Final draft ETSI EG 203 341 V1.1.1 (2016-08)

ETSI GUIDE

Core Network and Interoperability Testing (INT); Approaches for Testing Adaptive Networks

Reference

DEG/INT-00127

Keywords

conformance, interoperability, methodology

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

Important notice

Copyright Notification

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 final draft ETSI Guide (EG) has been produced by ETSI Technical Committee Core Network and Interoperability Testing (INT), and is now submitted for the ETSI standards Membership Approval Procedure.

Modal verbs terminology

In the present document "**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.

Introduction

The characteristics of "adaptive networks" such as virtualization, self-organization, self-configuration, self-optimization, self-healing and self-learning, dynamic network slicing promise to offer huge advantages in future networks. While technologies such as Network Functions Virtualisation (NFV), Self-Organizing Networks (SON), Mobile Edge Computing (MEC) and Autonomic Management and Control (AMC) of Networks and Services may not each exhibit all the characteristics they do have one thing in common: they are all dynamic rather than static, reacting to dynamic traffic conditions, applications, service demands as well as to changes in the eco-system environment.

By incorporating one or several of the technologies mentioned above, Adaptive Networks (AN) have the ability to automatically and dynamically manage and control network resources, configuration parameters or the network structure, with limited human intervention, in order to meet functional targets or operational policies. However, to achieve this type of autonomic behaviour, it has to be ensured that any modification that is performed automatically in the network does not produce undesired effects, e.g. instability or lower performance with respect to the end-user perspective.

Comprehensive testing, both on a general level as in type approvals and related to acceptance testing of a particular deployment, is therefore even more important than it is for conventional networks. Due to the fact that the components of an AN may interact in a more complex and interdependent way than in a conventional network, appropriate testing methodologies are required in all phases of operation. For instance, the effect of software updates in network components can be amplified by the more connected nature of these components in an AN.

The rest of the present document is organized as follows:

- Clause 4 gives the definition of an adaptive network, as used in the context of the present document.
- Clause 5 defines the entities and interactions that may be encountered in an adaptive network.
- Clause 6 defines the general functional targets that should be met by adaptive networks.
- Clause 7 defines the methods that may be used to test adaptive networks.

- Annex A gives an overview of the relation of the present document to other work performed in this area, e.g. NFV TST, NTECH-AFI.

1 Scope

The present document, "Approaches for Testing Adaptive Networks" defines a framework of testing principles and guidelines that may be used to test networks that exhibit some form of autonomic adaptive behaviour, which allows them to dynamically change their configuration, structure or operational parameters. The (re)-configuration is performed in response to stimuli such as changes in workload, operator policies that govern their operation, context (the network is context-aware and may have a degree of self-awareness); and challenges in the environment (i.e. conditions under which the network is operating, e.g. manifestations of faults, errors, failures in various parts of the network and its hardware and software components).

The functionality of individual components and basic interoperability can be ensured at design time. However, the complex interactions between various components or functions deployed in a live Adaptive Network (AN) may not be fully assessed or foreseen. Consequently, the document addresses methodologies to test ANs towards meeting their functional targets or policies, and ensuring a minimum trust level for autonomic operation of such networks.

NOTE: In the literature, both the terms "autonomous" and "autonomic" are being used in this context, whereas "autonomous" appears to indicate a higher level of automation. As adaptive networks are, at the time of writing, surely a technology still at its beginnings, "autonomic" may be a less ambitious and therefore more appropriate term for the time being. On the other hand, the NGMN 5G White Paper (V1.0) uses the term combination "autonomic/self-management functions" which points, clearly towards a level beyond "autonomic". As mobile networks are complex systems, it is most likely that the degree of automation will increase in the course of technical evolution, but not in an isotropic way; there will be areas with higher and others with lower levels of automation, and sophistication of respective functions. For these reasons, the present document will use the term "autonomic".

2 References

2.1 Normative references

Normative references are not applicable in the present document.

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: "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] ETSI TS 102 250-4: "Speech and multimedia Transmission Quality (STQ); QoS aspects for popular services in mobile networks; Part 4: Requirements for Quality of Service measurement equipment".

[i.3] Recommendation ITU-T P.10/G.100 Amendment 2 (07/2008): "Vocabulary for performance and quality of service Amendment 2: New definitions for inclusion in Recommendation ITU-T P.10/G.100".

[i.4] Recommendation ITU-T E.800 (09/2008): "Definitions of terms related to quality of service".

[i.5] ISO/IEC 9646: "Information technology - Open Systems Interconnection - Conformance testing methodology and framework".

[i.6] ETSI GS NFV-TST 001 (V1.1.1): "Network Functions Virtualisation (NFV); Pre-deployment Testing; Report on Validation of NFV Environments and Services".

[i.7] ETSI GS NFV-TST 002: "Network Functions Virtualisation (NFV); Testing Methodology; Report on Interoperability Testing Methodology".

[i.8] Dar, K.: "Autonomic Computing: An introduction to MAPE-K reference model".

NOTE: Available at http://www.uio.no/studier/emner/matnat/ifi/INF5360/v13/undervisningsmateriale/mape-k.pdf.

[i.9] IBM (2005):"An architectural blueprint for autonomic computing".

NOTE: Available at http://www-03.ibm.com/autonomic/pdfs/AC%20Blueprint%20White%20Paper%20V7.pdf.

[i.10] Hayan, Z.: "A novel autonomic architecture for QoS management in wired network".

NOTE: Available at http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=5700376&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxpls%2Fabs_all.jsp%3Farnumber%3D5700376).

[i.11] Strassner, J., Agoulmine, N., & Lethihet, E. (2006): "FOCALE - A Novel Autonomic Networking Architecture".

NOTE: Available at http://repository.wit.ie/189/1/2006_LAACS_Strassner_et_al_final.pdf.

[i.12] Clark, D. C., Partridge, C., Ramming, J. C., Wroclawski, J. T.: "A knowledge plane for the internet".

3 Definitions and abbreviations

3.1 Definitions

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

aggregation hierarchy: description of how detailed (granular) performance data will be aggregated into summary data, and vice versa, how to break down the summary data into details

attractor: state or behaviour toward which a dynamic system tends to evolve, represented as a point or orbit in the system's phase space

control loop: mechanism which uses observations of a system to make modifications to the observed system to meet a given target

3.2 Abbreviations

For the purposes of the present document, the following abbreviations apply:

AF	Adaptive Function
AFI	Autonomic Future Internet
AMC	Autonomic Management and Control
AN	Adaptive Network
CCO	Coverage and Capacity Optimization
DE	Decision Element
eNB	evolved Node B
FUT	Function Under Test
GANA	Generic Autonomic Network Architecture
IBM	International Business Machines
ISG	Industry Specification Group
ITU-T	International Telecommunication Union - Telecommunication standardization sector
KPI	Key Performance Indicator

LTE	Long-Term Evolution
MEC	Mobile Edge Computing
MRO	Mobility Robustness Optimization
NE	Network Element
NFV	Network Functions Virtualisation
NGMN	Next Generation Mobile Networks
NTECH	Network Technologies
NUT	Network Under Test
OCS	Overall Configuration State (of a network)
ONP	Overall Network Properties
QoE	Quality of Experience
QoS	Quality of Service
RAN	Radio Access Network
SDN	Software Defined Networking
SLA	Service Level Agreement
SON	Self Organizing Networks
UE	User Equipment
VoLTE	Voice over LTE

4 Definition of Adaptive Networks

4.1 Basic Concept

The term "adaptive network" (AN) refers to any network that has the ability to automatically modify its configuration, operational parameters or structure, in order to comply with pre-defined functional targets or operational policies, and with the ability to handle situations that were unknown at its design time (e.g. with predictions and forecasting capabilities as well), thus producing a dynamic environment with multiple potential network states. An adaptive network may include technologies such as Self Organizing Networks (SON), Network Functions Virtualisation (NFV), Software Defined Networking (SDN), Autonomic Management and Control (AMC) or any other technology which enables a network to exhibit the characteristics mentioned above.

Adaptive networks are comprised of one or more Adaptive Functions (AF) that dynamically and adaptively manage and control certain network attributes. These functions are fundamentally characterized by exhibiting control-loops which can be embedded at different layers e.g. protocol level, node level, network level, and exert different degrees of influence over the network. Similarly, the management and control of the AFs can be aggregated at different levels depending on the information required for their operation. Furthermore, ANs may function on different time scales and with different levels of complexity and views on which they operate on, depending on the type of AFs that are deployed. However, from an end user perspective, the presence or absence of AFs in a network is transparent, meaning that end users can only observe the functionality of the network service. Similarly to conventional networks, the internal structure and operation of the network is not visible from this perspective.

Depending on the type of AFs and the level where they are deployed, the frequency of changes performed throughout an AN can differ. In general, low level AFs can operate at faster time scales, i.e. fast control loops as they utilize information collected locally. On the other hand, high level AFs require information about the overall state of the network and thus typically operate in slow control loops. The architecture of an AN, in terms of the hierarchical placement of AFs and aggregation levels is important from a testing perspective and determines if and how the particular network can be tested. Figure 1 illustrates the different architectures of ANs and the associated control loops.

Two extreme cases can be distinguished:

- Fully distributed adaptive network, where all AFs operate at lower levels, e.g. at the protocol or node level, with no management and control aggregation at higher levels.
- Fully centralized adaptive network, where AFs operate at higher levels, e.g. network level and aggregate network wide information.

The fully distributed architecture poses higher challenges from a testing perspective, since the effect of AFs that operate in fast control loops may not be easily translated into functional KPIs that can be observed by a test system. Furthermore, their policies and functional targets are managed and executed locally, at an aggregation level where information may not be available for a test system. On the other hand, the fully centralized architecture is the most attractive from a testing perspective, since it operates using slow control loops and uses information that is aggregated at network level.

A typical AN will incorporate several types of AFs, that operate and aggregate information at different levels. Hence, from an architectural perspective it may use a hybrid model, which includes distributed, and centralized AFs or AFs that are aggregated at an intermediated level. Additionally, a peer-to-peer relationship may be formed between AFs operating at the same hierarchical level.

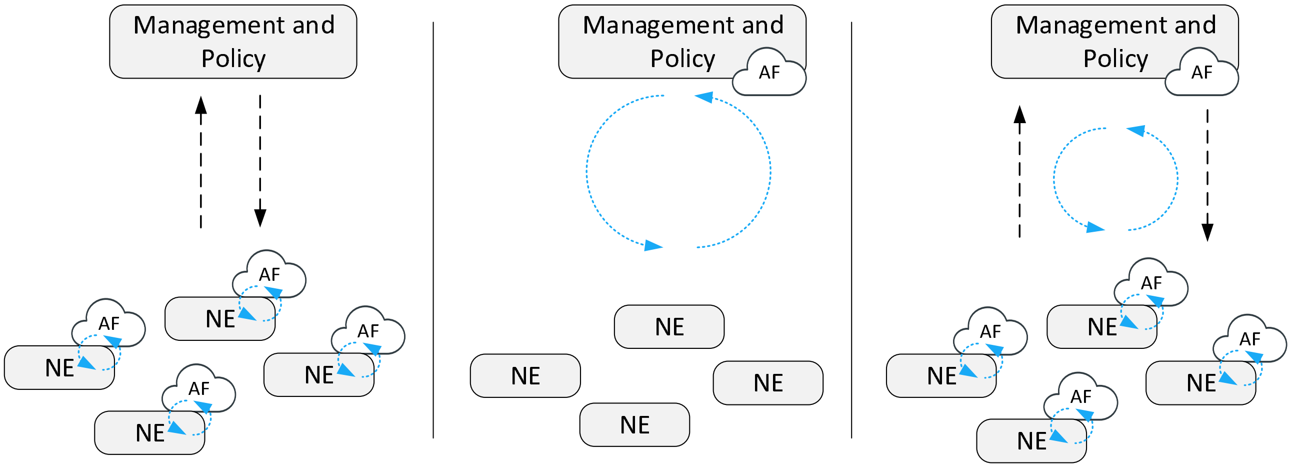

Figure 1: Adaptive Network Architectures: distributed, centralized and hybrid

The detailed internal structure and algorithms of the AN may not be known to an external test environment. However, a minimum set of information regarding the operation and structure of the AN may be required in order to interpret results generated from end-to-end functionality testing. This information can include details about the functional targets of the AN, the capabilities of AFs that are deployed, their operational status, e.g. active, idle, disabled, the network attributes that they control and their influence on the functional target being measured. Part of the information may be obtained out of band, i.e. be provided as external input to the test system, while part of the information may be obtained from the Network Under Test (NUT).

An adaptive network typically functions in a closed loop manner, with minimum human intervention using sensor information to make decisions and perform actions, according to policies set by the network operator. These actions can be categorized in:

- Actions that are performed on network configuration parameters or network resources, e.g. Transmission Power, antenna tilt, routing policies, bandwidth allocation.

- Actions that are performed on the network structure, e.g. adding/removing network elements (either physical or virtualized instances). These actions imply configuration changes in order to accommodate the structural change.

The events that can trigger an adaptive network to dynamically change its properties vary also depending on the specific AFs deployed in the network and the level at which they operate. They can be split in two categories:

- Externally generated events - when the adaptive behaviour is triggered by an external factor, e.g. increase in user traffic that creates unbalanced load in the network, detecting service-level performance degradation, failure of network elements.

- Internally generated events - when the adaptive behaviour is triggered as a result of an internal policy, independent of external activity, e.g. power savings mode, configuration of network properties to provide QoS for certain traffic types, e.g. low latency traffic, delay-tolerant traffic, low-bandwidth traffic.

NOTE: These events can occur in a chain like fashion, e.g. policy change can trigger several secondary events in lower level functional units.

4.2 General Terminology

4.2.1 Introduction

A fundamental characteristic of ANs is the ability to dynamically change their configuration and properties. In order to describe the testing methodology some basic concepts (configuration states, state transitions and attractors) have to be introduced, as their meaning is new or goes beyond well-known definitions for conventional networks.

4.2.2 Network States

A network is characterized by its hardware and software components, together with the configuration of these components. This configuration is given by control elements, which can be on hardware level (e.g. elements determining physical orientation of antennas) or on software level (parameters determining the functional behaviour of a component). A component can have multiple control elements which define its overall state. Similarly, the overall network state is defined by the overall states of each component. The total number of these controls - counting each degree of freedom separately - is typically large, but finite and a fixed property of a given network.

Each degree of freedom can be:

- a discrete value, out of a given set of choices or a range of integer values; or
- a continuous (analogue) value.

The totality of all degrees of freedom represents the settings space. Each combination of settings can be described as an N-dimensional vector, where N is the number of degrees of freedom, also called the dimension of the settings space. An individual control setting is then the i-th element of this settings vector.

Each possible combination of settings is represented by the corresponding vector. For the purpose of the present document, such a vector is termed Overall Configuration State (OCS).

A change of settings - regardless if done by human operators as in conventional networks or by automatic processes in AN - means a transition between an initial OCS S_1 to a new OCS S_2.

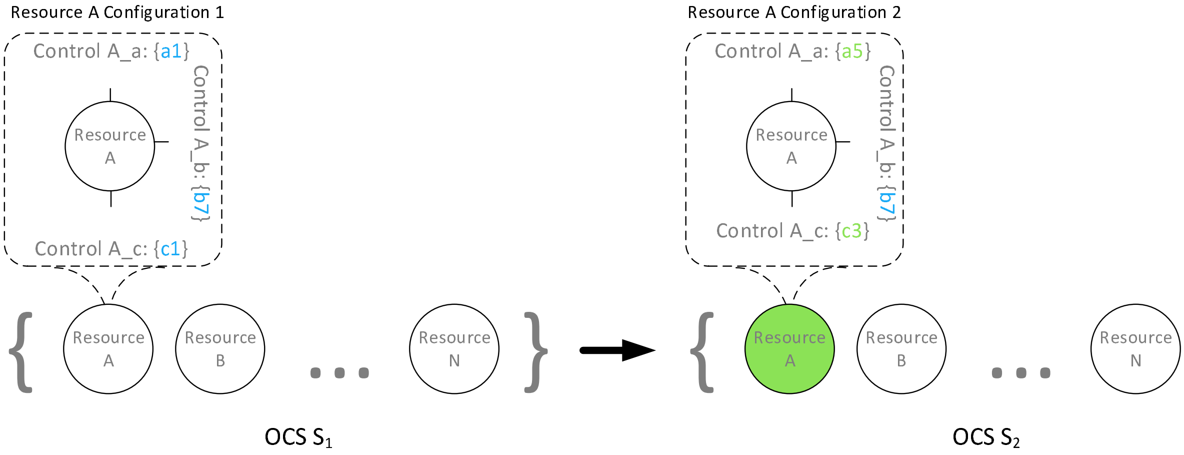

Figure 2: Concept of controls and Overall Configuration State (OCS) transitions

Also for this purpose and later usage, the term overall network properties (ONP) is defined which describes the appearance of the network as perceivable from the end user point of view or through other interfaces to the network operator (see also clause 5.1). Each OCS leads to a specific ONP.

NOTE: This relation is not symmetric; several OCS can lead to the same ONP, but the assumption is that the same OCS cannot lead to different ONP. If this was the case it would mean that some aspect of the network shows random behaviour which is a primarily unwanted condition.

4.2.3 Static and stationary states

In a conventional network, where controls are operated by humans, it is likely that settings, once made, do not frequently change after they have been made. In contrast, in ANs, settings and associated network properties constantly change as a result of various AF that operate in the network.

NOTE 1: In the context of the present clause, the term "state" represents the OCS, as introduced in clause 4.2.2.

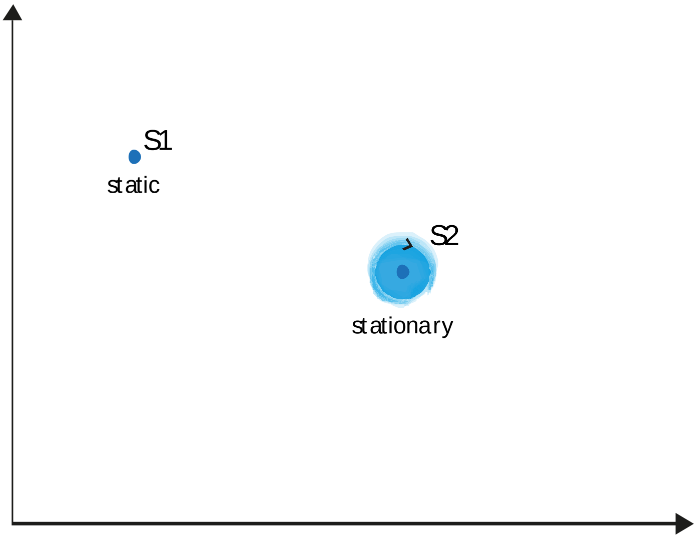

Figure 3: Explanation of static and stationary states

To describe this situation, figure 3 shows a two-dimensional state space with two entities, S1 and S2.

S1 and S2 represent two types of states. A state which is constant over time (S1) is called *static*. A state which fluctuates over time, around an identifiable point in the state space is called *stationary* (S2).Independent of the actual shape or distribution of values, the essential property of a stationary state is that fluctuations occur within a given area, which is sufficiently small compared to the overall state space.

NOTE 2: The definition of "small" is of course somewhat arbitrary. A pragmatic definition may be that effects on the ONP are small against measurement errors in determining these properties.

The time scale of fluctuations is also an important characteristic of a stationary state. It will depend on both the properties and capabilities of respective control elements and the characteristics of the decision processes in operation.

For later reference, state changes are called "microscopic" if they do not have a practical effect on ONP, and "macroscopic" if they do.

In this state picture, instability either means large cyclical or chaotic fluctuations of the OCS with observable effects on ONP, or a network state which is pulled towards some state with unwanted (unusable) ONP. Clearly, to determine the temporal behaviour requires time which is - in addition to statistical reasons on sample number - the reason why such measurements need appropriate time spans to perform.

4.2.4 State Transitions and Attractors

After having introduced the concept of static and stationary states, the question is how a NUT might change its state in the course of the adaptation process. For illustration see figure 4.

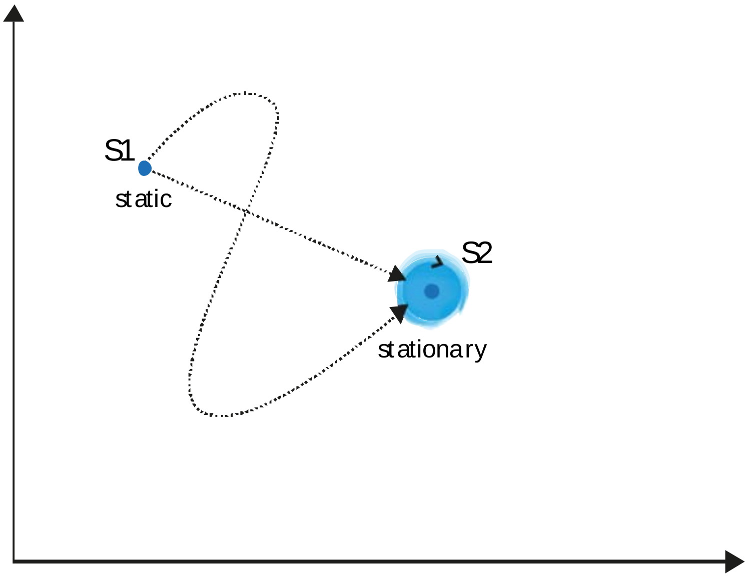

Figure 4: Examples of state transition paths

Here, S1 is the state before the adaptation starts (initial state), and S2 is the state after the adaptive process is completed (end state). The figure shows two paths from S1 to S2, a direct one and an indirect one. The actual path depends on the adaptive algorithms being used. Even if it appears unlikely that a NUT actually shows a behaviour as the one shown in this example, it cannot be excluded either. It may be the consequence of restrictions in network resource control or of actual properties of the algorithms used. Also, it is conceivable that such a behaviour is, in distributed adaptive networks, the result of interplay between "local" actions.

From the association between internal network configuration (OCS) and network performance (ONP), it follows that during a state transition, the QoS of the network may be degraded. From the testing viewpoint, this has to be considered too. While such a temporary degradation may be unavoidable in general, the impact as seen from a network subscriber's perspective will depend on its duration and seriousness. Therefore, respective properties need to be considered in the functional targets and assessment procedures used in testing. For example - in case of comparative testing or benchmarking of two ANs, one candidate may exhibit a faster adaptation process, or an adaptation towards a better end state while exhibiting a more serious or longer period of degradation than the other.

Basically, a network can have any state physically or technically possible, i.e. the initial state can be any point in state space. If an adaptive process sets in, the state will - if the NUT is not unstable - move towards the end state. As the adaptive process is actually an optimization of network parameters, there will be a finite number of end states, each of which represents a local optimum, or the global optimum of the network with respect to the targets given by administrative policies and current operating conditions. Figure 5 shows an example.

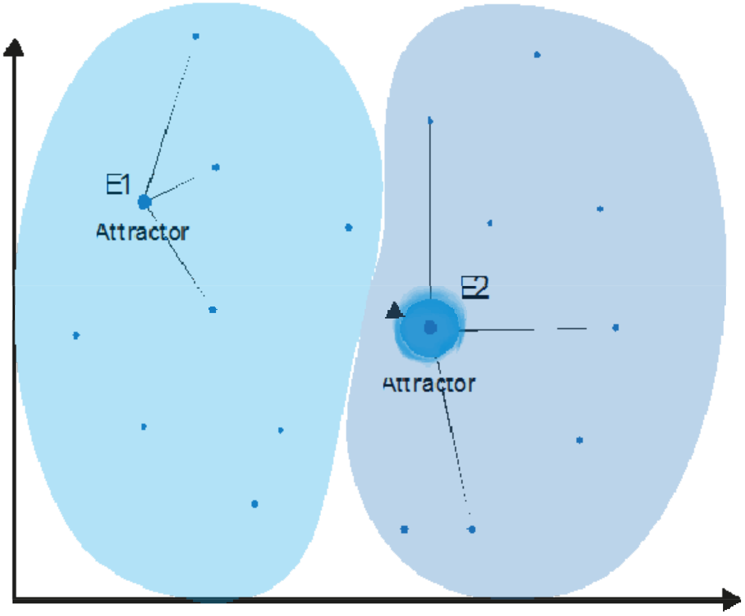

Figure 5: State space with local optima shown as end states

The points represent examples for initial states. If the NUT is brought into one of these states - and if such a state does not already represent a stable or stationary state - and adaptation is enabled, the adaptation will lead, according to the adaptation algorithm, to an end state. The concept of optimization implies that most of the possible states are not optimal with respect to the rule set applied. Therefore, a transition from the initial state to a "better" state will occur. Under the assumption that there are multiple (local) optima, the state space will have regions of initial states which lead to different (optimized) end states.

NOTE: The state space may have characteristics which do not allow a direct path from a given starting point to the global optimum of the system, by applying an incremental (mathematical, e.g. gradient based) optimization algorithm.

Initial states in the left region lead to the end state E1; initial states in the right region lead to end state E2. In analogy to the use of this term in other areas E1 and E2 are called *attractors* of the system.

The shape of the attractor landscape may have considerable effect on the dynamic behaviour and the predictability of the system in the field. Assume a situation with a complex or rather fragmented attractor space. Two starting points which are close to each other in the state space, may have attractors associated with quite different network configurations. In a laboratory environment the degree of control over the starting points is higher compared to operational networks. In effect, this may limit the ability to predict which configurations will be reached in actual operation.

The situations described above are idealized by assuming that during the time an adaptive process is taking place, the conditions which had caused this adaptation remain constant. If conditions change during this transition, and considering a fragmented attractor space, there may be a high probability that the system is oscillating between end states with probable negative effects on QoS. The test strategy should define means to detect such situations.

4.3 Adaptive Networks as Network Under Test

Testing ANs, implies testing a system of AFs that operate towards meeting functional targets defined by the network policies. The scenario can be compared to traditional interoperability testing, where the goal is to verify the end-to-end functionality (as experienced by a user) of several Functions Under Test (FUT).

Individual AFs typically pass through a conformance testing procedure at design time. However, AFs may be coupled and interact during operation, leading to situations that were not anticipated beforehand. Consequently, testing individual (or subsets) of AFs may not guarantee proper end-to-end functionality of the AN, unless it can be ensured that the tested function (or subset) is independent from other AFs that operate in the same AN.

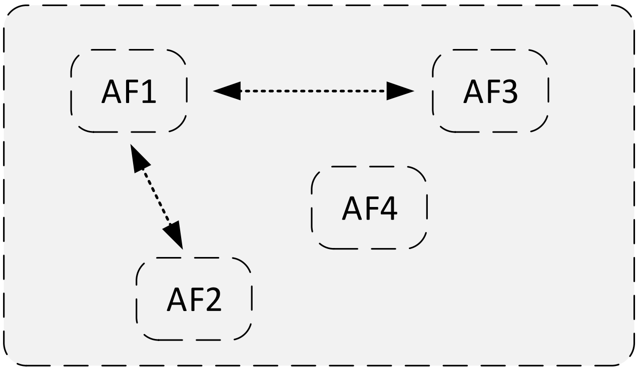

Figure 6: Example of NUT comprised of several AFs and potential interactions

The complexity of the NUT is given by the number of AFs it consists of, the hierarchical level at which they operate and are aggregated and also the time scales at which they operate. AFs may be standalone or interconnected with other AFs.

For testing purposes, it may be helpful to split a complex NUT into smaller segments. However, it is essential that any split does not impact control loops, in order to avoid altering the dynamic behaviour of the NUT. Potential criteria that may be used to segment an NUT are:

- Hierarchical aggregation level - the adaptive NUT will be tested only at a specific aggregation level.
- Time scale - the adaptive NUT will be tested only for adaptive functions that operate on a certain time scale, e.g. slow control loops.
- Functional target - the adaptive NUT will be tested only towards a certain number of functional targets.

However, as discussed above, testing different segments of the NUT may not be equivalent to testing the NUT from an end-to-end perspective.

5 Entities and interactions

5.1 Overview

Figure 7 shows the principal testing environment for adaptive network testing. It consists of:

- the network under test (NUT);
- effectors which constitute the stimuli for testing;
- sensors which provide information about the NUT;
- optional monitoring points for internal NUT information.

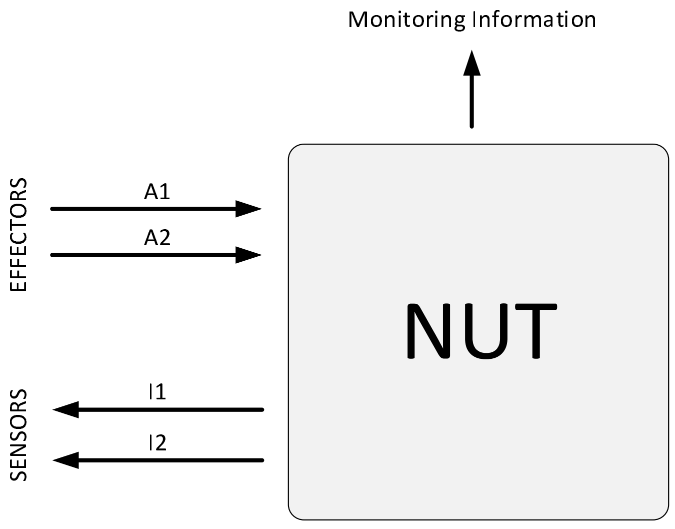

Figure 7: Testing environment

The test control domain is not shown in figure 7. The NUT is treated as a solid "black box" from a dynamic (behavioural) point of view; see clause 4 for assumptions about its inner structure. With respect to the (logical) architecture, assumption is that it can be described, on an abstract level, by a generic model, e.g. GANA. Given the nature and current state of development of ANs, any further assumptions about architectural or structural details should be avoided as these may be misleading.

On the effector side, activity type A1 denotes activities which are equivalent or identical to those coming, in real network operation, from end users. They include all types of traffic that can be applied to the network, e.g. audio or video calls (e.g. VoLTE or legacy telephony) as well as usage of data services such as Web Browsing, video streaming and other types of packet data based activities. A1 activities may include also any form of machine-type traffic relevant to the tested network.

Activity type A2 is the category for actions towards the NUT which cannot be triggered by end users. They include structural actions such as addition or removal of physical or virtualized network elements. Also, A2 activities include policy modifications or changes in defined functional targets that cause adaptive functions to change network settings or behaviour.

Likewise, on the sensor side two general types of information are distinguished:

- Sensor information type I1 is information related to properties of the network which are visible to subscribers, i.e. QoS information such as accessibility, retainability, throughput, or latency.
- Sensor information type I2 is information which is not directly visible to subscribers but may have an informational or business value for the operator, e.g. energy consumption of network nodes, traffic load or traffic mix.

I1 and I2 can, be described as multi-dimensional vectors (where the number of dimensions is network specific). One specific instance, therefore represents the *overall network properties* (ONP) as introduced in clause 4.2.2, at the respective point in time.

With respect to elaborations made in previous clauses, information of type I2 may require additional interfaces or insight into the network's structure or operation which cannot be assumed to exist in general.

The additional output labelled "Monitoring Information" stands for detailed information on the NUT's internal state and dynamics. The availability of such information is not mandatory, for the testing methodologies described in the present document. However, such additional information, beyond the information required to test the NUT from a functional point of view can be useful for diagnostics.

5.2 Effectors/Activities

5.2.1 User-equivalent activities (type A1)

5.2.1.1 Introduction

User equivalent activities comprise all types of traffic that can be generated by an end-user of the network, e.g. traffic scenarios, auto-configuration of network devices, procedures between device and network. Typical means to create such scenarios are load, QoS testing systems. From a scenario point of view, all types of such test systems are applicable for AN testing if they support the required mix and parameter settings to fulfil the test objective.

5.2.1.2 Systems delivering the required functionality

There are different types of systems - here, meaning the respective parts of the testing environment - with respect to the way they interface with the NUT.

1) End to end systems which connect to the NUT and simulate end-user behaviour. The connection can be made using "original type" connectivity, e.g. radio connection, or respective other access types for non-radio interfaces.

2) Systems replacing one or several network elements, e.g. traffic simulators which are connected in a position equivalent to an eNB. Typically such systems are related to load simulators or protocol conformance testers.

As described in previous clauses, the main scope of AN testing is about creating trust in the algorithms driving the AN, in particular, related to stability in field operation. For any simulated network element, functional equivalence, in respect to the tested use case, needs to be established. In effect, a trust chain has to be created, similar to a calibration chain in laboratory measurements.

Another aspect to consider are physical effects on the radio level which can have a strong effect on network performance. One of such critical effect is interference between radio signals from different eNBs or created by multipath propagation. The existence, location and extent of areas where such interference takes place depend on the interplay between various parameters such as antenna tilt, direction and shape of radio signal beams in the eNB which cover a given area. If the AN's functional range includes the variation of such parameters, and the testing environment uses simplifications such as cable connectors instead of actual radio signal propagation, the dynamic behaviour of the NUT can be completely different and there is a high risk that test do not reveal related problems.

5.2.2 Structural or other activities (type A2)

5.2.2.1 Introduction

Structural activities are actions which, in a typical network, cannot be caused by users, such as removal or addition of network resources or changes in operational policies. In case of removal of network resources, this covers both intended (e.g. switching off elements to save energy in low-load time regions) as well as unintended (e.g. defect-related failure of elements) actions of this type.

In case of NFV implementations of networks, there may also be additional degrees of freedom, such as performance or capacity changes of network resources due to changes in processing power or other variable resources.

Specifically for AN, changes of functional targets of the network, i.e. change of optimization for a given usage type profile to another one, through administrative interfaces also belong to this category.

It can be expected that type A2 actions will typically be "large" events, i.e. constitute macroscopic changes on at least a local, perhaps at regional level with respect to the network architecture.

5.2.2.2 Systems delivering the required functionality

The A2 category is actually a collection of very diverse kinds of activities, and many of those occur in ways which use non-standardized interfaces. Performing structural changes on a NUT requires various degrees of access to the particular elements of the network and may involve manual actions.

For any manual action, well-designed, thoroughly documented and audited processes are required to provide the necessary level of repeatability and reliability of testing. From both a technical and a commercial efficiency point of view automation is therefore considered to be the preferred option.

5.2.3 Additional controls

Beyond the two types of activities which are used to generate the stimuli for testing, there are controls which support the testing process itself, namely:

- Enable/disable adaptiveness.
- Save current OCS to a "file" #N.
- Restore current OCS from file #N.

Enabling and disabling adaptiveness, in conjunction with establishing a given OCS, is required to put the NUT into a defined state in an efficient way, and to control the testing process. The ability to save a given OCS (as a "snapshot") is the complementary function to generate the required OCS.

These functions are understood to work on a high level, i.e. for saving and restoring an OCS no details about the dimension or composition of the elements, file formats or other internals of the NUT are required.

5.3 Information/Sensors

5.3.1 Network performance from end user perspective (type I1)

5.3.1.1 Introduction

I1 type information can be measured by suitable measurement systems from the "outside" (i.e. from the network user's side) whereas it may also come from observation points within the network. I1 information is considered to be identical with existing QoS metrics. The assumption made here is that all properties of the network which possess business value in the customer's perception are represented by QoS KPI.

The QoS KPI inventory, while having its roots in human perceptions, is also applicable to machine-type communication or may be adapted as required. Likewise, this inventory may be expanded and adjusted with the emergence of new services or network usage types.

NOTE: In the literature, sometimes also the term QoE is used to describe perceived quality. Currently, there is a certain fuzziness in the standard literature with respect to the boundaries between QoS and QoE. For instance, Recommendation ITU-T P.10/G.100 Amendment 2 [i.3], defines QoE as "The overall acceptability of an application or service, as perceived subjectively by the end-user". Recommendation ITU-T E.800 [i.4] does not mention the term QoE, but defines four aspects of QoS, one of them being QoSE, defined as "A statement expressing the level of quality that customers/users believe they have experienced.", with note 1 adding "The level of QoS experienced and/or perceived by the customer/user may be expressed by an opinion rating." For the purpose of the present document, and in line with these definitions, only the term QoS is used. Further assumption is that QoE is related to QoS in a deterministic way, e.g. by applying a mapping function which expresses the subjective perception of the respective technical quantity.

5.3.1.2 Systems delivering the required functionality

To obtain information on QoS level, standard test and measurement systems can be used. Such systems typically provide both stimulus and the measured information. However, these systems need to have the required interfaces (Points of Control and Observation) to be integrated in automated testing environments, i.e. interfaces to control testing activities and to deliver measured data.

Assuming that I1 type information is also used as part of an AN's input for respective decision processes and therefore the NUT possesses respective sensors, such information may also be obtained from the NUT itself by means of respective interfaces. From a cost or effort point of view this approach offers advantages. It needs, however, careful consideration with respect to reliability of information.

5.3.2 Additional information about the network (type I2)

Sensor information type I2 is information which is not visible to subscribers but may have an informational or business value for the operator.

This may be information on the internal states of the network or the logical structure of the network. For example, on a structural level two entirely different configurations having exactly the same QoS profile may exist, which differ in operational cost.

5.3.3 Additional aspects of sensors

Network instability can lead to increased volatility/variance of values. When considering I1 type values seen as equivalent to QoS data, such data typically represents average over a certain number of samples. So individual values in output data used for QoS may need to be considered. To capture this information no additional sensors are needed. The sensor is still the same (QoS information) but not just the averages/aggregated values but also the statistical properties of the samples, (e.g. quantiles, variance or temporal trends) are evaluated.

6 Functional Targets

6.1 Introduction

This clause describes and categorizes functional targets in the context of testing adaptive networks. The methodology described should be flexible and therefore applicable to all stages of an adaptive network's lifecycle, i.e.:

- deployment, when the network is set up and optimized for the first time;
- converged, when the network is online and serves its subscribers;
- recovery, when the network takes measures to overcome an incident;
- reconfiguration, when the network adapts to new requirements.

However, it should be noted, that in each stage, different requirements will need to be met. The stable states that describe the final network condition that the network needs to reach at the end of a test run may vary in complexity. They may be defined in higher detail in the operation optimization or reconfiguration stages compared to the recovery stage where certain aspects (e.g. time needed for recovery) can become the main focus of the test.

6.2 Network stages

It is assumed that any adaptive network progresses through the same stages. Functional targets will focus on the transition between those stages.

1) Deployment

   Roll-out, the network is set up and turned on. The network reaches an operational stage fulfilling certain criteria described by an initial set of functional targets e.g. related to network auto configuration. Deployment also covers virtual deployment, in the sense of creating new logical network instances using existing resources, e.g. network slicing. Network optimization has not taken place yet.

2) Initial optimization

   The network has reached an operational stage for the first time. Based on the traffic put onto the network, the network optimizes its performance to fulfil the functional targets for operation.

NOTE 1: The traffic mentioned may either be test traffic or traffic generated by real subscribers using the network's functionalities.

3) Converged

The converged stage is considered a stable stage. It is first reached after deployment and initial optimization. During operation the network reacts to internal and external inputs with the objective of returning to this stable stage. The following list gives examples for inputs that may force the network to leave the converged stage entering an intermediate stage as described in figure 8.

1) Maintenance

A new software upgrade is applied to (parts of) the network, changing behaviour and performance of components or resources.

2) Addition/Extension

Additional network elements (e.g. an eNB) are added to the network or inactive elements are activated.

3) Self-healing

An incident (e.g. removal, failure, degradation of a component) occurred making part of the network non-functional with respect to the defined functional targets.

NOTE 2: The failure of components refers to failure in the components controlled by the AFs, not in the failure of the controlling AFs themselves. However, the failing component may contain local AFs.

4) Adaptation to meet functional targets

The conditions affecting the network performance change (e.g. intra-day traffic profiles, longer term traffic type structure changes).

5) Changing functional targets

The functional targets are modified for parts or the totality of the network's functionalities (e.g. restrictions to power consumption are applied, changes of optimization objectives are mandated).

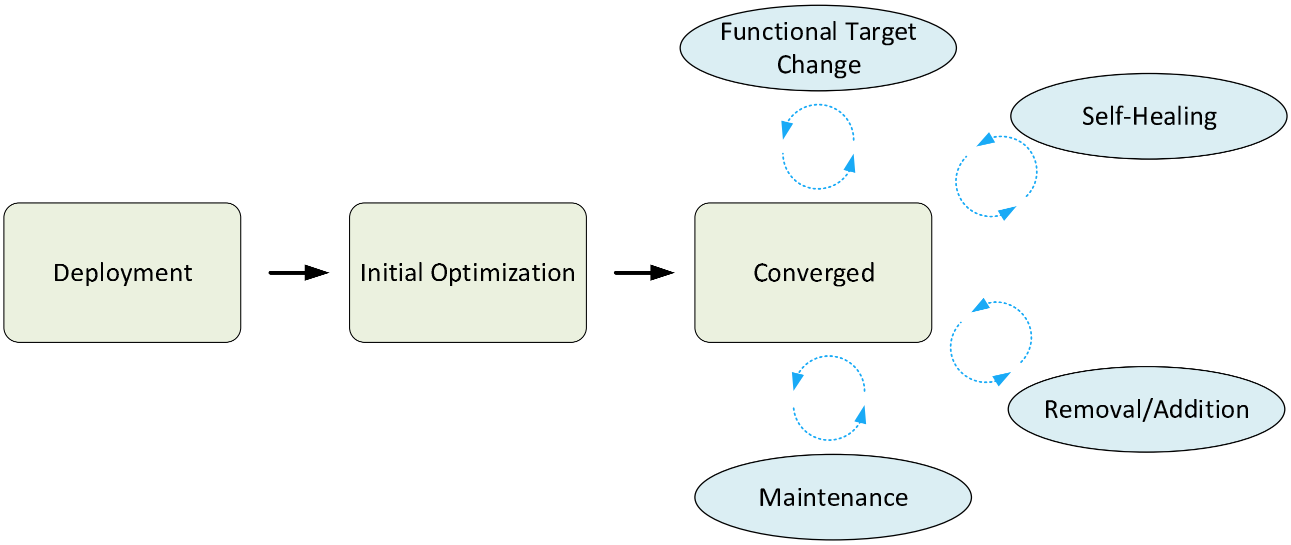

Figure 8: Stages related to AN testing and potential intermediate stages after reaching convergence

6.3 Classes of functional targets

The present clause categories functional targets to be met before, during and after the network progresses through adaptive phases by defining different classes.

- Service accessibility (from end user perspective)
  - Ability to register (rate of successful registrations)
  - Ability to setup connections (rate of successful access attempts)
- Service retainability (from end user perspective)
  - Sustainability (keep ongoing connection active)
  - Mobility
- Timing
  - Time to go back to converged stage after an unpredicted event (e.g. repair, self-healing)
  - Time to go back to converged stage after a planned event (e.g. operator triggered reconfiguration)
  - Time to get to converged stage at deployment
  - Time to get to converged stage at initial optimization
  - Time to go back to converged stage after moving to an intermediate stage, e.g. self-healing, maintenance
- Quality/Performance (from end user perspective)
  - Stability of connections
  - Stability of QoS/QoE values within a defined interval
- Quality/Performance (from network perspective)
  - Stability of connections (number of lost calls/transactions)
  - Stability of QoS/QoE values within a defined interval (e.g. service KPI required by an SLA)
  - This may include several viewpoints, both the view focused on single services and an overall view taking into account the totality of all deployed services and the potential trade-offs occurring from the optimization of one service type at the cost of another
  - Fidelity against network KPIs (e.g. number of active users, number of concurrent connections)
- Cost of the adaptive process
  - Physical resource consumption, e.g. number of additional eNBs needed
  - Processing resource usage, e.g. compute, store, network capacity
  - Energy consumption

NOTE: Lower-level targets such as protocol conformance are out of scope of the present document. Those targets are to be treated as minimum requirements which have to be fulfilled before an adaptive network is deployed.

6.4 Applicability of functional targets to network stages

Table 1 shows which functional target is applicable to which network stage.

Table 1: Matrix Functional targets / Network stages

Functional target / Network stage	Service access	Service retain	Timing	Quality/Performance End User	Quality/Performance Network	Cost
Deployment			x			x
Initial Optimization	x	x			x	
Converged	x	x		x	x	
Maintenance	x	x	x			x
Removal/Addition	x	x	x			x
Self-Healing	x	x	x			x
Functional target change			x	x	x	x

7 Generic Framework and Methods for Testing Adaptive Networks

7.1 Basic Assumptions

At this point in time, high-level models of adaptive networks such as GANA exist. However, the instantiation of this type of models on reference architectures, e.g. 3GPP, or the set of decision algorithms to be utilized is not precisely defined. Consequently, the testing methodology is described at a level of abstraction that allows it to be applicable for the anticipated spectrum of AN implementations.

For the definition of this framework, the following assumptions are made:

- Depending on the functional target under evaluation, there is a lower limit to the size or complexity of the NUT in order to achieve meaningful test results.
- Testing ANs requires that the test environment recreates the same conditions that trigger an adaptive response, according to the type of adaptive functions deployed in the NUT. This may include creating high load regions generating certain types of traffic, inserting/removing network elements, simulating loss of coverage, etc.
- Based on the assumption that adaptive reactions need a certain threshold duration of conditions (with a number of different context events needed for the complex event processing), there is a macroscopic delay between applying stimuli (test patterns) to the network, and its response. This delay needs to be taken, as respective parameter, into account when designing test conditions.
- Conventional high-load testing (e.g. signalling-level traffic generation/simulated users) may not be sufficient to produce the required level of trust in stability and general fulfilment of the functional targets when the network goes live.
- The access to elements within the NUT is restricted with respect to existence of sensor and effector interfaces. It is however also assumed that this does not constitute an actual disadvantage as using extended interfaces poses the risk of too narrow testing, as these interfaces may not be available in all types of ANs.
- It is however deemed to be necessary that an AN provides a minimum set of control functions. Also, a minimum functionality for observing the behaviour of the adaptive network is required.
- It is assumed that testing is based on a set of functional requirements which is equivalent to a set of compliance questions as used typically by network operators when selecting equipment. These functional requirements should describe the standard behaviour that is expected from the AN during normal operation.
- Testing an AN requires that the NUT provides mechanisms to be managed and controlled, e.g. governance interface.

NOTE: Testing individual AFs is out of scope of the present document. However, NTECH-AFI developed a comprehensive testing framework covering AF-level testing - see clause A.3.3.

7.2 General aspects and related terminology

A typical test case for an adaptive network consists of the following phases or steps:

- Bring the NUT into the initial state prescribed by the definition of this test case. As this may by itself be a dynamic process, it is assumed that a set of technical criteria is required which defines how the arrival at this state is indicated. After having reached this state, the sensor information (values of type I1 and I2) is taken.

NOTE 1: The AN decision algorithms may maintain an internal history or some other type of statefulness. In this case, establishing a defined situation may either mean to clear this memory or to pre-set the memory with a defined set of information. This does not mean that all relevant state variables or other details have to be directly accessible, but that respective functionality to achieve this goal has to be available.

- Apply the prescribed set of stimuli (set of activities of types A1 and/or A2).
- Wait until a prescribed amount of time has elapsed, and/or indication that the adaptation process is completed is present, i.e. the final state of the NUT is reached.
- Collect sensor information (types I1 and I2) to determine the NUT's properties (ONP).
- Evaluate correctness of ONPs and assign a test verdict.

NOTE 2: See clause 5 for the definitions of A1, A2, I1 and I2.

Unlike other types of testing such as protocol conformance testing, each phase needs a macroscopic amount of time to complete, according to the set of basic assumptions (see clause 4). In the case of QoS-type information, and most probably also for other types of sensor information, data is "noisy", requiring a certain number of samples taken to reach a desired level of accuracy. Also, assumptions on the dynamics of adaptation processes lead to the necessity that respective input stimuli have to be present for a certain amount of time.

According to the initial outline of motivation and functional targets of testing, assurance of stability is of topmost importance in the range of these targets and is a central element in testing.

NOTE 3: See clause 6 for the definition of functional targets related to stability.

7.3 Testing Process

7.3.1 Introduction

As described in previous clauses, and explicitly with reference to clause 5, there are two types of actions which can be applied to the network in the course of testing. It is expected that in practice some test cases also use a combination of such actions. For example, a test case covering the failure of a network resource can be a combination of a background load scenario (type A1) with the removal of a particular resource (type A2).

7.3.2 A1 based testing scenarios

A1 based test scenarios model the - usually concurrent - activity of several real or simulated subscribers. The actual form of such scenarios is, in the current context, not of concern. There are, however, some requirements which need to be fulfilled in any case.

In most cases the scenario is expected to consist of a number of (actual or simulated) UEs creating a background activity "floor" of a given traffic mix, and a smaller number (probably only one UE in many cases) doing actual measurement of the same type and structure as in typical QoS testing.

An important requirement is reproducibility of the scenario. Reproducibility implies that all information to repeat tests in the same way need to be documented. Also, systems used to generate such scenarios (e.g. load generators) need to have respective properties to ensure repeatability. The actual amount of information depends on the type of activity. As far as activity-generating systems resemble QoS testing systems, ETSI TS 102 250-4 [i.2] gives some practical examples for the extent of parameters to be documented and to be controlled by the system.

NOTE: This requirement can easily be underestimated. For instance, a typical smartphone carries out several background activities using packet data transfer (e.g. checking for new app versions, responding to push messages by accessing servers, etc.) which can generate considerable traffic in parallel to controlled activities on the device.

Typically, testing based on A1 type effectors includes parallel activities to create a given traffic pattern and a certain load level. Active measurements using multiple probes are also a means to achieve a higher yield of data samples to improve statistics or to shorten the period of time required to achieve e a given level of statistical certainty.

Repeatability demands that in subsequent test runs, basically similar activity patters are created. In parallel activities, this could be achieved by using fixed timing relations between individual sequences of activities. A fixed activity pattern however carries the risk that the space of operating conditions is not fully covered as there is a fixed correlation between activities. Therefore, a certain randomness of relative timing is required to create activities which are statistically independent of each other as it is the case in actual real-subscriber activities.

From these considerations, the actual kind of randomness needs to be defined or documented well enough to allow assessment of potential shortcomings.

In conclusion, when a test scenario is designed, the various aspects mentioned above should be weighed against each other to create a solution which is functionally appropriate at a reasonable effort.

7.3.3 A2 based testing scenarios

A2 based test scenarios model actions or events which, in a typical network cannot be produced by end users. An A2 based test scenario (see figure 9) consists of these steps:

- Establish the initial state of the NUT and take a sufficient volume of measurements to establish I1 and I2 values as the base line for subsequent analysis.

- Apply the A2 actions and wait for the prescribed time to allow AN functionality to take effect. The time can either be fixed by some vendor or customer specific definition, depend on indicators for AN activity (i.e. reaching a static or stationary state indicated by presence or absence of respective indicators), or be defined as a combination of both (allowing for earlier completion signalled by said indicators but with a maximum time allowed). It is assumed that here customer-specific compliance requirements exist.

- Take I1 and I2 measurements to establish the final network properties (ONP) values for assessment of the NUT's behaviour.

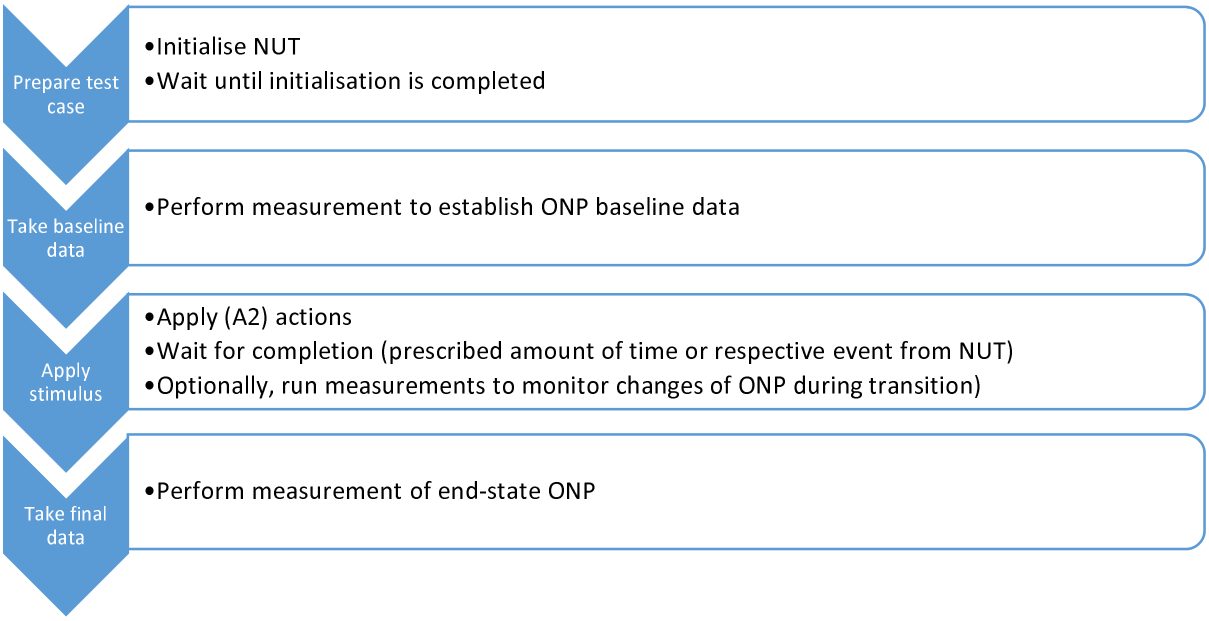
Prepare test case
•Initialise NUT
•Wait until initialisation is completed
Take baseline data
•Perform measurement to establish ONP baseline data
Apply stimulus
•Apply (A2) actions
•Wait for completion (prescribed amount of time or respective event from NUT)
•Optionally, run measurements to monitor changes of ONP during transition)
Take final data
•Perform measurement of end-state ONP

Figure 9: Typical A2 based AN test scenario

Examples of A2 based scenarios, with adaptive actions from the AN, are the removal or failure of an eNB in a LTE network.

In case of removal or permanent failure, the eNB stays out of service. The expected reaction of the network can be a reconfiguration of the radio access network in the affected area in order to compensate - as best as possible - for the loss of coverage. This can include:

- Update the neighbour relations for the remaining eNB.
- Change physical parameters (e.g. antenna tilt, direction or angle - to optimize coverage for the affected area.
- Change resource allocation policies to compensate for the loss of air interface capacity.

In case of temporary failure, the eNB may try to restart (reboot), being only temporary out of service. In this case, the reaction described above may take place initially and may be followed by a subsequent reaction when the eNB comes back into service.

7.4 Evaluation of results

A test produces "N" indicators corresponding to "N" functional targets. The test verdict is computed from the totality of the indicator-target pairs.

In the simplest case, there is a Pass or Fail verdict for each indicator-target pair and these verdicts are combined AND wise. The case of Inconclusive, as provided by ISO/IEC 9646 [i.5] relates to an indicator which cannot be interpreted. On the end result level, assuming that the AND connection one Inconclusive verdict renders the end result Inconclusive only if the remaining indicators compute to Pass, otherwise the end result will be Fail.

In more elaborated schemes, functional requirements can be grouped into categories of different priority levels. For the highest category, e.g. mandatory or critical, the combination typically is AND, while other categories have associated weights. For instance, a failure to meet stability criteria should lead to the overall verdict of Fail, while different combinations of QoS KPIs versus Cost may be expressed by scores within a defined range.

NOTE: The categorization of the hierarchy of the functional targets and associated weights is considered a non-trivial task and may be very dependent on the type of NUT, the network stage, the function to be tested and the testing context (e.g. long term versus short term).

Annex A: Relation to other work done in this field

A.1 Introduction

This clause discusses work relevant to the scope of the present document, which is being done by other standardization groups that address adaptive networks.

A.2 ISG NFV

A.2.1 Group description

The Industry Specification Group (ISG) on Network Functions Virtualisation (NFV) has a subgroup on testing (NFV-TST). At the date of the publication of the present document the group has published ETSI GS NFV-TST 001 [i.6], on validation of NFV environments and services for pre-deployment testing. A further work item, ETSI GS NFV-TST 002 [i.7] addresses test methodologies on interoperability testing.

A.2.2 Network Functions Virtualisation (NFV)

Network Functions Virtualisation (NFV) is an important concept that falls under the adaptive networks umbrella. Essentially, any network that implements NFV can be seen as an adaptive network, since the network structure and resources are automatically controlled, in order to meet functional targets. However, it should not be assumed that all adaptive networks have to have virtualized components. One example are traditional networks which have adaptive functions that operate on physical network elements in the radio access network, e.g. coverage and capacity optimization (CCO) mobility robustness optimization (MRO). This type of adaptive functions are grouped under the 3GPP term of Self-Organizing Networks (SON).

From a test methodology perspective it is important to understand the relationship between AFs, and physical or virtual network functions. Two types of AFs can be distinguished:

- Generic AF - AFs that operate independently from virtualization, i.e. the functions behave the same regardless if the network attribute they control is part of a physical or virtual network element.
- Virtualization AFs - AFs that are inherent to the virtualization infrastructure and control the attributes of this infrastructure, e.g. automatic control of processing resources.

A.3 NTECH AFI

A.3.1 Group description

The Technical Committee (TC) Network Technologies (NTECH) has a working group on evolution of management towards Autonomic Future Internet (AFI). The group has started a work item to study design guidelines and testability for building confidence in autonomic functions. NTECH AFI also defined the Generic Autonomic Network Architecture (GANA) model in ETSI GS AFI 002 [i.1].

A.3.2 GANA model overview

The Generic Autonomic Network Architecture (GANA) model provides a general framework that can be applied to describe autonomic behaviour in any implementation-specific network architecture, e.g. 3GPP. It specifies the functional blocks, interactions and operational principles that allow automatic functions. AFs in GANA are referred to as Decision Elements (DEs). They control network parameters or resources and perform Self-* functions, e.g. self-configuration, self-optimization, self-healing. The GANA model can be used as a general reference when describing an adaptive network.

GANA also describes interfaces through which a network administrator can interact and control DE's at different levels in the network and also the functional block that maintains a high level overview of the network, both providing value from a testing perspective. However, ANs may not always follow the GANA architecture, thus the model should not impose any limitations in respect to testing methodologies.

A.3.3 Concepts of the Generic Test Framework for Testing Adaptive Functions

The scope of the present document covers testing methodologies of ANs, as a whole. It does not address testing the functionality of individual AFs that operate in the AN. NTECH-AFI has provided a generic test framework for testing AFs, which is reproduced in the present clause.

The framework identifies different types of test systems that could be employed to the problem space of testing AFs, and are to be applied in phased testing starting at design time up to the point when a network consisting of trusted and certified AFs is tested as a whole (for integration and user acceptance testing). The following topics are addressed:

- What type of testing is performed for an AF during design time and who performs the testing and owns test components used?
- What role Testing plays in the following phases of an AF lifecycle?
  - Validation of an AF (or collective interworking AFs);
  - Trustworthiness building on an AF (or collective bundle of interworking AFs);
  - Certification of an AF (or collective bundle of interworking AFs).
- Where Conformance Testing comes into play, and where Interoperability Testing comes into play?
- Where Integration and User Acceptance Testing of an Adaptive Network comes into play?
- Criteria/basis for assigning verdicts in Test cases employed at various phases of Testing AFs?
- The need for the Test Systems or Test Components that test AFs to be intelligent (i.e. themselves being autonomic-like) as to mimic AFs themselves?
- Where Passive Testing plays a role and where combined Active and Passive Testing plays a role?

Using knowledge from reference models for adaptive networks, such as the GANA (which incorporates and unifies/fuses concepts from various leading autonomics models, including FOCALE (Strassner, Agoulmine, & Lethihet, 2006) [i.11], IBM-MAPE (Dar) [i.8], (IBM, 2005) [i.9], 4D architecture (Hayan) [i.10], Knowledge Plane for the Internet (Clark, Partridge, Ramming, & Wroclawski) [i.12], and other models, as a unified holistic Reference Model for Autonomic Networking, Cognitive Networking and Self-Management (AMC technology), as well as research efforts on Testing and Validation for Autonomic and Self-Managing Systems, the following aspects can be deduced:

1) According to Reference Models such as the GANA: Autonomics introduce Functional Blocks (FBs) and their associated Reference Points that are specific to enabling to implement autonomics (AFs and the enabling components) in a target network architecture such as the 3GPP architecture. The implication of this is that Conformance Testing and Interoperability Testing is required on the Reference Points for Autonomics instantiated in a target architecture and environment. The reason is that the various FBs for autonomics may come from different vendors/suppliers of those autonomics specific FBs. The GANA defines various reference points whose instantiation in target architecture and environment calls for conformance and interoperability testing.

2) Individual AFs or their composition into Autonomic Systems need to undergo the following processes in the lifecycle: Validation**,** Trustworthiness-building, and then Certification**.** Providers/suppliers of AFs are responsible for performing these processes. Deployability of an AF should be based on the condition that the AF passed all the processes up to having been certified. Figure A.1 illustrates the processes.

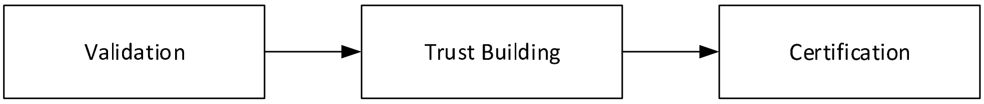

Figure A.1: Potential AF certification process before inclusion in ANs

3) AFs may be designed as run-time loadable or replaceable software modules (better AFs in terms of quality of decision-making capability may be used to replace low quality AFs), and may be deactivated and activated.

4) Network Scope (Domain) for which a supplier of AFs and their embedment in network equipment or some host platforms should be clarified by the supplier of the AFs. The scope could be the whole network segment and its management architecture (e.g. core network and the associated management architecture, backhaul and it management architecture, RAN and its management architecture, or a larger edge to edge or end-to-end (E2E) network and its management architecture).

5) The Self-* features realized by a particular AF, as well as "claims" on what the AF strives to achieve during its operations, with indications on the metrics (e.g. KPIs) that can be measured and monitored, appearance/manifestations of new instances of objects the AF causes to be created, or change in state of certain objects impacted, should all be used to verify the claim and should be described by the AF provider and made known to the tester.

6) Testing of AFs involves various techniques and approaches, ranging from:

   a) Integrated self-testing within an AF (i.e. embedded testing using a test component embedded within the AF) as shown in figure A.2.

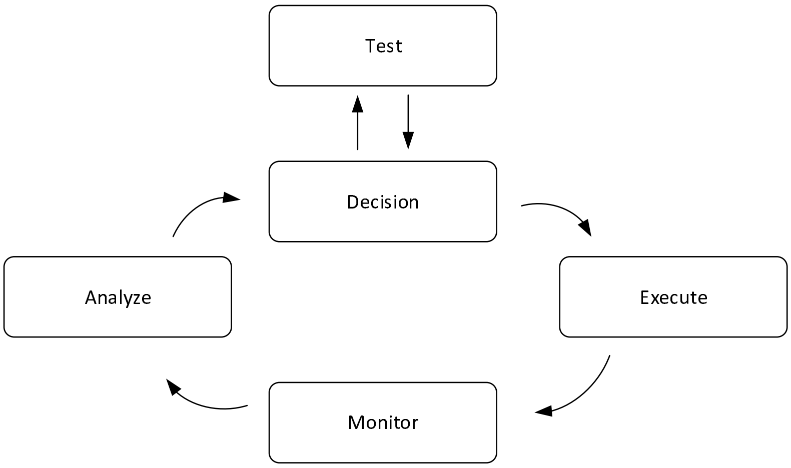

Figure A.2: Self-Testing concept for AFs

   b) Testing a collective group/bundle of interworking AFs as a black box (applies especially to AFs within nodes).

   c) Testing system/component may intercept and observe actions of the AF under test that are performed in response to stimuli (mainly based on the operating region of the AF's control-loop) and use the actions in inferring correctness of the action (depends on the intelligence and correctness of the test component's algorithms it employs in the testing). This applies to environment in which AF actions can be intercepted during active testing (with injection of stimuli data to the AF under test), e.g. in tests conducted by the AF owner.

d) Passive testing may be used by the test system to observe the metrics (KPIs) and the objects that may be instantiated or intentionally impacted by the AF, using monitoring techniques and inferring whether the changes are desirable for meeting the objectives of the network and claims made about the AF's impact on the monitored metrics and/or objects instantiated or whose state gets modified intentionally by the AF's actions. The approach could be as follows: a set of metrics (e.g. KPIs) determined to be critical to be observed is derived based on impacts the AF is claimed to positively have on the metrics or observable objects (such as services or service nodes),and base acceptable values for the metrics or state of objects are first established, and then the test system passively monitors the metrics and objects over time while the environment and the workload around the AF is known to be changing over time, and then the test system/component keeps tracing if and how the KPIs or objects are impacted over an observation window. Verdicts are then assigned after the sampling. The test objective may seek to determine whether the measured values improve, remain close to acceptable values.

e) Combination of active and passive testing may be employed.

7) Test and Validation Verdicts for an AF or a bundle of AFs (treated as black box) should be based on the following, depending on the testing approach used:

a) Verdict passing may be based on determining whether an intercepted action performed by an AF within a certain acceptable time that is measured relative to some event of interest to the test system/component, has significant impact on meeting the objectives required of the node or network, depending on the claims attached to the AF on what it does with respect to the objectives. The impact factor can be used in determining correctness of the action during validation of the AF. This applies to environment in which AF actions can be intercepted.

b) Verdict passing may be based on observing the impacts (metrics and/or objects instantiated or modified as a result of the adaptive behaviour of the AF under test) and assessing whether the impacts support the claims concerning what the AF is meant to achieve in its operation. This applies in environment and testing in which it may even not be possible to intercept AF actions.

c) Verdicts may be based on various criteria, such as a combination of actions, timing and impacts observed on metrics and objects of interest to the test case.

d) Because each AF may be designed to realize multiple Self-* features, such as auto-discovery and self-configuration, self-optimization, self-healing, etc., verdicts may be defined that specifically target the individual Self-* features of an AF.

8) Test Systems or Test Components that test AFs should be intelligent (i.e. themselves being autonomic-like) as to mimic AFs they are meant to test, meaning that confidence in the test system also needs to be built up over a certain time and application to various test scenarios.

9) Testing of Adaptive Networks is to be *considered as decomposed into various testing needs* and associated test systems and components, from component level testing of AF as individual software modules up to the highest level Integration and User Acceptance Testing of an Adaptive Network as a whole, which can only be done under the conditions that AFs passed all testing phases and types of testing and validation to the point of having been certified. AFs that are trusted and ideally certified should be the ones that can be made to participate in the overall Integration and User Acceptance Testing of an Adaptive Network as a whole, implying dependencies on the tests conducted in various phases of an AF lifecycle.

10) Integration and User Acceptance Testing of an Adaptive Network as a whole comes into play when individual AFs have undergone as complete as possible the whole chain of Validation, Trustworthiness-building, and then Certification. At such a stage the Test system needs to access the system boundary that is defined by all the open interfaces for control and observation exposed mainly by the AFs and their interfaces with other Functional Blocks that enable the AF to operate. Some test cases on this higher level testing may depend on passive testing.

11) Input to deriving test cases for an AF should be based on the following items:

a) Reference Points that apply to the AFs or bundled AFs to be tested.

b) The "Operating-Region" of a Control-Loop(s) associated with an AF.

c) The provider/supplier of an AF specifies what the AF is designed to achieve when running in the network (even without having to disclose the algorithms in the AF), specifying the network metrics (e.g. KPIs) that get improved by the AF or kept to a certain threshold by virtue of optimizations by the AF.

12) Test System for an AF can evolve in its test capabilities along with the need to test evolved AF algorithms.

13) Test Data may be synthetic or include in-service data involving a real environment in which AFs are being tested.

Table A.1 categorizes the types of testing and associated test Systems and components that should be applied in Testing AFs during their lifecycle.

Table A.1: Types of testing and associated deployment phases

Type of Testing	Validation phase of an AF	Trustworthiness building phase	Certification Phase for the AF	Test Network Deployment Phase	AF deployment and activation Phase	Test Network Operation phase	Test Network Optimization Phase
AF Testing and Validation	x	x	x				
Conformance Testing			x	x	x		
Interoperability Testing			x	x	x		
Integration and User Acceptance Testing of an Adaptive Network as a whole				x	x	x	x

History

Document history		
V1.1.1	August 2016	Membership Approval Procedure MV 20161025: 2016-08-26 to 2016-10-25