH(17)000013: "Requirements for Protocols and APIs for Enabling GANA based Autonomics, Cognitive Networking and Self-Management of Networks and Services in Evolving and Future Networks".

[i.9] IETF RFC 4192: "Procedures for Renumbering an IPv6 Network without a Flag Day".

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the following terms and definitions apply:

NOTE: Some definitions are inherited from IETF RFC 7575 [1].

autonomic node: network node that supports a set of basic Self-organizing functions

NOTE: E.g. the ASAs for basic connectivity as described in clause 9.1.

autonomic domain: set of autonomic nodes compose a domain within which the autonomic node could create stable connectivity with each other and share the same intent

autonomic service agent: agent implemented on an autonomic node that implements an autonomic function, either in part (in the case of a distributed function) or whole (IETF RFC 7575 [1])

intent: abstract, high-level policy used to operate the network

NOTE: Its scope is an autonomic domain (IETF RFC 7575 [1])

network-wide knowledge: valuable information extracted from the data in various nodes or some network-level policies/information input from the administrators

self-X Network: network supports a set of "Self-" features such as Self-Configuration, Self-Orchestration, etc. (as described in clause 6) to form a self-organizing control and management plane

3.2 Abbreviations

For the purposes of the present document, the abbreviations given in ETSI TR 121 905 [i.5] and the following apply:

NOTE: Should apply to scenarios that include mobile network architectures.

ACO	Automatic Cluster Optimization
ACP	Autonomic Control Plane
AFI	Autonomic Future Internet
AI	Artificial Intelligence
AN	Autonomic Network
ANI	Autonomic Networking Infrastructure
API	Application Programming Interface
ASA	Autonomic Service Agent
ASG	Aggregation Site Gateway
BRSKI	Bootstrapping Remote Secure Key Infrastructures
BSS	Business Support System
CA	Certificate Authority
CSG	Cell Site Gateway
DE	Decision Element
DHCP	Dynamic Host Configuration Protocol
ECMP	Equal-cost Multi-path Routing
FCAPS	Fault, Configuration, Accounting, Performance, Security
GANA	Generic Autonomic Networking Architecture
GRASP	Generic Autonomic Signalling Protocol
IETF	Internet Engineering Task Force
IGP	Interior Gateway Protocol
IP	Internet Protocol
IPRAN	IP-based Radio Access Network
IRP	Integration Reference Point

ISG	Industry Specific Group
ISIS	Intermediate System to Intermediate System
ISP	Internet Service Provider
MPLS	Multi-Protocol Label Switching
ND	Neighbour Discovery
NGP	Next Generation Protocols
NMS	Network Management System
OAM	Operations, Administration, and Maintenance
OSPF	Open Shortest Path First
OSS	Operation Support System
PW	Pseudo-Wire
RQ	Requirement
RSG	Radio Service Gateway
SDN	Software Defined Network
SLAAC	Stateless Address Autoconfiguration
SMN	Self-Managed Network
SON	Self-Organizing Networks
SXN	Self-X Network
TE	Traffic Engineering
ULA	Unique Local Address
VPN	Virtual Private Network

4 Overview

The ISG Next Generation Protocols (NGP) aims to review the future landscape of Internet Protocols, identify and document future requirements and trigger follow up activities to drive a vision of a considerably more efficient Internet that is far more attentive to user demand and more responsive whether towards humans, machines or things.

A measure of the success of NGP would be to remove historic sub-optimized IP protocol stacks and allow all next generation networks to inter-work in a way that accelerates a post-2020 connected world unencumbered by past developments.

The NGP ISG is foreseen as having a transitional nature that is a vehicle for the 5G community and other related communications markets to first gather their thoughts together and prepare the case for the Internet community's engagement in a complementary and synchronized modernization effort.

Therefore NGP ISG aims to stimulate closer cooperation over standardization efforts for generational changes in communications and networking technology.

The present document introduces the NGP, ISG view on how the network could get self-managed, through interaction between devices based on a set of new protocols. One important principle is taking an incremental approach that the Self-X Networks (SXN) should co-exist and interact with current network.

The present document presents the vision of Self-Managed Networks in clause 6. The vision is separated into several goals, which are mainly inherited from the classic FCAPS model. In clause 7, the present document discusses a couple of important principles of designing the SXN.

Then, according to clause 6 and clause 7, the architecture of SXN is introduced in clause 8. The architecture is angled from a node perspective; and the node is called Autonomic Node (AN). The essential component in an AN is the Autonomic Service Agent (ASA), which could be considered as applications running in network devices to fulfil specific network management functions/tasks without human intervene. There could be various kinds of ASAs to fulfil different functions/tasks; some ASAs, which are considered as basic and common functions in a network, are introduced in clause 9.

There are also two use cases of the proposed architecture introduced in clause 10. At last, a summary of future protocol requirements are documented in clause 11.

5 Background

5.1 Motivation of Self-Management and Control

The success of the Internet has made IP-based networks bigger and more complex. The scale of networks is quickly increasing; the numbers of network devices are also quickly increasing. With the increasing of new features and functionalities, network devices has been becoming more and more complicated and new network services have been continuing emerging. Network controlling is becoming more multidimensional, beyond the routing reachability. Diversified network management requirements are growing while the granularity of network management is required to be finer and more precise. The controlled and managed objectives in the network have complicated relationships, which have not yet been considered. The cooperation and interference among devices are complicated.

In the current IP based network systems, only routing functions may be considered as autonomic. Even that requires manual provisioning of peer neighbours, route policies and other attributes to achieve the desired effect. It results in a rigid network traffic management. Although several network management tools can automate repeatable work through scripts, the overall network operations, control and management functions still require human intelligence and experts with in-depth knowledge of all aspects.

Currently, the network controlling and management are mostly through the device parameters, which rely on the decision and implement of network administrator. With the growing network infrastructures, network changes are more frequent and impact vast geographies. A network administrator needs to modify configurations as often and in timely manner. However, manual verification and validation processes are usually slow, painstaking and still error prone. It is reported that most network problems (above 95 %) are caused by human's mis-configurations. The network administrator is both the key and the bottleneck, even the source of problem.

All of the abovementioned situations are extremely demanding for dynamic management that needs to response to a large amount of information. Human based management are not able to meet the requirements any more. A more flexible, extensible and self-managing system is urgently needed. A completely automatic solution for network control and management could simplify human management, avoid human errors, and reduce the cost of network maintenance.

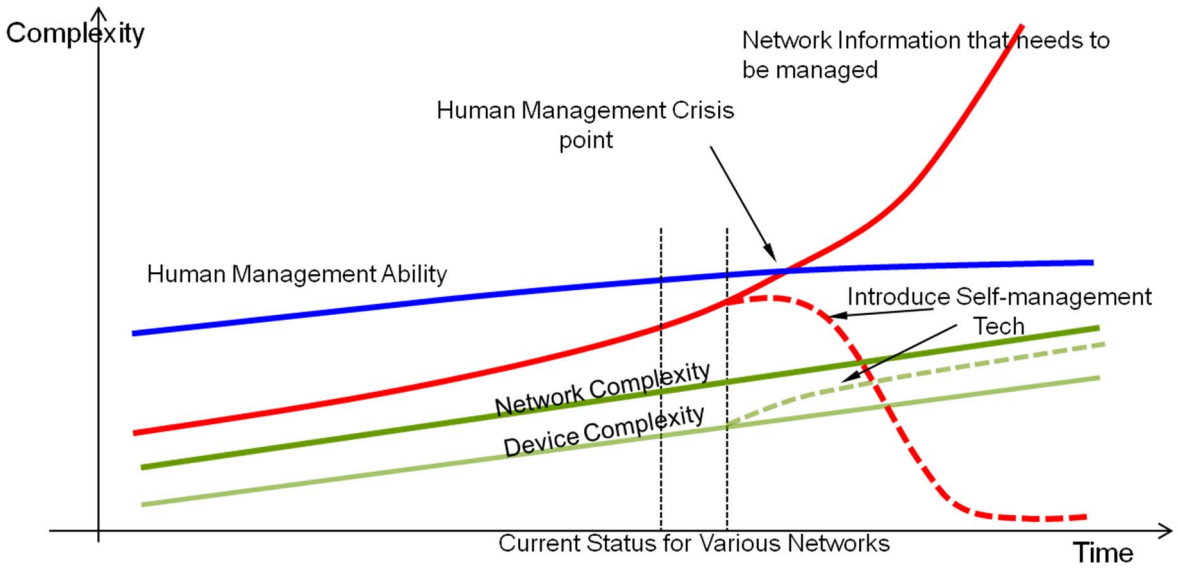

Figure 1: Trend of Network Complexity and Impact of Introducing Self-managing Technologies

5.2 Evolution History of Network Control and Management

IP networking was initially designed with similar properties in mind. An IP network should be distributed and redundant to withstand outages in any part of the network. Routing protocols such as OSPF and ISIS exhibit properties of self-management and can thus be considered autonomic in the definition of the present document.

However, as IP networking evolved, the ever-increasing intelligence of network elements was often not put into protocols to follow this paradigm, but was put into external north-to-south configuration systems. This configuration made network elements dependent on some process that manages them, either a human or a network management system, which is still human based with some enhanced tools.

While the network scale keeps increasing, the complexity becomes a bigger issue. There are two diverted opinions regarding to evolving directions:

a) Centralized control and management systems are introduced to ease the management of a large number of devices and largely reduce the inconsistency/conflicts among devices However, centralization does not mean more intelligence; rather, it only stands for the aggregation of information and the management is still essentially relying on the intelligence of the administrators. The network complexity will increase beyond the handling capability of human.

b) Distributed Intelligence should extend to other network aspects beyond the reachability. DHCP and ND are also moving towards this direction, but these two protocols are only deployed in the edge network where the end devices communicate with the network directly. This evolving direction emphasis more on sensing and communication in the horizontal level among the network devices, and it can only deal with very limited management tasks.

In a nutshell, to achieve a more self-managing network without increasing human burden, neither only aggregating information and control centrally, nor simply increasing horizontal communication is enough. The essential thing is that each of the network devices needs to be enhanced with more intelligence.

5.3 Relationship with Existing Work

1) 3GPP SON

3GPP has a set of technologies called "Self-Organizing Network (SON)". In one aspect, 3GPP SON focusing on specific 3GPP systems while the SXN in the present document is more generic; in another aspect, current 3GPP SON is mostly regarding to wireless interface self-optimization while the SXN could be the candidate architecture and technical approaches for the 3GPP SON of the fixed network part.

2) AFI GANA

The AFI (Autonomic network engineering for the self-managing Future Internet) is an ETSI ISG that dedicated for autonomic networking. AFI had launched the GANA (Generic Autonomic Network Architecture) reference model for autonomic networking, cognitive networking and self-management in 2013 (the latest version, ETSI GS AFI 002 [i.1]).

GANA is a very generic and comprehensive model, of which the main objective is "to define, iteratively, a generic, conceptual architectural reference model intended to serve as guideline for the design of the future generation networks exhibiting autonomic characteristics or capabilities", as stated in the GANA document. The SXN essentially keeps consistent with some important concepts in GANA. For example, the Autonomic Node defined in clause 8 is essentially the same with Network Element in GANA; the ASAs described in clause 9 could be considered as specific instances of the GANA Decision Element; the Network-wide Knowledge in clause 8 is a simple instance of Knowledge Plane defined in GANA.

So, in general, the SXN is consistent with the GANA model, and there is no conflict. However, the SXN concepts and technologies are not as generic as GANA; rather, they aim at defining specific components that much more closed to implementation based on current network.

3) IETF Anima

Anima (Autonomic Networking Integrated Model and Approach) is an IETF working group aims at developing protocols/mechanisms that could be directly implemented and integrated into current networks to improve the autonomics. Anima's approach is to identify some very basic and common technical components that could be re-used among different scenarios. These components are called ANI (Autonomic Network Infrastructure). Currently, there are three technologies defined as ANI, as the following.

- GRASP (GeneRic Autonomic Signalling Protocol):

  GRASP is the protocol used between autonomic nodes to cooperate to fulfil management tasks. It provides generic and basic communication schemes such as Discovery, Negotiation, Synchronization and Flood. GRASP is a realization of the Discovery Agent and Information Distribution Agent in clause 9.

- ACP (Autonomic Control Plane):

  ACP enables a secure and stable management channel between autonomic nodes without any manual configuration. ACP is one realization fits into Autonomic Reliable Connectivity Agent in clause 9.

- BRSKI (Bootstrapping Remote Secure Key Infrastructures):

  BRSKI allows new devices joining the Autonomic Domain by authentication of the device certificate; and also makes the new devices assigned with Autonomic Domain certificates for secure communication afterwards. It is a realization of Secure Bootstrap Agent in clause 9.

Overall, Anima provides specific IP-based realization of some functions specified in the present document; but the present document has a more general scope and not binding to IP protocols. The present document only considers Anima as an instance/reference of realizing corresponding self-managing functions.

6 Vision of Self-X Networks

6.1 Overview

This clause describes the high-level goals that are expected to be achieved by the SXN.

According to ISO FCAPS model, network management contains Fault Management, Configuration Management, Accounting Management, Performance Management and Security Management. This classic model is also a very suitable reference for setting up the high-level goals of the SON.

However, this clause excludes the Accounting from the FCAPS; and narrows down the Performance Management and Security Management to Self Optimization and Self Defense. Apart from FCAPS, this clause includes Self Service Orchestration into the scope.

6.2 Self Configuration

Self-configuration means that all the Autonomic Nodes are configured autonomically and dynamically. "Autonomically" means the configuration is done by the node and the network without human intervene; while "dynamically" means the configurations are not static rules but rather generated according to the node and the networks' features and conditions.

The configurations mainly contain two parts:

1) Initial configuration: when a node newly joins in the Self-Managed Network, it needs to get the basic connectivity configurations such as addressing, routing, etc. (clause 9.1 introduces ASAs for this purpose).

2) Service configuration: when the SXN wants to enable a service (e.g. MPLS VPN), it needs to make configurations on a certain of nodes. (clause 9.3 introduces ASAs for this purpose).

Self-configuration could be considered as a very basic yet very important feature in SXN. Since the network can simply begin to run after the Self-configuration.

6.3 Self Service Orchestration

When deploying a service, in a perspective of an Autonomic Node, there could be two approaches:

1) The nodes receive specific configurations which have been sorted out by a central management server or controller, according to the service request. This is also the traditional manner in service orchestration.

2) The nodes directly interpret the service request and sort out the configurations by themselves. When multiple nodes are involved for one service, the ANs should be able to coordinate with each other autonomically.

In SXN, both of the approaches should be utilized. The former one is more suitable for sophisticated service layout where the logic is complex and it is very difficult for the distributed nodes to cooperate; while the latter could be used in the scenarios that does not has much complex logic so that some simple interactions between ANs can fulfil the task.

Whatever approach is taken, the common precondition is that the server/controller and ANs need to understand the service requests and translate them into configurations without human intervene. This requires standardized interface of the service request. More ideally, the service requests should be in an abstract or even nature language which can simplify the users/administrators' burden to deliver the services to the SXN systems.

6.4 Self Fault Management

The Self Fault Management mainly includes two aspects: Self-Fault Discovery and Self Fault Recovery.

For Self Fault Discovery, sometimes it is not only regarding to a single node; and the fault might be very implicit that could not be discovered a single node or even a group of nodes cooperated together. In these cases, sophisticated analysis of logs from different devices might be needed. While in some other scenarios, the fault is more explicit and it could be probably discovered by ANs through some real-time measurement.

The fault could be software fault and hardware fault. Ideally, the SXN should be able to fix all software faults. Hardware fault is basically out of scope of SXN. However, in some cases, the SXN might reduce the impact of the fault by simply isolating the fault devices.

6.5 Self Optimization

In massive scale systems, it is extremely difficult for a system administrator to tune parameters for best possible network performance due to the lack of knowledge or time. A self-optimization module can smartly learn network performance status and optimize the configurations without human intervention.

First of all, the SXN needs to learn about the current network performance. This could be done through measurement/probing by the ANs. Then, the SXN could adjust the network resources allocation, the traffic paths, and/or any other actions that could affect the network performance. After adjusting the configurations, the SXN could again measure/probe the performance to check the effectiveness of the adjustment. Thus, a closed loop is formed.

6.6 Self Defence

The SXN needs to detect network attacks in real-time, and make defence accordingly. Similar to the Self Fault Discovery, some attacks could be detected on a single node; while some might need sophisticated analysis.

Ideally, the SXN should not only be able to defend the known attacks automatically, but also recognize and defend some new attacks. This could be possibly done through learning the network attack behaviours, so that the SXN could extract some come criteria of detecting some kind of attacks.

Similar to Self Fault Recovery, SXN can do self-healing to the attacked devices/resources. It is not easy to fully heal the attack, but there should be a bottom line that isolating compromised or faulty nodes and auto-upgrade of software patches.

7 Considerations for Realizing Self-X Networks

7.1 Request for New Protocols and Enhanced Infrastructure

Towards achieving the SXN vision described above, it is no doubt that there needs to be some new protocols introduced into the network. Behind the protocols, the essential thing is that new functions/features are needed. This is mainly discussed in clause 9.

Although one important principle of SXN is not to take a clean-slate approach, normally one new function cannot be realized only through deploying on a single node (or even a small set of nodes). Thus, most of the functions need more or less support from the infrastructure. Some functions (e.g. the ASAs discussed in clause 9.1) might even require every node in the SXN to support. However, the new functions should be integrated into current devices as easy as possible.

7.2 Distributed and Centralized Approaches

The IP network was originally designed according to a distributed approach, which is mostly represented by the routing protocols. Although there are also some centralized approaches such as network management. The core of the IP network is in general distributed.

On the other hand, the hot trend of the network evolving is SDN (Software Defined Network), which is a typical centralized approach.

These two approaches are not conflict with each other. In SXN, both of the two approaches would be used and should be suitable for different scenarios or use cases respectively.

7.3 Transition Considerations

The SXN should evolve in an incremental way rather than a "flag day" approach or being deployed isolated with current networks. So it is important to remain current network architecture and services not be impact significantly. Pieces of autonomic mechanisms could be added one by one and co-exist with current network, to achieve a smooth transition to SXN.

7.4 Security Considerations

Since there are lots of crucial behaviours of the SXN fully autonomic and without any human monitoring and auditing, the SXN should have stronger security requirements than normal networks, to prevent malicious nodes joining the SXN and making attacks which might also be propagate autonomically.

Major security requirements of SXN are as the following:

1) Access authentication of ANs

   When an AN gets online, the SXN should be able to authenticate the identity of the node. This is a very basic security requirement of SXN, and should be the bottom line.
   A further requirement would be auditing whether the valid identity had been used in another network. This is to prevent valid identity be stolen or misused.

2) Authentication between ANs

   When ANs starting to communicate with each other, they also need to make sure the counterpart is a legal node. Normally, the authentication scope could be based on domain, where the ANs might share a common trust anchor.

3) Encrypt communications between ANs

   The communication between ANs should be encrypting by default. If applicable, the encrypt communication should be separated from normal data plane communications. This is to gain a high reliability that the communication between ANs won't be affected by other traffic. In extreme case, even if the normal data crashes, the ANs communication should also be stable and secure.

4) Authorization of prescriptive behaviour

   Besides identity authentication and encryption, the authorization of some behaviour is also very important, since one AN might receive indication which might contain some behaviours for the node to execute. Some behaviour might have significant impact to the network, so the receiver should make sure the node that sent it the indication has the right to do so.

SXN security mechanisms themselves should also be autonomic as much as possible, as they could be also considered as a set of functions/tasks that shall be fulfilled by the SXN.

8 Architecture of Self-X Network

8.1 Architecture Overview

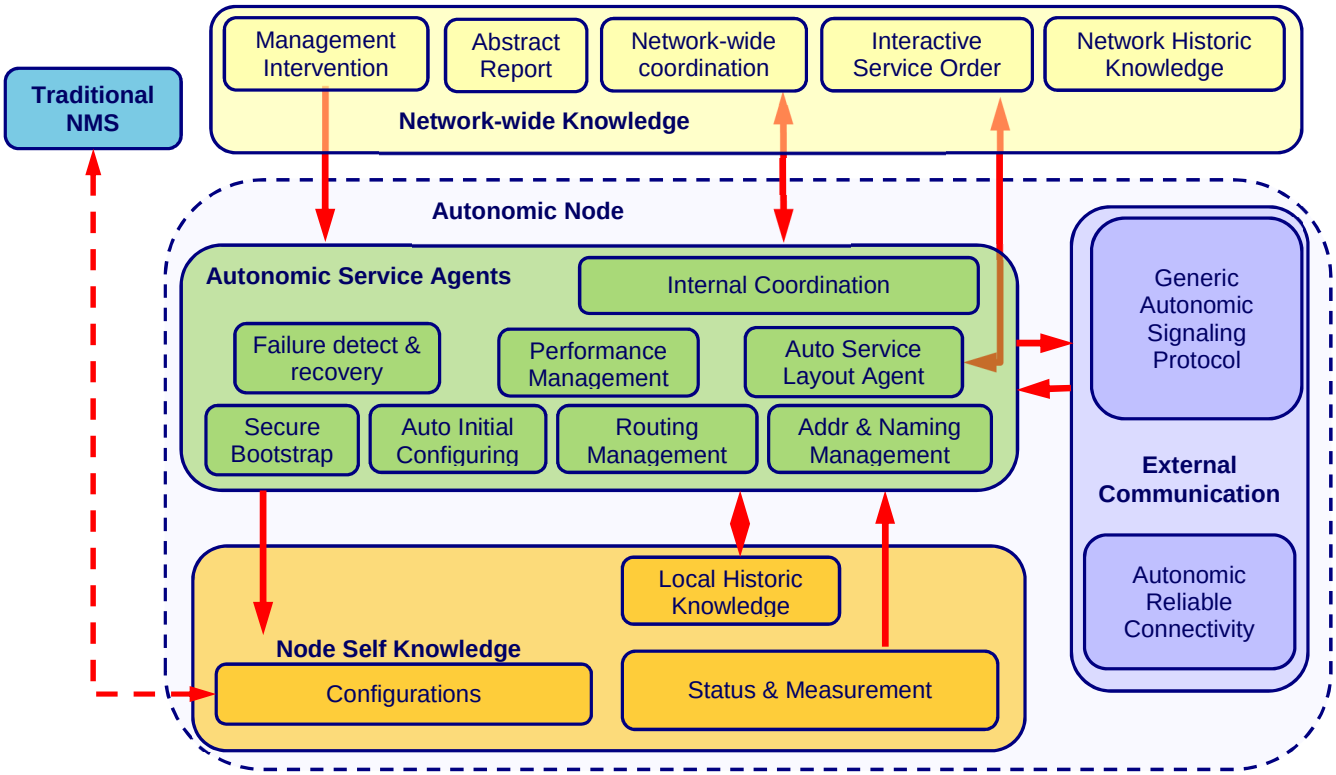

Figure 2: SXN Architecture Overview

As figure 2 shows, the architecture is angled from a node perspective; and the node is called Autonomic Node (AN). The essential component in an AN is the Autonomic Service Agent (ASA), which could be considered as applications running in network devices to fulfil specific network management functions/tasks without human intervene. There could be various kinds of ASAs to fulfil different functions/tasks; some ASAs, which are considered as basic and common functions in a network, are introduced in clause 9. The ASAs locate in different ANs communicate with each other through the External Communication module.

The node is not a closed system; it can accept external management intervene, in a Network-wide Knowledge approach. The network-wide knowledge is directly delivered to the node; and the node makes behaviour accordingly. The node also reserves an interface to traditional NMS (Network Management System).

8.2 Self Knowledge on Autonomic Node

There is some information that one autonomic node can get by itself. For example, the node could easily get its current configurations, which is usually valuable knowledge; or, the node does some measurement to learn the current status such as bandwidth, delay, etc.

8.3 Interaction Functions on Autonomic Node

Autonomic nodes need to interact with each other. First of all, the nodes need to build up a stable and secure communication channel automatically. One example of this kind of communication channel is the ACP (Autonomic Control Plane), which under standardization in IETF. The ACP leverages the Ipv6 link-local addresses to build up hop-by-hop secure tunnels; and leverage the ULA (Unique Local Address, which does not need to be applied from registrar) addresses for domain routing. The whole process of ACP is without any human intervene. Besides, the ACP does not allow manual configuration to get stability.

The communication channel is only a basic communication utility. To fulfil autonomic control and management functions, there shall be a standard protocol for the autonomic nodes to interact with each other. There is also an example from the IETF, which is called GRASP (GeneRic Autonomic Signaling Protocol). GRASP has two basic abilities: one is a discovery mechanism to find nodes that support some specific functions; the other is a negotiation mechanism which allows multi-rounds interaction between two nodes to fulfil a control/management goal.

8.4 Autonomic Service Agents on Autonomic Node

An ASA is an application executed in an autonomic node, to do the control and management tasks. ASAs in different autonomic nodes can discover and coordinate with each other through autonomic signalling protocol.

There could be various kinds of ASAs to fulfil different network control and management tasks respectively; one autonomic node could host multiple ASAs which could be executed simultaneously.

An incomplete list of ASAs is described in clause 9.

8.5 Network-wide Knowledge

The Network-wide Knowledge is a concept that represents the valuable information extracted from the data in various nodes or some network-level policies/information input from the administrators. In principle, extracting the Network-wide Knowledge does not necessarily require a dedicated information exchange mechanism among nodes, although these mechanisms might be present in some designs.

One example of the Network-wide Knowledge is the network wide coordination which could find conflicts between different policies and make resolution. Such coordination should be done at the network-level.

Another example is to extract knowledge from historic data. The data could be nodes' configurations, logs, etc. Data mining and machine learning might be used.

8.6 Interaction with External Input/Intervention

As well as the network to be self-managed of its own, it also needs to reserve input/output interfaces to external systems or administrator intervention.

There are mainly two kinds of external input that needs to be processed:

- The network administrators' commands. When the administrators want to interfere the network's status, they could input some commands to the network. And the network would interpret the commands and react autonomically. To be noticed, the commands should not be as detailed as the traditional command lines; it should be abstracted goal that the network needs to achieve. For example, the administrators could command the network to run at power-saving mode.
- Service order. When there are new service orders requested from the customers, the service order system (e.g. BSS) could input the orders to the self-managed network, and the network would execute the service orders automatically.

Besides input, the self-managed network also needs to output the external systems. One typical output is to report the current status of the network, or the results of an operation request. The report should be abstract and easy to review. Visualization of complicated network status is essential.

8.7 Negotiation between Autonomic Nodes for Autonomic Decision

As briefly described in clause 7.2, the IP network was originally designed in a distributed approach. However, the distributed design is mostly reflected in the routing. It is plausible to extend the distributed approach to network management.

One obvious way is to enrich the interaction between the autonomic nodes so that they can make consensus on some management tasks. In a nutshell, the devices need an negotiation capability to cooperate with each other. Unified definition form of negotiation objects, which means the decision objects and corresponding decision logics/policies are independent to individual protocol and can be defined in a model/structure form.

8.8 AI Technologies for Autonomic Decision

AI (Artificial Intelligence) has gained a very rapid development and some breakthrough in recent year. Current AI technologies are mostly powered by Machine Learning techniques; so AI in the present document also mostly refers to Machine Learning, which is a mechanism for self-decision that extracts rules used in network management and classify the various statuses inside and outside the system (obtained from measuring and monitoring).

If the detected event or situation has been considered in designing time, the network is normally able to deal with it according to the solution defined by the designer. This case is regarded as traditional system design. If not, the case is classified as "uncertainty". AI is needed mostly for this kind of uncertain cases. AI can make decision, and make the system have the capability of solving problem all by itself. AI itself is also realized by algorithms, but these algorithms have a certain degree of generic characteristic, thus developing a specific logic for every new situation is no more needed.

Some technologies in AI can also be used in data analysis; on the other side, traditional data analysis technologies are also used in AI.

Note that, there is another Work Item in NGP (ETSI GS NGP 006 [i.7]) dedicated for discussing AI technologies applied into network for gaining full autonomicity.

9 Autonomic Service Agents (ASAs)

9.1 ASAs for Basic Connectivity

Discovery Agent

Discovery is a very basic and critical behaviour in networking. In autonomic networks, discover behaviour is also very basic and critical.

Specifically, in the proposed architecture in the present document, the discover behaviour of an Autonomic Node is initialled by one ASA in the node to discover a specific kind of ASAs in locate in other Autonomic Nodes. Thus, in the discovery messages, there needs to be information indicating the objective ASA that the initialling node is willing to find.

There are two basic approaches for discovery:

1) Flood approach

   The Discovery message is propagated in a flood approach. The initial node sends the message to all the neighbours on the same link. The node receives the Discovery message would relay it to all the nodes of its link. When the message reaches to the node that supports the target ASA, it would directly response to the initiator according to the locator information in the Discovery message.

   Note that, in order to prevent the Discovery message looping around the autonomic domain, there needs to be corresponding loop avoidance mechanism in the Discovery Agent. There could be a simple hop-count limitation to restrict the distance one Discovery message travels; or the agent could record the Discovery message ID and abandon the same message travel back to it.

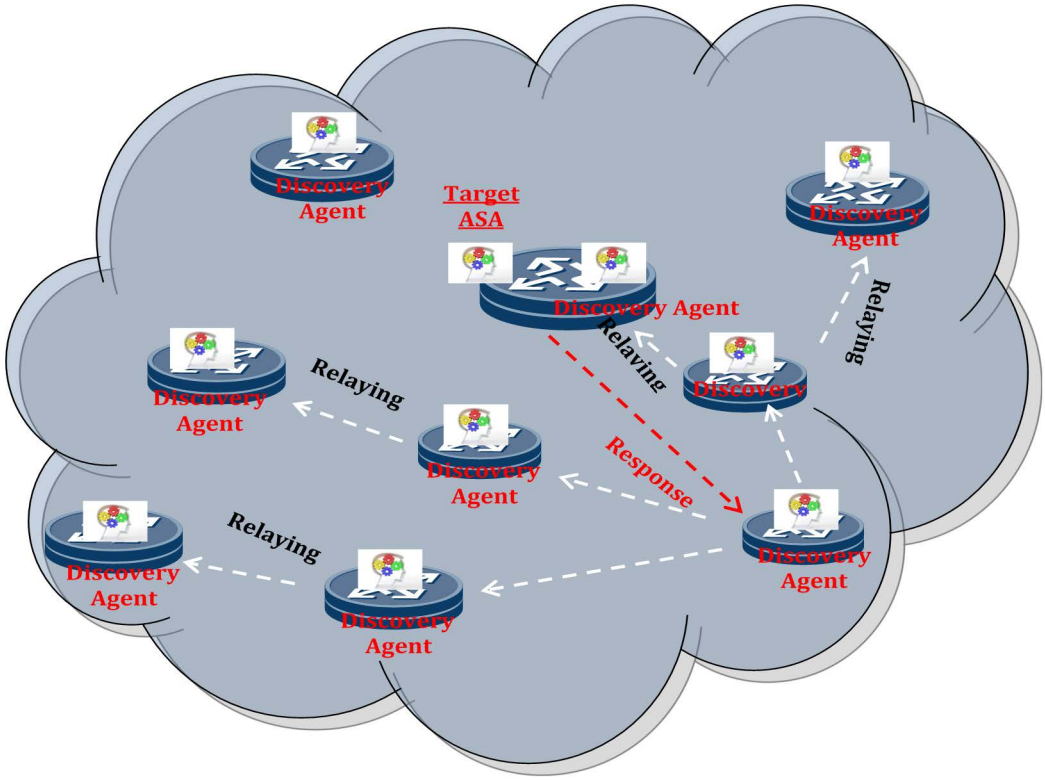

Figure 3-1: Discovery (flood approach)

As figure 3-1 shows, in an Autonomic Domain, every Autonomic Node should support Discovery ASA and relay discovery messages to its neighbours.

2) Directory approach

As the below figure shows, every Autonomic Node should support Discovery ASA and submit its supporting ASAs to the Discovery Server. When one node wants to find some an ASA, it only sends unicast message to the Discovery Server and get the result also through unicast.

When one ASA is added or deleted, or the ASA's locator is renumbered, the Discovery Agent needs to update the record in the Discovery Server as soon as possible.

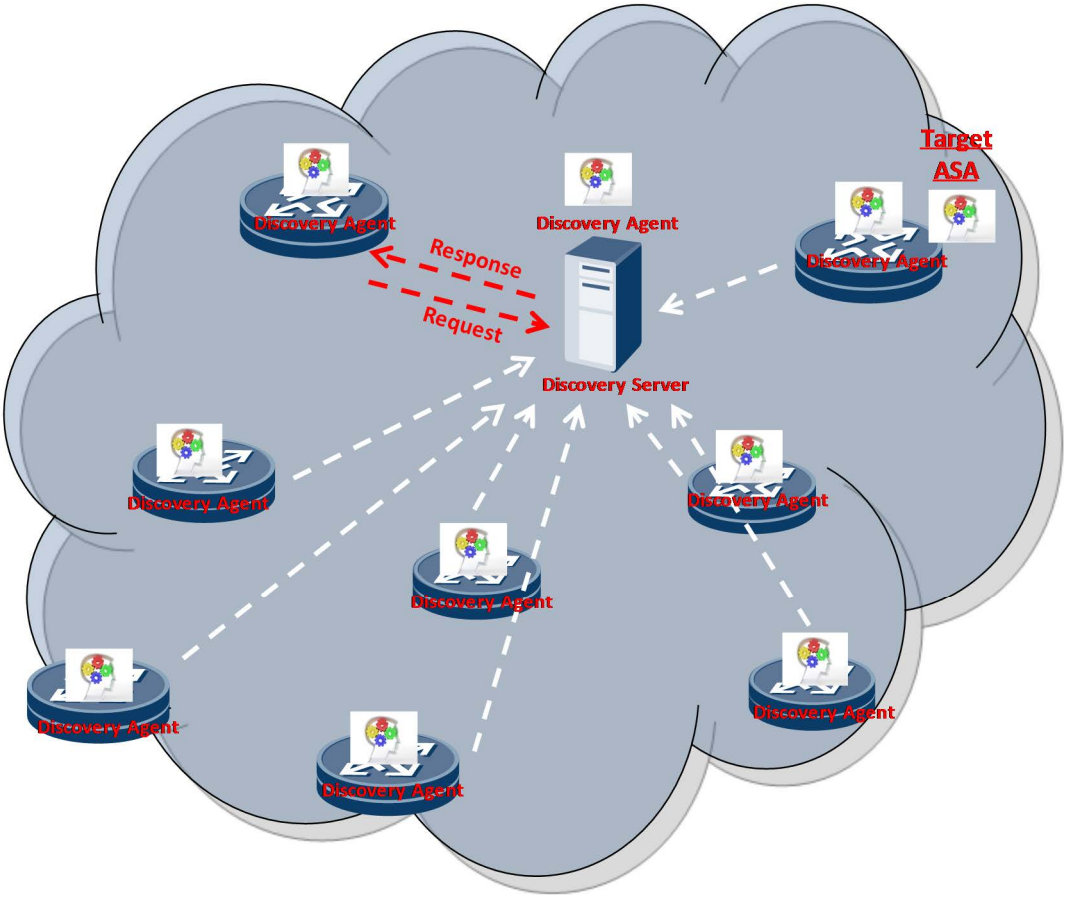

Figure 3-2: Discovery (flood approach)

The flood approach has the benefit that it does not assume any pre-configuration of the central Discovery Server while the directory approach has the benefit of high efficiency.

The idealist way is to enable both of the two approaches, but in different stage. At the very beginning of one node gets online, it utilize the flood approach to discover the Discovery Server first; after that, it could use the directory approach to submit its supporting ASAs or query some a ASAs located in any other nodes.

Secure Bootstrap Agent

This ASA is for a new device to find a local domain and join it.

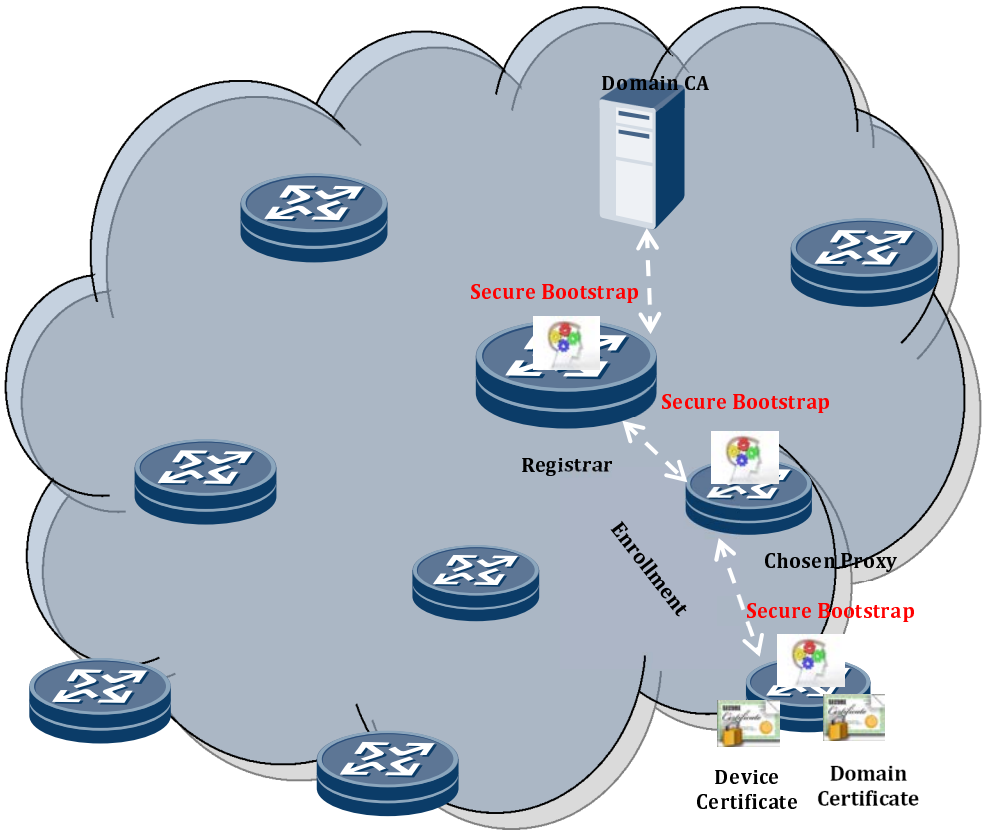

Figure 4: Secure Bootstrap

The new device needs to be authenticated to access to the network. A typical way for authentication is that the device is pre-installed a certificate. When getting online, the bootstrap ASA in the device would initiate the Discovery process (by Autonomic Signalling protocol) to find another entity, which could be named as Registrar and could do the authentication for the new device. However, at this point, since the new device hasn't got access, normally it could not get multi-hop communication with the Registrar. Thus, there might need a local Proxy to relay the communication for the new device. Every device that has got online could act as a Proxy by default so that no extra configuration would be needed to enable Proxy function.

When the device's neighbours receive the discovery messages, they would response to the new device, and the new device would choose one as its proxy. Then the new device enrols itself to the Registrar via the proxy by authentication of the Device Certificate. The enrolment interaction could be done by a dedicated protocol.

After authenticating the new device, the Registrar requests a Domain Certificate for the new device from the Domain CA. The keys in the Domain Certificate could be used by the new device for any further encrypt communication with other autonomic nodes in the same domain.

Autonomic Reliable Connectivity Agent

As introduced in clause 8.3, autonomic nodes need build up a stable and secure communication channel automatically to interact with each other. In each node, there needs to be an ASA to build up the communication channel. When nodes bootstrap, they can initiate the discovery process to find the neighbours that could build the reliable communication channel with it.

The reliable connectivity ASA should be initiated by default without any human intervene and should not be touched by human or mixed with the normal data plane, so that it mains a solid robust to survive through data plane or administration crush.

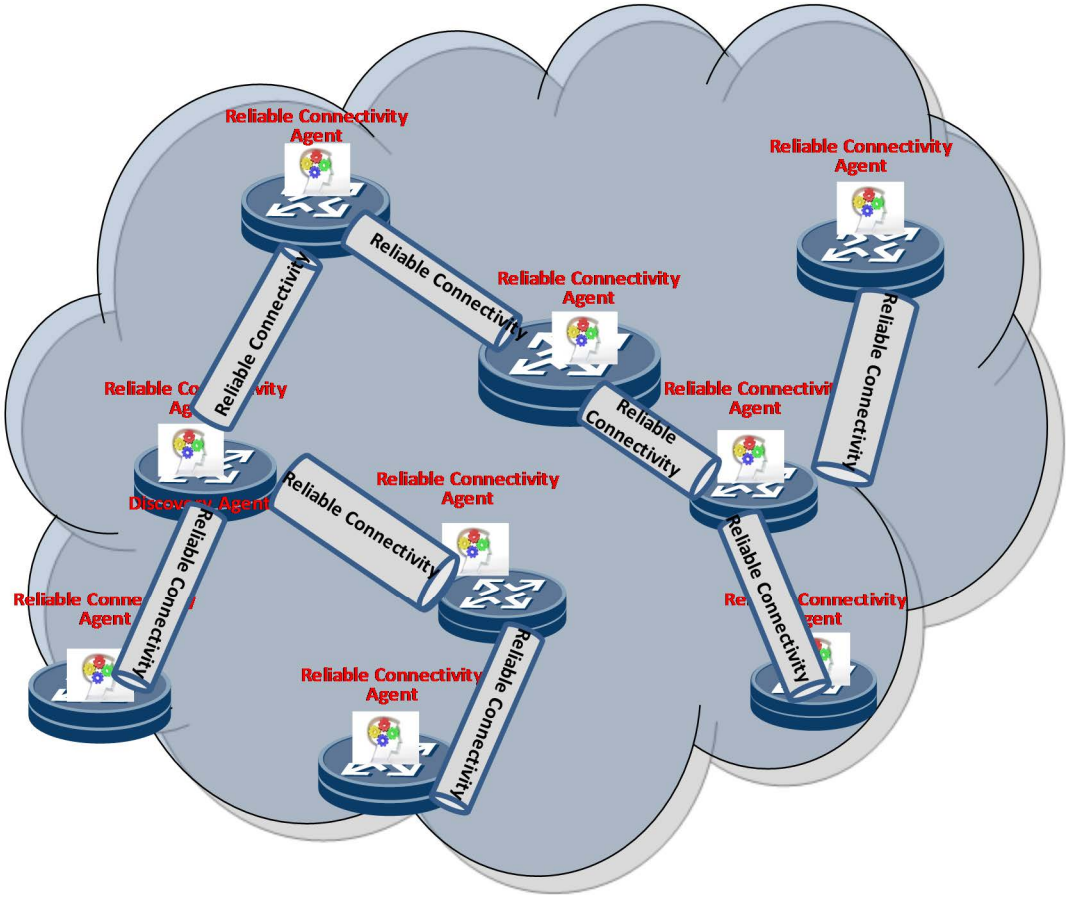

Figure 5: Reliable Connectivity between Autonomic Nodes

Addressing Configuration Agent

Addressing is very basic configuration in any networks. In an autonomic network, the addressing for sure should require minimal human intervene.

Addressing management ASAs should achieve maximum autonomous addressing. For interface address configuration, one basic principle for addressing is to assign each interface a unique address. There are basically two kinds of interface address configurations as the following:

1) Self-configuration

   A good example of address self-configuration is IPv6, which has built-in address self-configuration mechanisms for link-local addresses and Unique Local Addresses (ULAs). Once the devices boot up, they could self-configure themselves link-local addresses or ULAs on each interface. This could be a fully-autonomous process. However, this process has obvious limitations that it only can fit some scenarios: the link-local addresses could only be used on link; ULAs could be used off link, but they could not directly connect to the global Internet and the ULA prefixes would not be aggregated due to the random prefix generation defined by IETF RFC 4192 [i.9].

2) Auto-configuration based on network provision

   Also taking an example from IPv6, which provides two address auto-configuration mechanisms: SLAAC and DHCPv6. This process is autonomic at the host side; however, there are not a few configurations that are not easy to be autonomic on the network side.

For link prefix configuration, most of the scenarios might need global prefixes. Thus there should be global unicast prefixes provisioned from the network. Normally, each link should be assigned only one prefix so that the nodes in the link could be under one subnet to allow direct communication without host routing configurations. Link prefix could be easily advertised by protocol (e.g. ND) from the router to other nodes. However, how to allocate the prefixes to different links requires additional prefix management mechanisms might be needed.

For Prefix Management, the autonomic network is given an (or multiple) address block(s); then the ASAs on the devices communicate to slice the block(s) based on some algorithm(s) and to distribute the sliced blocks to different links.

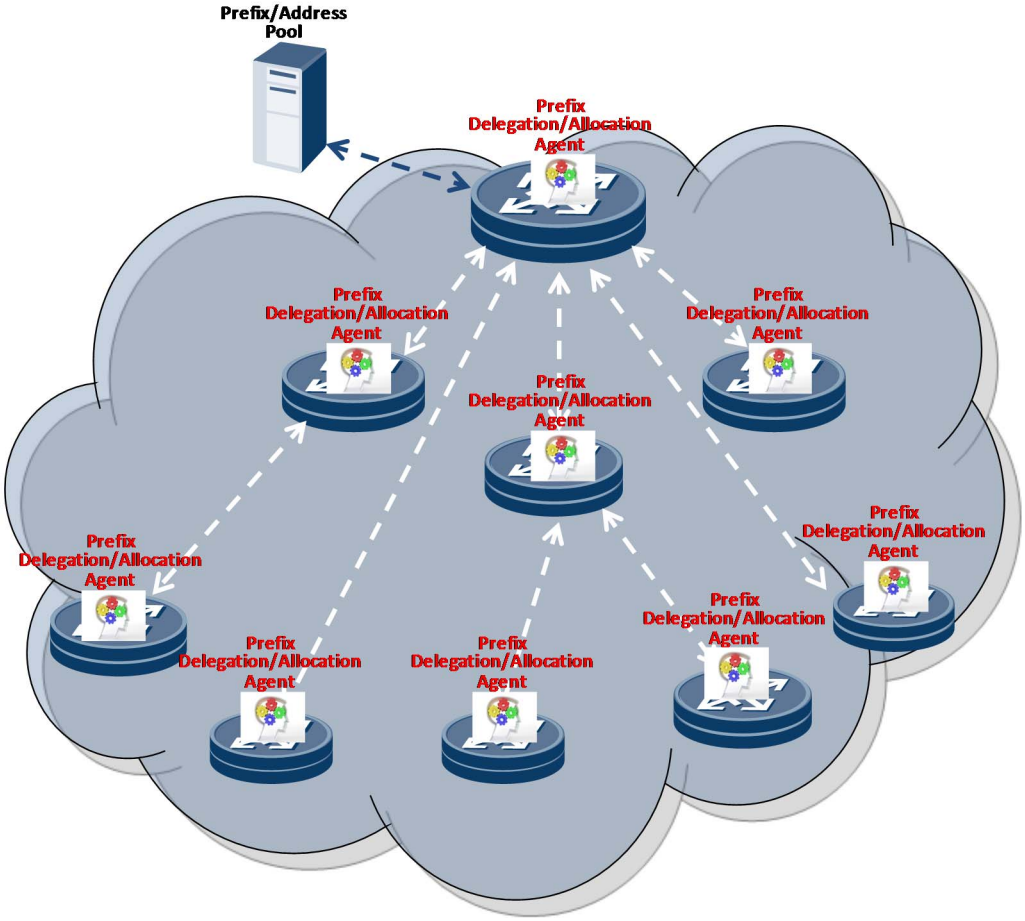

Figure 6: Prefix Delegation/Allocation

In some networks, especially large-scale ISP networks, a sophisticate but typical addressing schema is to separate the addressing space into several:

- Device interconnection addressing, which is for devices communicating on the same link.
- Device loopback addressing, each of which represents a single device. This is usually to be the router ID when doing routing calculation.
- Service interconnection addressing, which interconnects the device and the service system.

The address spaces might be separated visually numbers so that it could be easy for people to recognize or just separated in logic that could not be easily distinguished. The main purpose of separating the addressing spaces is for management considerations. When there is some fault happens, it would be easier to debug if the addressing spaces are separated.

Routing Configuration Agent

Every network node needs a routing table for forwarding. The simplest case of routing table is the default routing in a host, where there is only one route to the upstream gateway. Current protocols such as DHCP and ND have well-supported default route configuration for host automatically. However, if the autonomic node needs to join an IGP/BGP routing, the configurations for routing protocol are much more sophisticated than the default routing.

Basically, the routing protocol configuration could be divided into two parts as the following:

1) Routing domain/area partition: when the IGP nodes amount reaches to a certain number (normally, hundreds), the node would need to be partitioned into multiple domains for scalability consideration. The domain/area partitioning is difficult to be archived by distributed nodes interaction, thus, the ASA doing the domain partition needs to be located in a central node such as controller or NMS.

   However, once the initial domain/area partition is done, the new nodes could learn the domain/area information from its neighbours and there is no need to contact the controller/NMS for domain configuration. But there is one exception that when a new node learns different domains from its neighbours, it might still need to contact the controller/NMS for determining which domain/area to join.

2) Routing parameters: there are not a few parameters need to be configured for a routing protocol. However, many of them just don't impact the routing connectivity; thus, default value is always workable for them. For the parameters that need to be consistent among the whole domain/area (e.g. Hello time in OSPF), they also need the ASA in the controller/NMS to do it.

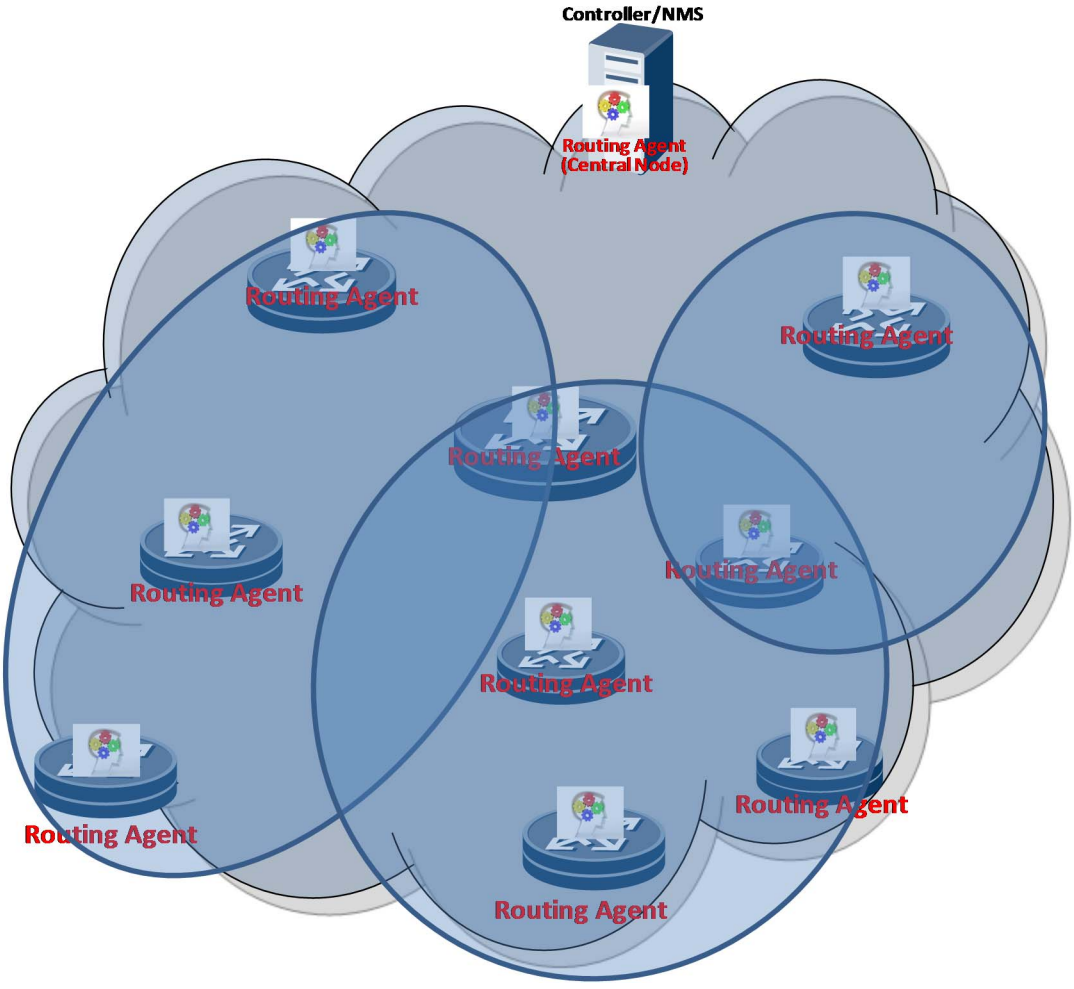

Figure 7: Routing Configuration Agent

9.2 ASAs for Management Infrastructure

Information Distribution Agent

In an autonomic network, some management and control information (e.g. network policies and intents) need to be distributed across the domain. There are two different scenarios:

1) Whole Domain Distribution

   Once the information is input to the autonomic network, the node that firstly handles it is be able to distribute it to all the other nodes in the autonomic domain. The distributed policy/intent might not relevant to every autonomic node, but it is flooded to all the devices.

2) Selective Distribution

   When one node receives the information, it only replicates it to the neighbours that fit for certain of conditions. This could reduce some unnecessary signalling amplification. However, this scenario implies there needs to be corresponding mechanisms to represent the conditions and to judge which neighbours fit for the conditions.

Figure 8 is two examples of selective distribution. The first one is distribution along with a path; the second one is distribution within certain hierarchical levels.

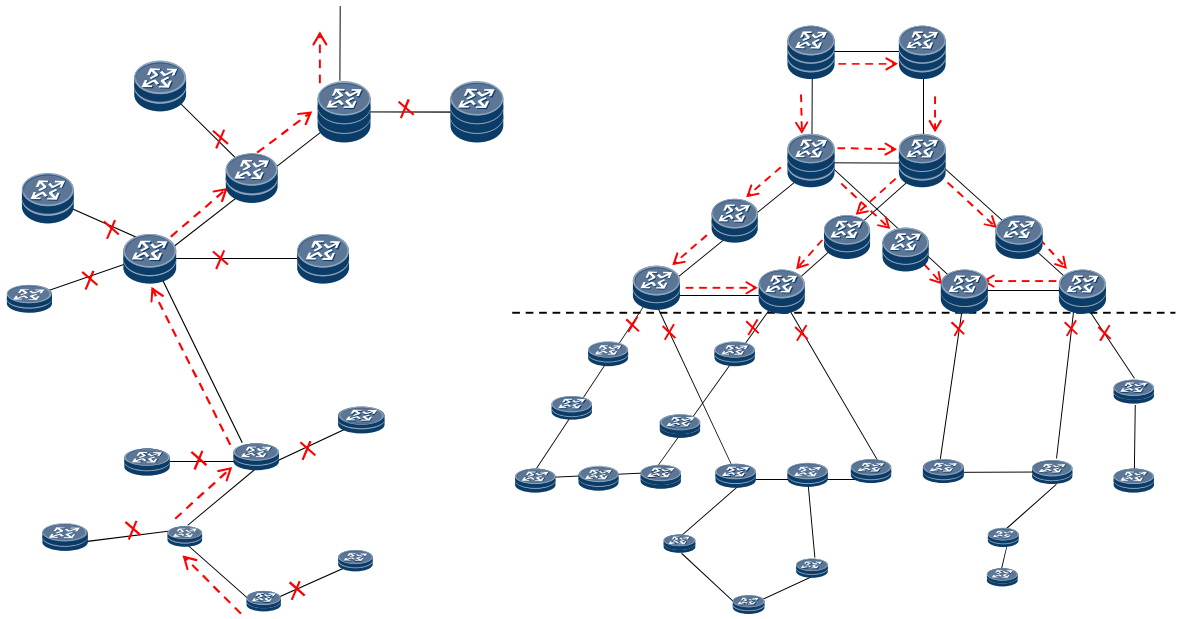

Figure 8: Selective Distribution of Network Policy/Intent

ASA Management Agent

An ASA is essentially an executable application hosted in the network device. In this sense, the ASA itself is also a managed object. Thus, there needs to be a kind of more fundamental ASA to manage all the other ASAs. This kind of ASA is called ASA Management Agent. For ASA management, it is mostly regarding to one ASA's status through its whole lifecycle:

- ASA installation

  Some ASAs are installed in the devices by default, while some might be dynamically loaded. Thus, the Management Agent needs to bootstrap the dynamic load, to make sure the ASA to be installed is from a reliable source, as well as properly installed and configured in the host device.

- ASA execution

  During the execution of ASAs, the Management Agent needs to monitor the status of various ASAs.

- ASA shutdown/uninstall

  The Management Agent should be able to shutdown the ASAs that behaviour badly. For some ephemeral ASAs, after their execution, they might need to be uninstalled for better efficiency or less security thread.

  There are two kinds of Management Agent. One kind is only responsible for the device where Management Agent locates in; the agent manages all the other ASAs' installation, execution monitoring, uninstall in the host device. The other kind Management Agent is a sort of orchestrator among a set of ASAs which are actually the same functional agents but distributed among different devices among the autonomic domain; in this case, the Management Agent might not have the decent capability to well handle the installation and shutdown/uninstall of the other ASAs, but it can monitor and control the others' behaviour remotely through signalling interaction.

Internal Coordination Agent

When multiple ASAs are operating the same resources, it is possible that the ASAs might conflict with each other. Thus, coordination between ASAs is needed. A central coordination ASA is needed to detect and handle the confliction.

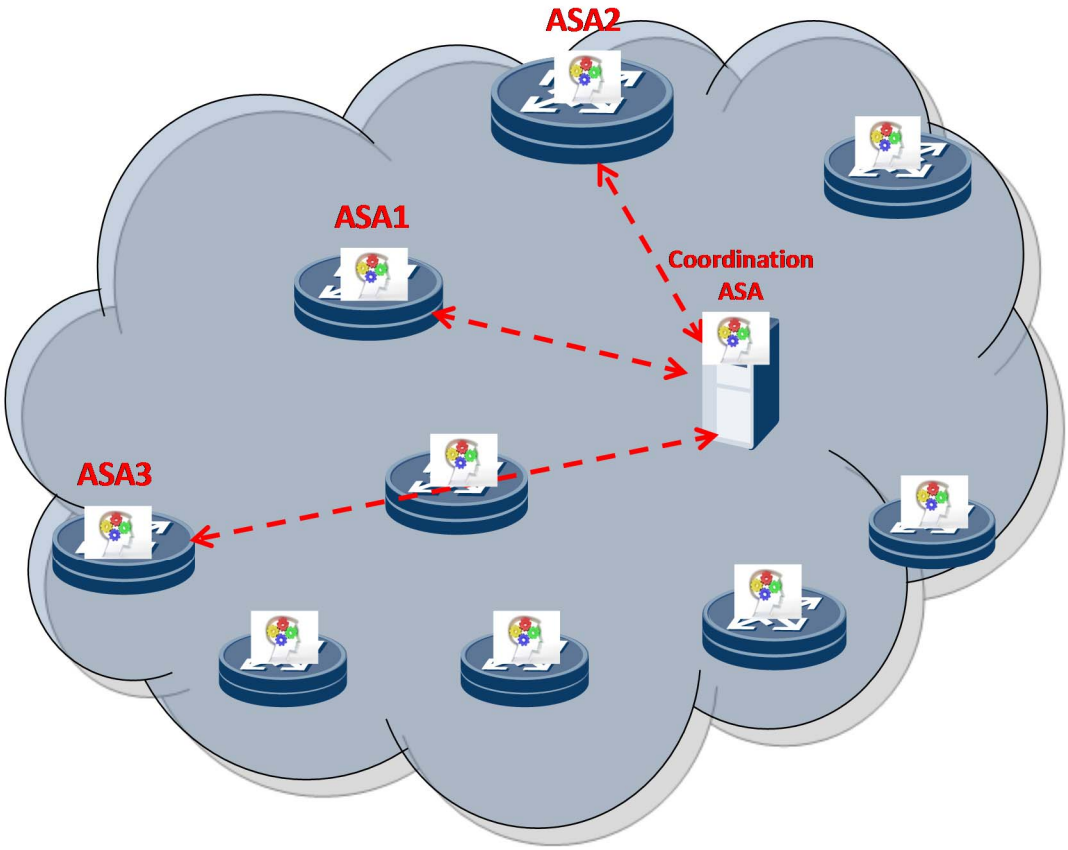

Figure 9: Coordination between ASAs

9.3 ASAs for Management Functions

Naming Management

Inside a domain, each autonomic device needs a domain specific identifier:

- Naming requirements:

  - Representing Devices: in an autonomic domain, each device is assigned (or self-generated) a name. The assigned/generated name is binding to the device.

  - Uniqueness: the names shall not collide within one autonomic domain. It is acceptable that the names in different domains collide, since they could be distinguished by domains.

- Semantic Encoding: it is recommended that the names encode some semantics rather than meaningless strings. The semantics might be:

  1) Device type;

  2) Functional role;

  3) Location;

  4) Ownership; etc.

  This is for ease of management consideration.

- Consistency: the devices' names should follow the same pattern within a domain.

- Proposed Mechanisms for ASA generating names:

  - Structured Naming Pattern: the whole name string could be divided into several fields, each of which representing a specific semantic as described above. For example: Location-DeviceType-FunctionalRole-DistinguisherNumber@Domain. The structure should be flexible that some fields are optional. When these optional fields are added, the name could still be recognized as the previous one. In above example, the "DistinguisherNumber" and "NameofDomain" are mandatory whereas others are optional. At initial stage, the devices might be only capable of self-generating the mandatory fields and the "DeviceType" because of the lack of knowledge. Later, they might have learned the "Location" and "FunctionalRole" and added the fields into current name. However, the other devices could still recognize it according to the same "DistinguisherNumber".

  - Advertised Public Fields: some fields in the structured name might be common among the domain (e.g. "Location" "NameofDomain"). Thus, these parts of the names could be propagated through Intent.

  - Self-generated Fields: The mandatory fields should be self-generated so that one device could name itself sufficiently without any advertised knowledge. There should various methods for a device to extract/generate a proper word for each mandatory semantic fields (e.g. "DeviceType", "DistinguisherNum") from its self-knowledge.

Network Sensing and Measurement Agent

The network sensing and measurement ASA is for sensing the network status/event and measuring the network quality of service such as network use ratio, packet drop ratio, and network delay, etc. There are two forms of sensing and measurement ASA:

1) Local sensing and measurement agent. The ASA is in each autonomic node; it only does the measurement behaviour with the node itself. It can directly measure the traffic or data passing through or located within itself; it can also does the measurement through some standard interaction with other nodes (e.g. ping the other nodes). In some cases, local ASAs can also coordinate to do one sensing or measurement task, as shown in figure 10.

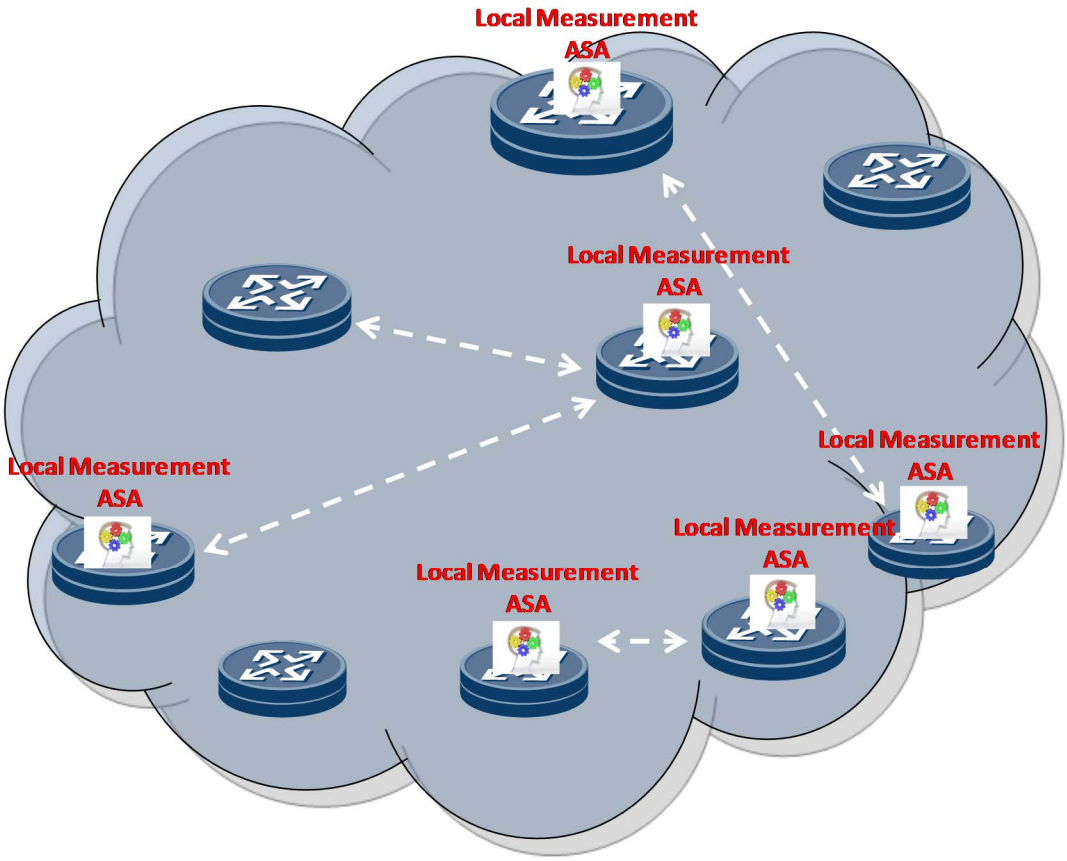

Figure 10: Measurement ASA (Coordinated Measurement)

2) Global sensing measurement agent. The ASA collects measurement from multiple local measurement agents, which act as measuring probe, and calculate some network-level results (e.g. end-to-end network delay).

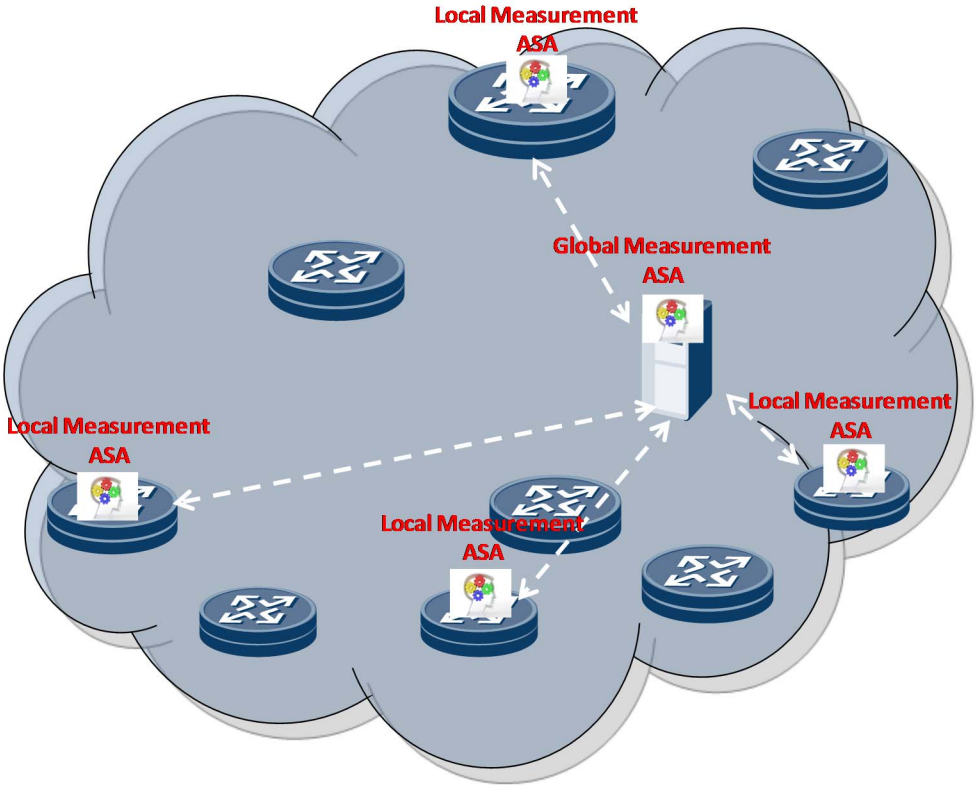

Figure 11: Measurement ASA (Global Measurement)

Traffic Control Agent

Based on the traffic measurement, the traffic control ASA could manipulate the traffic forwarding to improve the network performance. There are several possible ways to manipulate the forwarding:

1) ECMP. This is a classical approach to get network load balance for a given destination. It could in some degree avoid the congestion, especially when the traffic to the destination is very big.

2) Traffic engineering. This is to make specific paths for the given destinations. Combine with the measurement agent, it can even predict the traffic later, and then calculates the corresponding TE paths accordingly.

3) Multiple paths. This is to use parallel paths simultaneously to gain a better performance. It can divide the target traffic into N-tuple flows, and deliver them to the each path; or even deliver each single packet to the paths.

Failure detect and Recovery Agent

The failures that are very explicit and happened in a single node could be easily detected by a local failure detection and recovery ASA. For example, when a hardware failure happens in a node, normally there would be explicit abnormal behaviours or hardware signals for the ASA to judge. Thus, the recovery could be done very fast as fast control loop. However, some failures, especially hardware failures, are very difficult to be recovered.

There are also failures of which the root cause is very difficult to find. For such cases, there needs to be an ASA located in the controller/NMS to analyse the logs.

9.4 ASAs for Service Provisioning

Autonomic Service Layout Agent

A service layout ASA is responsible for interpreting the abstracted service (which could be defined as intent) come from the NMS or human intervene. Ideally, the administrators should be able to choose an arbitrary node to input the service.

The chosen node acts as a controller to translates the service into specific configurations; and delivers them to the devices respectively.

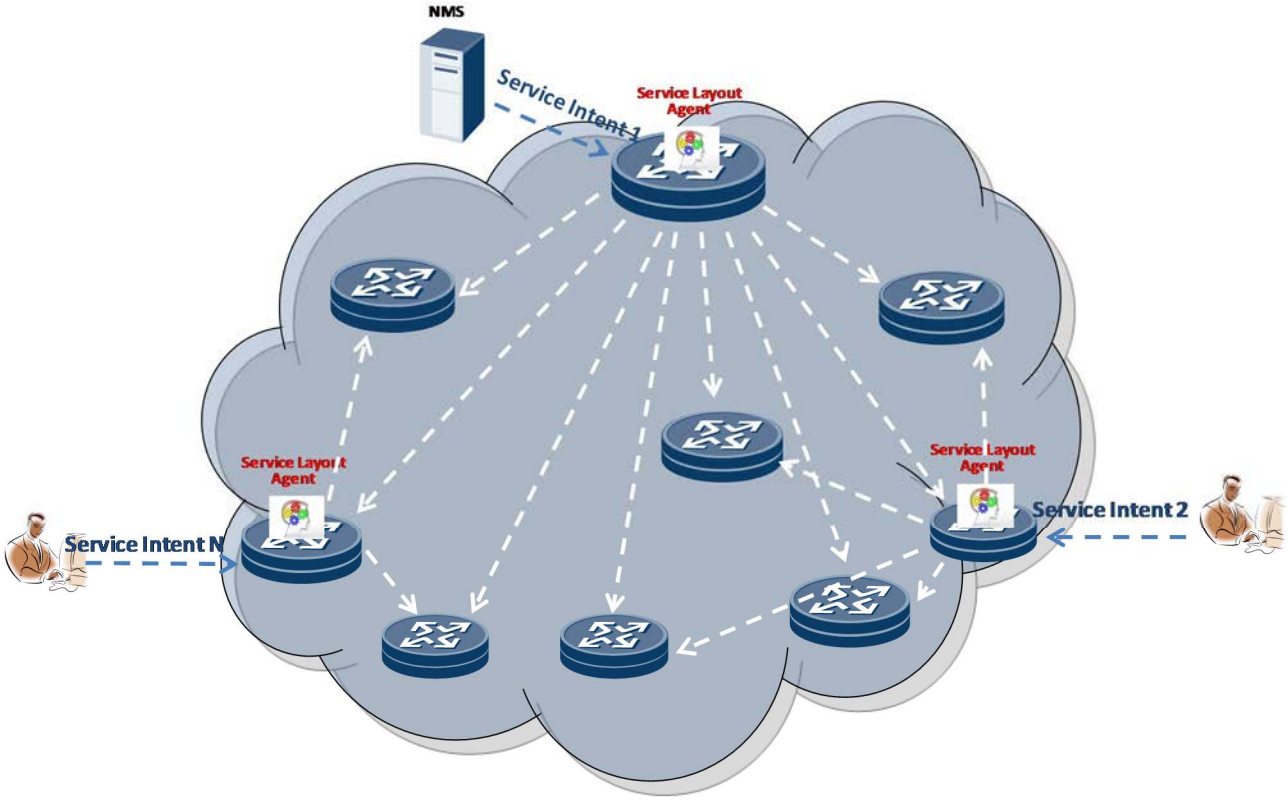

Figure 12: Service Layout ASA

10 Use Cases of Self-X Network

10.1 IP-based Radio Access Network Self-configuration (IPRANconf)

Use case description:

This use case is to enable autonomic configuration of the IPRAN network devices at the initial stage. It includes the basic connectivity configuration; wireless network services bearer configuration and the synergy with the base station on the devices locate on the cell site.

Use case context:

1) Actors and Roles: the NMS (Network Management System) is the user in this use case.

2) Telecom resources: the network including its OSS.

3) Assumptions:

   3-1) The base station devices need to cooper with the backhaul devices in order to fulfil the autonomic configuration.

   3-2) Backhaul devices can interact with each other autonomically to learn/determine the configuration parameters.

   3-3) The NMS can issue some network-level policies in the form of Intent to guide the configurations on the backhaul devices.

   3-4) The devices get online in an incremental manner.

4) Begins when: the devices first get online.

Technical Steps:

1) The higher level devices (RSG and ASG as showed in figure 5) already get online. (Traditional configuration involves human might be needed.)

2) The access devices (CSG as showed in figure 5) get online and discover the higher level devices through discovery agent.

3) The newly getting online CSG interacts with ASG/RSG to get the basic connectivity configured.

4) The newly getting online CSG discover the base station device.

5) The base station triggers the wireless network services (mostly in the form of MPLS VPN) configuration request.

6) The CSG makes further interactions with ASG/RSG to fulfil the MPLS VPN configuration.

7) A cascaded interaction between ASG and RSG might be needed.

Fits into the proposed architecture

The IPRANconf use case could fit into the proposed architecture as figure 13.

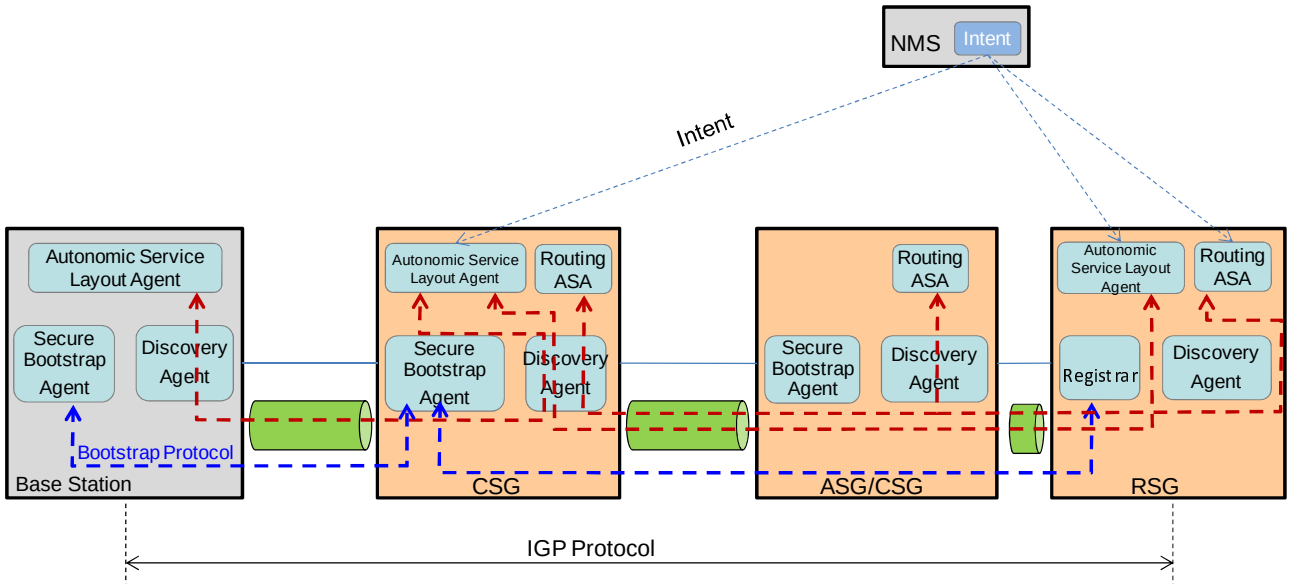

Figure 13: IPRAN Use Case

As figure 13 shows, the essential entities in the use case are the ASAs in the devices.

1) Autonomic Service Layout Agent

This ASA needs to be enabled both in the wireless network device (Base Station) and the backhaul network device (CSG, Cell Site Gateway). And they need to collaborate to trigger the backhaul configurations supporting wireless network services.

The wireless network services are mostly enabled by MPLS VPN technologies in the backhaul network. So, the Autonomic Service Layout Agent in this scenario needs to configure various kinds of VPN mechanism such as L3VPN, L2VPN, PW, etc.

2) Routing Configuration Agent

This ASA is to configure the protocols/functions regarding to basic connectivity, such as IGP/BGP configurations.

10.2 Automated Cluster Organization (ACOr)

Use case description:

ACOr enables automatic initial configuration of a cluster of network resources from a set of library templates and bounds. The use case may be employed for multiple clusters of multiple size and scope within a network.

Use case context:

1) Actors and Roles: the IRPManager is the user in this use case.

NOTE: The IRP is the Integration Reference Point defined in ETSI TS 132 501 [i.6].

2) Telecom resources: the network including its OSS.

3) Assumptions:

   3-1) OAM connection is working.

   3-2) IRPAgent can determine sufficient "RAN+CN+transport" configuration data on its own.

4) Begins when: network configuration data is to be made known to the IRPAgent.

Technical Steps:

1) IRPAgent indicates need for network configuration data to the IRPManager. (Optional.)

2) IRPManager transfers the network configuration data to IRPAgent or indicates to IRPAgent where the configuration data is available and IRPAgent retrieves the data from there. (Mandatory.)

3) IRPManager requests IRPAgent to validate the received configuration data. (Optional.)

4) IRPAgent validates the received configuration data. (Mandatory.)

10.3 Automated Cluster Optimization/re-organization (ACOp)

Use case description:

ACOp enables automatic re-configuration of the network as Cluster Optimization and Re-Organization as the network use and size grows and shrinks.

Use case context:

1) Actors and Roles:

   1-1) User: operator (sets bounds of operation).

   1-2) Autonomic Control Algorithm ACO.

   1-3) System: Next Generation Multi-Access Mobile Network.

2) Telecom resources: the network including all its management sub-systems.

3) Assumptions:

   3-1) The network is properly installed and running.

   3-2) The self-optimization objectives and targets have been set by operators.

4) Begins when: based on the monitored parameters (KPIs, Alarms, etc.), targets for the objectives defined for the self-optimization functions are not met.

Technical steps:

NOTE: The order of the bullet points in the list below does not imply any statements on the order of execution.

1) The input parameters (KPIs, Alarms, etc.) are monitored continuously.
2) When the monitored parameters do not meet the optimization targets, the optimization function is triggered.
3) Optimization function proposes corrective actions.
4) Selective option for operator to confirm the execution/activation of the proposed actions if needed or leave in automated mode.
5) Corrective actions are executed.
6) Optimization function monitors system status for a certain pre-defined monitoring time period.
7) The configuration prior to the corrective action is memorized if needed.
8) If the system status is satisfactory during the monitoring time period, then go to step 1).
9) Option for operator to confirm if fall back is needed, otherwise may be left in automatic mode.
10) Fall-back is executed.
11) The operator is informed about the progress and important events occurring during the self-optimization process.

11 Future Protocol and API Requirements

11.1 Protocol Requirements

This clause summarizes future protocol requirements as derived from clauses 7 to 10. Some of them already have candidates under standardization; while others need future work.

The requirements are mainly grouped according to clause 9's structure. Note that, each item listed below does not necessarily imply a corresponding protocol; multiple protocol requirements might be fulfilled by one specific protocol:

- Protocols for basic connectivity:
  1) A discovery mechanism:

     AN nodes shall be able to discover each other in the SXN. The discovery scope should be within the whole SXN. Furthermore, one AN node can do discovery in a ASA granularity rather than only discovering another AN node.

     The discovery mechanism in GRASP protocol [i.3] could be considered as a candidate.

  2) An autonomic reliable connectivity channel:

     As described in clause 9.1, the ASAs can utilize such protocol to build a stable communication channel which should be separated from normal data plane.

     The Autonomic Control Plane [i.4] can provide full functionality.

- Protocols for management infrastructure:
  1) Generic Control Protocol for ASA-to-ASA Horizontal Control Information Exchange and Synchronization:

     This requirement is inherited from the Protocol Requirement RQ_1 ("Generic Control Protocol for DE-to-DE Horizontal Control Information Exchange and Synchronization") of the AFI liaison input document (NTECH(17)000013 [i.8]). As explained in clause 5.3, the ASA concept in present document document is consistent with the "Decision Element" defined in the AFI GANA model.

The control protocol should define a set of a variety of selectable control semantics that may be employed by ASAs in exchanging control messages and information (e.g. simple one-way or two-way or multi-party control information flow, indications of whether an acknowledgement of information reception is needed or not, solicitations for information or push/pull behaviours (which may be employed by ASAs), negotiations for parameter value settings, negotiations for orchestrations or (re)-configurations, and other control information exchange that may be useful for ASA-to-ASA collaboration).

GRASP protocol [i.3] was mainly designed for this purpose.

2) Distribution of operational intervention:

The Intent needs to be distributed to the whole SXN. Flooding is a natural way for this, however, in some cases selective flooding might be needed to reduce the signalling storm.

3) A common way to identify nodes:

Autonomic nodes need to know each other. There needs to be a common naming scheme to represent the AN nodes.

- Protocols for service provision:

1) Coordination mechanism between AN nodes:

When AN nodes are doing service layout, they need to coordinate with each other for eliminating conflicts. Coordination between distributed nodes is not easy, more effort needs to be paid in the future.

- Protocols for security:

1) Access authentication:

As discussed in clause 7.4, new devices need to be authenticated to access into the SXN; and the access authentication procedure should be automatic.

BRSKI [i.2] is a good example for access authentication.

2) Authorization:

Also as discussed in clause 7.4, some prescriptive behaviour needs authorization.

3) Encrypt Communication:

The Autonomic Control Plane [i.4] also provides encrypt communications between AN nodes.

11.2 API Requirements

1) Operational intervention interface:

SXN should reserve the human intervene capability. However, the human intervene should be as abstract as possible.

The concept Intent mentioned in the present document is for this. Further work is needed to explore how to define and standardize intent.

2) AN internal API for ASAs:

The required API enables innovation in autonomics by enabling ASA developers to load and replace ASAs of varying decision-making capabilities into AN.

This aligns with the API RQ_8 (GANA node internal API for enabling GANA Level-3 and Level-2 DE to access and perform management & control of protocol stacks and other resources as Managed Entities (Mes) at the resource layer) of the AFI input document.

Annex A (informative): Authors & contributors

The following people have contributed to the present document:

Rapporteur:
Dr. Sheng Jiang, Huawei Technologies Co. Ltd.

Other key contributors:
Richard Li, Huawei Technologies Co. Ltd.

Kiran Makhijani, Huawei Technologies Co. Ltd.

Bing Liu, Huawei Technologies Co. Ltd.

Gerry Foster, University of Surrey

Seiamak Vahid, University of Surrey

Annex B (informative): Bibliography

- IETF RFC 7576: "General Gap Analysis for Autonomic Networking".
- Contribution number NGP(17)000034: "Requirements for Protocols and APIs for Enabling GANA based Autonomics, Cognitive Networking and Self-Management of Networks and Services in Evolving and Future Networks, LS from TC NTECH WG AFI to ISG NGP, February 2017.
- IETF draft-ietf-anima-autonomic-control-plane: "An Autonomic Control Plane", October 2016.
- Network Machine Learning Research Group, Internet Research Task Force.
- Autonomic Networking Integrated Model and Approach Working Group, Internet Engineering Task Force.

NOTE: Available at https://datatracker.ietf.org/wg/anima.

- ETSI AFI ISG Autonomic network engineering for the self-managing Future Internet (AFI).
- ETSI NGP ISG Intelligence Defined Network.

Annex C (informative): Change history

Date	Version	Information about changes
May 2016	0.0.1	First Draft, structure and scope.
July 2016	0.0.2	Added some content.
September 2016	0.0.3	Adjusted the structure with adding a couple of new topics.
November 2016	0.0.4	Added not a few content as the new clause 9, mostly regarding to ASA description. Added several use cases in clause 10.
December 2016	0.0.5	Completed the content in clause 9, as well as other blanket content.
February 2017	0.0.6	Addressed the comments in NGP#5, and made scattered small technical and editorial revision among the whole document.
March 2017	0.0.7	Changed the SMN terminology to SXN; respond the NTECH(17)000013 liaison to integrate some content from GANA model relevant document.

History

Document history		
V1.1.1	May 2017	Publication