ETSI GS ENI 001 V2.1.1 (2019-09)

GROUP SPECIFICATION

Experiential Networked Intelligence (ENI); ENI use cases

Disclaimer

The present document has been produced and approved by the Experiential Networked Intelligence (ENI) 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

RGS/ENI-008

Keywords

artificial intelligence, management, network, use case

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

Essential patents

IPRs essential or potentially essential to normative deliverables 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.

Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

Foreword

This Group Specification (GS) has been produced by ETSI Industry Specification Group (ISG) Experiential Networked Intelligence (ENI).

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 present document specifies a collection of use cases from a variety of stakeholders, where the use of an Experiential Networked Intelligence (ENI) system can be applied to the fixed network, the mobile network, or both, to enhance the operator experience through the use of network intelligence. The present document is a revision of ETSI GR ENI 001 [i.1]. It identifies and describes additional use cases and scenarios. It also gives the baseline on how the studies in ENI can be applied as solutions of the identified use cases in accordance with the ENI Reference Architecture.

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.

Not applicable.

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 GR ENI 001 (V1.1.1): "Experiential Networked Intelligence (ENI); Use Cases".

[i.2] NGMN Alliance: "Description of Network Slicing Concept", Version 1.0, January 13, 2016.

NOTE: Available at https://www.ngmn.org/fileadmin/user_upload/160113_Network_Slicing_v1_0.pdf.

[i.3] 3GPP TR 23.799 (V14.0.0): "3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Study on Architecture for Next Generation System Release 14", December 2016.

[i.4] A. Morton, AT&T Labs: "Considerations for Benchmarking Virtual Network Functions and Their Infrastructure", July 2017.

[i.5] ETSI TS 132 101 (V11.4.0): "Digital cellular telecommunications system (Phase 2+); Universal Mobile Telecommunications System (UMTS); LTE; Telecommunication management; Principles and high level requirements (3GPP TS 32.101 version 11.4.0 Release 11)".

[i.6] ETSI TS 128 530 (V15.1.0): "5G; Management and orchestration; Concepts, use cases and requirements (3GPP TS 28.530 version 15.1.0 Release 15)".

[i.7] ETSI GR NFV-EVE 012 (V3.1.1): "Network Functions Virtualisation (NFV) Release 3; Evolution and Ecosystem; Report on Network Slicing Support with ETSI NFV Architecture Framework".

[i.8] ETSI GS ENI 002 (V1.1.1): "Experiential Networked Intelligence (ENI); ENI requirements".

[i.9] ETSI GS ENI 005: "Experiential Networked Intelligence (ENI); System Architecture; Release 1".

[i.10] ETSI GR ENI 004: "Experiential Networked Intelligence (ENI); Terminology for Main Concepts in ENI".

[i.11] IETF RFC 6645: "IP Flow Information Accounting and Export Benchmarking Methodology".

3 Definition of terms, symbols and abbreviations

3.1 Terms

For the purposes of the present document, the terms given in ETSI GR ENI 004 [i.10] apply.

3.2 Symbols

Void.

3.2 Abbreviations

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

AI	Artificial Intelligence
AP	Access Point
API	Application Programming Interface
BBU	Baseband Unit
BRAS	Broadband Remote Access Server
BSS	Business Support System
CCO	Capacity and Coverage Optimization
CGN	Carrier Grade Network address translation
CPRI	Common Public Radio Interface
CPU	Computing Processing Unit
C-RAN	Centralized RAN
DC	Data Centre
DDOS	Distributed Denial Of Service
DHCP	Dynamic Host Configuration Protocol
D-RAN	Distributed RAN
E2E	End-to-End
ENI	Experiential Networked Intelligence
FTP	File Transfer Protocol
IDC	Internet Data Centre
INFP	Intelligent Network Failure Prevention
IP	Internet Protocol
KPI	Key Performance Indicator
MANO	Management and Orchestration
MEC	Multi-access Edge Computing
MIMO	Multiple Input Multiple Output
MPLS	Multi-Protocol Label Switching
NF	Network Function
NFV	Network Function Virtualisation
NFVI	NFV Infrastructure
NGFI	Next Generation Fronthaul Interface
NGMN	Next Generation Mobile Networks
NSI	Network Slice Instances

OPEX	OPerational EXpenditure
OS	Operating Systems
OSS	Operations Support System
PHY	PHYsical layer
QoE	Quality of Experience
QoS	Quality of Service
RAM	Random Access Memory
RAN	Radio Access Network
RAU	Remote Aggregation Unit
RCC	Radio Cloud Centre
RF	Radio Frequency
RRU	Remote Radio Units
RSRP	Reference Signal Received Power
SDN	Software Defined Networking
SD-WAN	Software-Defined Wide Area Network
SLA	Service-Level Agreement
TCP	Transmission Control Protocol
UE	User Equipment
VM	Virtual Machines
VNF	Virtualized Network Functions
WAN	Wireless Access Network
WLAN	Wireless Local Area Network

4 Overview

4.1 Background

Operators see human-machine interaction as slow, error-prone, expensive, and cumbersome. For example, operators are worried about the increasing complexity of integration of different standardization platforms in their network and operational environment; this is due to the vast differences inherent in programming different devices as well as the difficulty in building agile, personalized services that can be easily created and torn down. These human-machine interaction challenges are considered by operators as barriers to reducing the time to market of innovative and advanced services. Moreover, there is no efficient and extensible standards-based mechanism to provide contextually-aware services (e.g. services that adapt to changes in user needs, business goals, or environmental conditions).

These and other factors contribute to a very high OPerational EXpenditure (OPEX) for network management. Operators need the ability to automate their network configuration and monitoring processes to reduce OPEX. More importantly, operators need to improve the use and maintenance of their networks. In particular, this requires the ability to visualize services and their underlying operations so that the proper changes can be applied to protect offered services and resources (e.g. ensure that their Quality of Service (QoS) and Quality of Experience (QoE) requirements are not violated). If such visualization could be provided, then operators would be better able to maintain their networks.

The associated challenges may be stated as:

a) automating complex human-dependent decision-making processes;

b) determining which services should be offered, and which services are in danger of not meeting their Service-Level Agreement (SLA)s, as a function of changing context;

c) defining how best to visualize how network services are provided and managed to improve network maintenance and operation; and

d) providing an experiential architecture (i.e. an architecture that uses various mechanisms to observe and learn from the experience an operator has in managing the network) to improve its understanding of the operator experience, over time.

The aforementioned challenges will require advances in network telemetry, big data mechanisms to gather appropriate data at speed and scale, machine learning for intelligent analysis and decision making, and applying innovative, policy-based, model-driven functionality to simplify and scale complex device configuration and monitoring.

4.2 Overview of the ENI System

4.2.1 Brief Description

The ENI system is an innovative, policy-based, model-driven functional entity that understands the configuration and takes actions in accordance with changes in context, such as the environment, the dynamic demand of the resources, and the varying service requirements. By exploiting emerging technologies, such as big data analysis and artificial intelligence mechanisms, and also by automating (where possible) complex human-dependent decision-making processes, the ENI system enables intelligent service operation and management, and provides the ability to ensure that automated decisions taken by the system are correct and are made to increase the stability and maintainability of the network and the applications that it supports.

Examples of the possible functionalities of an ENI system are given in Figure 4-1.

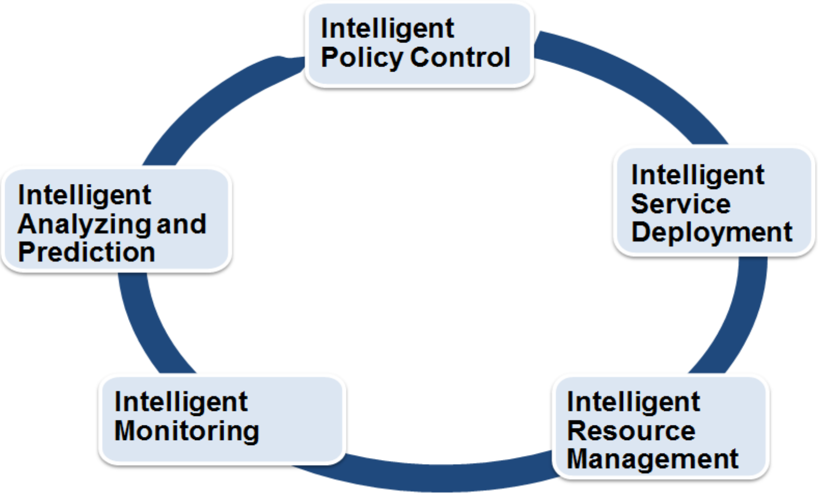

Intelligent Policy Control
Intelligent Service Deployment
Intelligent Resource Management
Intelligent Monitoring
Intelligent Analyzing and Prediction

Figure 4-1: Example of functionalities of ENI system

4.2.2 Expected Benefits

ENI system delivers enhanced customer experience by allowing operators to understand the operating status of their network and networked applications in near-real-time, and reconfigure their network. The ENI system automatically collects network status and associated metrics, faults, and errors, and then uses artificial intelligence to ensure network performance and quality of service are met at the highest possible efficiency (e.g. with the minimum required resources). An ENI system can also be used to find bottlenecks of service and/or failure of network. Both of these benefits are done on-demand, in response to changing contextual information.

The ENI system helps to increase the value of services provided by an operator to its customers by rapidly on-boarding new services, enabling the creation of a new ecosystem of cloud consumer and enterprise services, reducing Capital and Operational Expenditures, and providing efficient operations.

5 General use cases

5.1 Introduction

This clause describes the use cases and scenarios identified by the ENI ISG. Each use case includes a description of how an ENI system can be applied, and the benefits it provides. Examples to show how the mapping of the Reference Architecture of ENI, specified in [i.9], can be done through a few different use cases, are also given. It is noted that such mapping, including the reference points and roles of functional blocks, is not the specification of the implementation of these use cases, but should only be seen as examples and based on the current ENI Reference Architecture [i.9]. It is also noted that the applicability of each functional block in terms of what is its role in the overall implementation of the Use Case can only be seen as an example, based the current ENI Reference Architecture [i.9]. When the reference Architecture [i.9] changes, and overall doubts have been solved by architecture experts, text on quotes and particular interpretation of the contents may be modified.

A list of the use cases included in the present document are categorized into the following four categories (Table 5-1):

1) Infrastructure Management: This category of use cases covers the processes related to the management of the network infrastructure (e.g. adjustment of allocated and provided services, maintenance, capability specification, and planning). In particular, it is about using policies for managing the network infrastructure, enabled by placing analytics in the control loop and using the results of the analytics as part of the input to policy-based management of the infrastructure.

2) Network Operations: Use cases described in this category are concerned with running the network, where the runtime contexts of the network are extracted and analysed, and the management operations are performed and optimized dynamically at runtime.

3) Service Orchestration and Management: This category of use cases relates to the service and order management, covering processes such as activation using the operator's business channels or customer portals. It is about providing differentiated SLAs for different applications, including vertical applications, through the application of machine learning in an intelligent entity, i.e. ENI. For example, services can be differentiated based on level (e.g. gold vs. silver vs. bronze classes of service) as well as based on the type of application within a level (e.g. a video streaming service has a different service than FTP, even though both are applications that a particular customer has).

4) Assurance: Use cases described in this category are concerned with the functionality of network monitoring, trending, and prediction, as well as taking policy-based actions using knowledge learned from the network to facilitate network maintenance. This includes service runtime operations dedicated to guarantee continuous service delivery.

Table 5-1: Summary of ENI Use Cases

Category					
1 - Infrastructure Management	Use Case #1-1: Policy-driven IDC Traffic Steering	Use Case #1-2: Handling of Peak Planned Occurrences	Use Case #1-3: DC Energy Saving using AI		
2 - Network Operations	Use Case #2-1: Policy-driven IP Managed Networks	Use Case #2-2: Radio Coverage and Capacity Optimization	Use Case #2-3: Intelligent Software Rollouts	Use Case#2-4: Intelligent Fronthaul Management and Orchestration	Use Case #2-5: Elastic Resource Management and Orchestration
	Use Case #2-6: Application Characteristic based Network Operation	Use Case #2-7: AI enabled network traffic classification	Use Case #2-8: Automatic service and resource design framework for cloud service	Use Case #2-9: Intelligent time synchronization of network	
3 - Service Orchestration and Management	Use Case #3-1: Context-Aware VoLTE Service Experience Optimization	Use Case #3-2: Intelligent Network Slicing Management	Use Case #3-3: Intelligent Carrier-Managed SD-WAN	Use Case #3-4: Intelligent caching based on prediction of content popularity	
4 - Assurance	Use Case #4-1: Network Fault Identification and Prediction	Use Case #4-2: Assurance of Service Requirements	Use Case #4-3: Network fault root-cause analysis and intelligent recovery		
	Use Case #5-1: Policy-based network slicing for IoT security	Use Case #5-2: Limiting profit in cyber-attacks			

5.2 Infrastructure Management

5.2.1 Use Case #1-1: Policy-driven IDC Traffic Steering

5.2.1.1 Use case context

This use case relates to intelligent link load balancing and bandwidth allocation between Internet Data Centres (IDCs). The tenants of IDCs include enterprises that have requirements that dynamically adjust service and/or resource behaviour (e.g. reliable network connectivity and changes to an offered service based on network load).

There are a number of problems with how current traffic steering is performed between IDCs. These include the use of multiple possible links between IDCs (e.g. which link is the best to use at a given time). Currently, the link for a tenant is normally determined as the shortest path between the IDC that the tenant resides in and the IDC that the tenant is connecting to. In addition, the link load is not considered when calculating the traffic path. Furthermore, the bandwidth allocated to a tenant is not always fully used.

5.2.1.2 Description of the use case

5.2.1.2.1 Overview

Operators are deploying IDCs in Metropolitan Area Networks (MANs) to provide network access with load-balancing and resiliency. Current network configuration practices include:

- In order to provide service assurance for important tenants, network administrators typically schedule the traffic in specific periods. Traditional network management is usually complex, with a long cycle caused by manual actions, so it is difficult to meet the requirement of real-time traffic optimization.

- Large service provider's traffic usually is sensitive to the events of a day. For example, online big sales and usage of social media with video steaming cause a significant increase in traffic. This means that the network administrator cannot provide bandwidth assurance for some important tenants.

- The bandwidth requirements of tenants tend to change dynamically. Traditional static bandwidth allocation leads to low bandwidth utilization and redundancy.

- The imbalance across multiple links leads to inefficient resource utilization. For example, it is possible that the utilization of a link reaches a certain threshold, while other links' loads remain low.

5.2.1.2.2 Motivation

The ENI system can be used to achieve intelligent link load balancing and intelligent bandwidth allocation. In ENI, policies can be modified by using machine learning to fill in important parameters, such as available links, link bandwidth, real-time link utilization, and other predefined constraints. Three examples of the predefined constraints to be considered before modifying the policies are:

1) each link is predefined with a threshold of the maximum bandwidth and cannot be exceeded;

2) flow of a client at a specific service level (e.g. gold) cannot be switched;

3) the maximum times of switching specific service from one link to another link is predefined and cannot be exceeded.

Such policies can be used to better manage the network and achieve autonomous service traffic monitoring and network resource optimization. It can also be used to adjust the service along different links of an IDC, thus improving the operator's experience through enhanced network resilience and service QoS and QoE.

The ENI system also:

- predicts changes by using AI in the tenant's service requirements based on historical data (e.g. the type of QoS to be provided for a given service based on the type of application and metadata);

- collects and analyses real-time data, given the service adjustment recommendations (e.g. which metadata and metrics to monitor based on the type of service and the type of changes applied);

- corrects the prediction result according to the adjustment recommendations, and converges to an ideal service management policy;

- analyses QoS and other applicable data and metadata to make the final service policy modifications; this is then stored as a reusable set of objects.

By using the above intelligent service adjustment policy provided by the ENI system, real-time, dynamic, and automated resource allocation and adjustment to the service can be achieved. The bandwidth utilization is improved. Meanwhile, it provides bandwidth assurance for important tenants according to the service level.

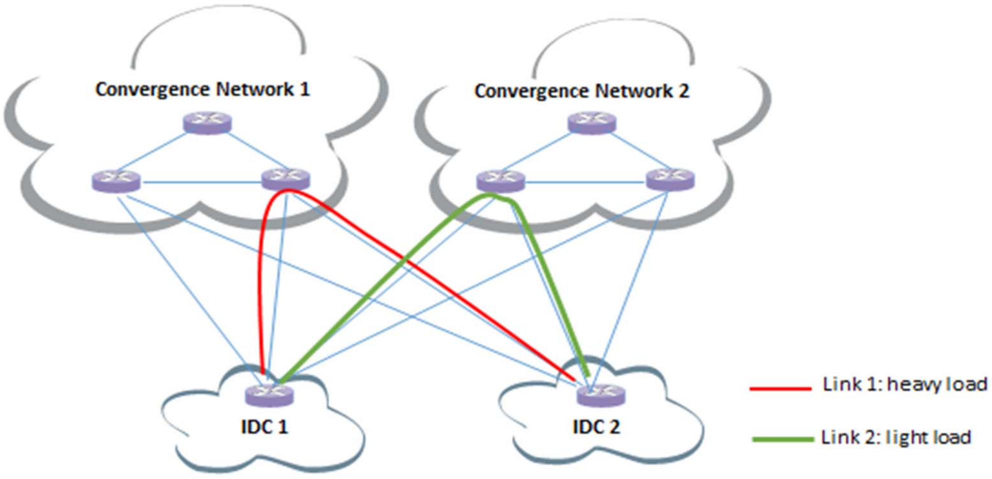


Figure 5-1: Policy-driven, automatic IDC traffic steering

As shown in left portion of Figure 5-1, two IDCs can connect to each other via two different paths. There are multiple links between the two IDCs. When link 1 is heavily loaded, as much traffic as necessary can be moved to link 2.

5.2.1.2.3 Actors and Roles

- IDC network.
- ENI System.
- Network manager (City Level).

Stakeholders managing the above:

- Operators.

5.2.1.2.4 Initial context configuration

- The network administrator's inputs the policies.
- IDCs connect to each other via different links.
- Network traffic routed via different links is defined according to policies.
- Bandwidth of tenant may need to be adjusted in real time according to the dynamic needs of the tenant and the operational context.

5.2.1.2.5 Trigger conditions

- Utilization of a link or bandwidth of a tenant exceeds the configured threshold (e.g. as defined in an SLA).
- Change in operational requirements.

5.2.1.2.6 Operational Flow of the actions

For intelligent link load balancing:

1) network administrator pre-configures the threshold/constraint of link utilization and appropriate metadata and metrics to monitor link loads;

2) ENI system uses the network administrator's input to modify policies considering, for example, available links, bandwidth, link utilization, and constraints;

3) network administrator executes policies and ensures they all execute correctly (i.e. without error);

4) the ENI system adapts to monitor metrics, metadata, and other information (as defined by the above generated policies) to achieve measured improvements;

5) IDC network uses the policies to manage the network behaviour.

For intelligent bandwidth allocations:

- network administrator pre-configures the threshold of bandwidth utilization and appropriate metadata and metrics to monitor bandwidth;
- ENI system collects the bandwidth usage data for each tenant for a specified time period;
- ENI system pre-processes the data to extract the appropriate characteristics of the tenant's service to determine if the allocated bandwidth is sufficient or not;
- ENI system establishes an appropriate mathematical model to predict the bandwidth requirements of tenants at different times in the coming year;
- ENI system collects and analyses real-time bandwidth usage data for tenants;
- when the configured bandwidth utilization threshold is danger of being reached, the ENI system proactively adjusts the bandwidth allocation policies for the affected tenants, taking into account the QoS policy and other SLA policies of each tenant.

5.2.1.2.7 Post-conditions

The impact of dynamic polices:

- Network traffic is balanced.
- Appropriate metrics and metadata are continually gathered to ensure that the service requirements are met or exceeded.
- Bandwidth for the tenant is assured and automatically adjusted in real-time.
- Bandwidth utilization is improved.

5.2.2 Use Case #1-2: Handling of Peak Planned Occurrences

5.2.2.1 Use case context

Currently, most services share a common infrastructure where resource allocation is a very critical process. When a network operator extends its infrastructure to a new area or upgrades an already existent, it makes an assessment on the number of customers and services the infrastructure will serve under normal operation scenarios. Then, when provisioning services, the configuration of the infrastructure is performed once and usually does not change during the service lifecycle. Although advanced QoS strategies help to mitigate peak resources usage scenarios, it still constitutes a very static and slow process for today's services, which imply the need for the process to become more and more dynamic. Moreover, when considered as adequate and feasible, a network operator may make use of mobile stations for such temporary increases on network capacity.

Typical peak scenarios may be characterized by the occurrence of localized and temporary bursts of network traffic caused by planned events, e.g. soccer games, or unplanned, e.g. natural catastrophes, which may lead to critical service level degradation or even service disruption along with the subsequent impact in operators, in services as well as in end user's experience. In particular, for network operators, service degradation and/or disruption constitutes something that is to be avoided no matter at what cost as it jeopardizes network operator's image as a service provider among its customers.

In the present Use Case, only planned events will be taken into account.

5.2.2.2 Description of the use case

5.2.2.2.1 Overview

Service prioritization and management of resource sharing infrastructures are very complex processes for operators, which take a considerable amount of time for planning, and are normally performed only once in a given area. When dealing with temporary planned events, it is necessary to calculate the stress on the network infrastructure and define backup action plans to mitigate potential service degradation or even disruptions. Additionally, after the end of the event it is necessary to revert the temporary changes to the normal usage conditions.

An example of such a temporary planned event could be the case of a certain area, served by a network infrastructure for telecommunication services, which will be hosting a music event that will be broadcasted by live television. Currently, the network infrastructure is providing resources to several service instances in a shared manner. A relatively large crowd is expected at the event and if no actions are taken by the network operator there is the possibility of degradation on some of the services that make use of that region's infrastructure.

Analysis performed during the planning of events may also encompass the ability to extend the current infrastructure capacity of that area.

The current Use Case is further described by the following set of components and features.

5.2.2.2.2 Motivation

With the ENI system, the use of AI methods on helping to understand the context dynamicity and on predicting potential peak traffic scenarios becomes quite important. More specifically, the AI can perform the calculation of possible scenarios for planned events by making use of machine learning, e.g. by taking into account events history. On the other hand, it can also assist on the calculation based on the expected response of network equipment under stress, which can also help on the preparation and definition of the necessary backup action plans. In addition, the ENI system can also evaluate, for all these scenarios, if the use of resource sharing techniques is enough to support the increase of network traffic or if there is the need for additional measures, e.g. mobile stations that provide additional physical resources.

Still another benefit related to the AI capability to provide more realistic predictions lies in the possibility for network operators to use narrower margins of the total amount of resources when they wish to extend the current resource capacity of a given region. With these new tools, network operators may enforce pre-defined policies to govern the responses of the ENI System, e.g. do not use mobile stations if the peak consumption is not expected to exceed 90 % of the current network capacity.

5.2.2.2.3 Actors and Roles

The presence of the following actors/entities as well as their associated roles are envisaged in the current Use Case:

- Customers/clients: end users that enjoy the delivery of a service.
- Network Administrator: entity/person responsible for the initial policy design that encompasses the planning of the Network Infrastructure regarding the mitigation impact of planned events, which may involve the extension of the infrastructure capacity. With the assistance of the ENI System regarding these planning activities, this entity/person is in a position to choose the most suitable backup action/plan to be enforced.
- Network Infrastructure: network elements and resources that participate in service fulfilment procedures.
- Network Operator: owner of the Network Infrastructure that is used to provide services to customers/clients.
- Operations support system/Business support system (OSS/BSS): operational and business systems that belong to the management system of network operators. In this case they are providing, among others, monitoring, actuation, internal records of very different items that may range from products to resources, as well as other business interfaces dedicated to external entities.
- ENI System: component that governs service fulfilment and participates in planning and configuration procedures upon occurrence of planned scenarios that may impact service delivery, which may encompass situations involving extension of infrastructure capacity.

5.2.2.2.4 Initial context configuration

The network is operating in perfect conditions with all its components working in good shape.

5.2.2.2.5 Triggering conditions

A music event is scheduled for a certain area and may lead to local service degradation or disruption. On occurrence of the planned event, backup actions, previously calculated by the ENI System and validated by the Network Administrator, are triggered. Those backup actions may encompass extensions on infrastructure capacity.

5.2.2.2.6 Operational flow of actions

The following sequence of actions may be identified:

1) After receiving a notification of a new event e.g. a music festival, the ENI System makes use of AI methods to calculate and produce a report containing several scenarios and their respective outcome depending on the size of the crowd and expected local resource consumption. For each scenario, it also produces a backup action plan, i.e. possible changes to local QoS profiles or additional resources needed, taking into account previously defined policies.

2) In its notification report, the ENI System signals one of the scenarios as the most suitable and asks the Network Administrator for validation.

3) The Network Administrator evaluates the proposed scenarios and backup plans, validates one of them, and notifies the ENI System about its choice.

4) Upon receiving the Network Administrator's reply, the ENI System elaborates a schedule containing a roadmap of the backup actions to be subsequently performed.

5) On occurrence of the planned event, the ENI System triggers the proper configuration operations via OSS components on impacted network infrastructure resources, including possible redundant resources that may be reserved for any deviation on the predicted consumption.

6) During the event, the ENI System increases the monitoring resolution on the previously mentioned resources.

7) If found as necessary, it may activate additional resources, if available, previously reserved for any deviation on the predicted consumption.

8) At the end of the event the ENI system triggers the rollback of the network resources configuration to the state where it was immediately before the event.

5.2.2.2.7 Post-conditions

The local network infrastructure is operating according to the planned deployment prior to the event. All information regarding network infrastructure during the event is stored to increase the prediction capabilities of the ENI System.

5.2.3 Use Case #1-3: Energy optimization using AI

5.2.3.1 Use case context

By introducing Network Function Virtualisation (NFV) different virtual networks can be deployed on the same NFV Infrastructure (NFVI) for different network services. The Virtual Network Function (VNF) instances are implemented on Virtual Machines (VMs) or Containers. And the VNF instances can be instantiated, scaled in/out, or terminated on demand by using Management and Orchestration (MANO) system or any other form of orchestrator. The VNF instances can be easily moved from one server to another server by using VM/Container migration technologies. Therefore, the services provided by the VNFs can be steered from one server to another server along with the VM/Container migration, while the network services provided by the VNFs are uninterrupted.

With the trend of NFV, more and more DCs will be deployed to replace the traditional Central Offices in the operators' network. The data centres (DC) are made up of many servers with huge power consumption. Typically, the servers in a DC take 70 % of the total power consumption. The other equipment including switches, routers, storage equipment and air conditioners take the other 30 % of the total power consumption. The servers are deployed and running to meet the requirement of peak hour service, which means the servers are normally at high power-up state at full time even in non-peak hours. It is however possible to move the services to some of the servers and turn the other servers to idle or underclocking state in non-peak hours, with the aim of optimizing the power usage at the DC. It should be noted that such mechanism of energy optimization can be applied widely to other network resources in addition to data centres. In the following, reducing waste energy in individual DCs is used as an example for this use case to elaborate how NFV and AI can be combined to optimize usage of the energy in networks.

5.2.3.2 Description of the use case

5.2.3.2.1 Overview

Traditional ways of DC energy saving are normally done manually and the effect is not obvious. Power consumption of the DCs, same as the other network physical resources, represents a large portion of the cost for operators, and causes environmental concerns. Consisting primarily of a homologous architecture/resource pool, the scope of what can be optimized in an intra-DC context is limited. It is however, a necessary first step towards greater AI-driven improvements that are realizable with the additional consideration of both inter-DC orchestration and/or the exploitation of heterogeneous network resource pools (such as edge or IoT devices). The consideration of these additional factors will enable the minimization of the carbon foot print through intelligent resource management. For example, by relying solely on edge device compute resources in periods of low demand, an ENI system could identify and act on these requirements in an autonomous fashion ensuring that OPEX is optimized, among other Key Performance Indicators (KPIs).

5.2.3.2.2 Motivation

By using ENI system, the usage pattern of the services can be learned from historical data and updated in real-time way. The ENI system can help to trigger the movement of the services and turn the spare servers to idle state. As shown in Figure 5-2, if the actual load of service in one day is represented by a curve, then the shadow between the peak and the curve is potential energy saving for the DC. The optimization may take information from multiple sources and predict and analyse in an autonomous way.

In addition, the ENI system can predict the peak hours by using artificial intelligence techniques such as deep learning or machine learning, and then wake up necessary number of servers into full load state. If an unexpected event is detected, more servers can be woken up to support this burst. By using ENI system and AI techniques, the energy saving for DCs can be achieved and OPEX can be saved.

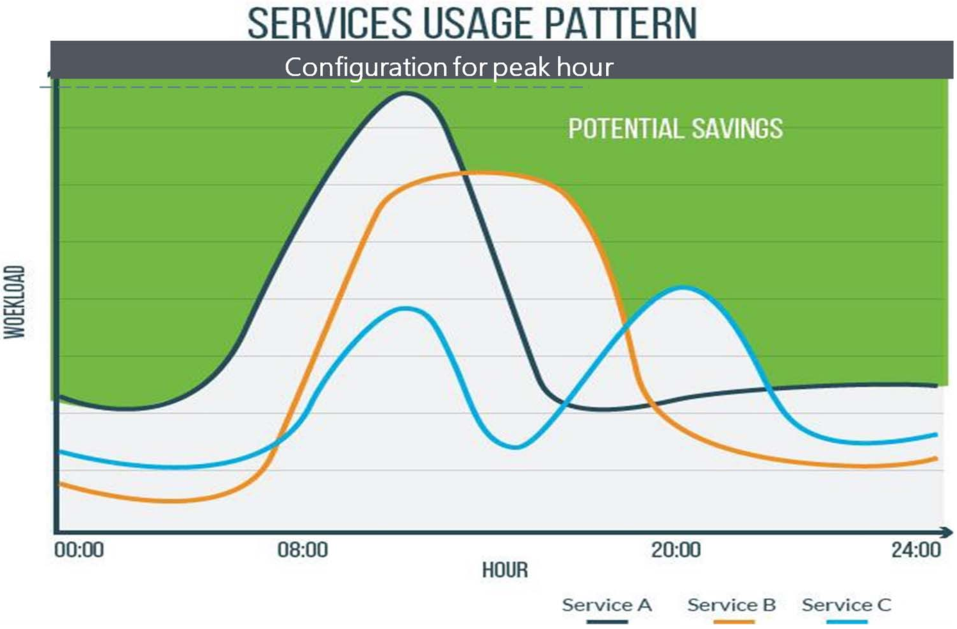


Figure 5-2: Potential DC energy saving by AI

5.2.3.2.3 Actors and Roles

- Operator: manages the DCs and confirms the VM/Container migration policies and scale in/out policies.
- ENI System: collects and learns service pattern from the data collected from the DC servers; determines the VM migration policies and scale in/out policies according to prediction of the service requirements; triggers steering of the service flows from one VM to another VM.
- DC servers: provide the required information to the ENI system, execute the VM migration and VNF scale in/out according to the policies.
- DC environmental monitoring and control system: provide the required information to the ENI system, and execute the operation of environmental adjustment.
- NFV MANO: executes the lifecycle management operation of the VNFs according to policies.

5.2.3.2.4 Initial context configuration

All servers in the DC are running all time and the energy consumption is high. The ENI system performs some initial actions related to the collection of information, use of AI algorithms and service patterns learning.

5.2.3.2.5 Triggering conditions

The following trigger types associated with the ENI system may be identified:

- The ENI system predicts that the required resources of a service will fall below a certain threshold in a certain period.
- The ENI system predicts that the required resources of a service will grow up higher than a certain threshold in a certain period.
- The ENI system decides to change the DC environmental settings.
- The ENI system detects a change of the service pattern learned before.

5.2.3.2.6 Operational flow of actions

The following initial sequence of actions may be identified:

1) The ENI system collects and stores information of the virtual networks, including CPU usage, storage usage, and network usage for each VNF, etc. as well as the power consumption information and environmental information.

2) The ENI system uses AI algorithm to build the relations between the network service and its required resources, and the relations between the power consumption and the environment settings including e.g. the location of the running servers, the setting of the cooling system, etc.

3) The ENI system learns the service pattern and predicts the required resources of the service in a certain period in the future, e.g. the next hour.

The following triggers and subsequent actions may be identified:

1) When the ENI system predicts that the required resources of a service will fall below a certain threshold in a certain period, and the service configured by the operator as able to be moved, the ENI system triggers, directly or indirectly, the NFV MANO system to migrate the services and VMs/Containers providing this service to another selected server:

    a) If the VMs/Containers on one server are all migrated to another server, the spare server is turned into idle mode.

2) When the ENI system predicts that the required resources of a service will grow up higher than a certain threshold in a certain period, the ENI system triggers the scale out of the existing VNF and bring up new VMs/Containers:

   a) If the running servers cannot provide the required resources of a new VM/Container according to prediction, the ENI system wakes up a selected idle mode server.

3) The ENI system may decide to change the DC environmental monitoring and control system to adjust the environmental settings when a server is woke up or turned into the idle mode.

4) When the ENI system detects a change of the service pattern learned before, the ENI system will adjust the VM/Container migration policies and scale in/out policies.

5.2.3.2.7 Post-conditions

Servers in the DC are dynamically turned to idle and waken up according to the service pattern; therefore the cost of power consumption is reduced as much as possible.

5.2.3.3 Mapping to ENI reference architecture

5.2.3.3.1 Functional blocks

The mapping to ENI architecture for energy optimization using AI is shown in Figure 5-3.

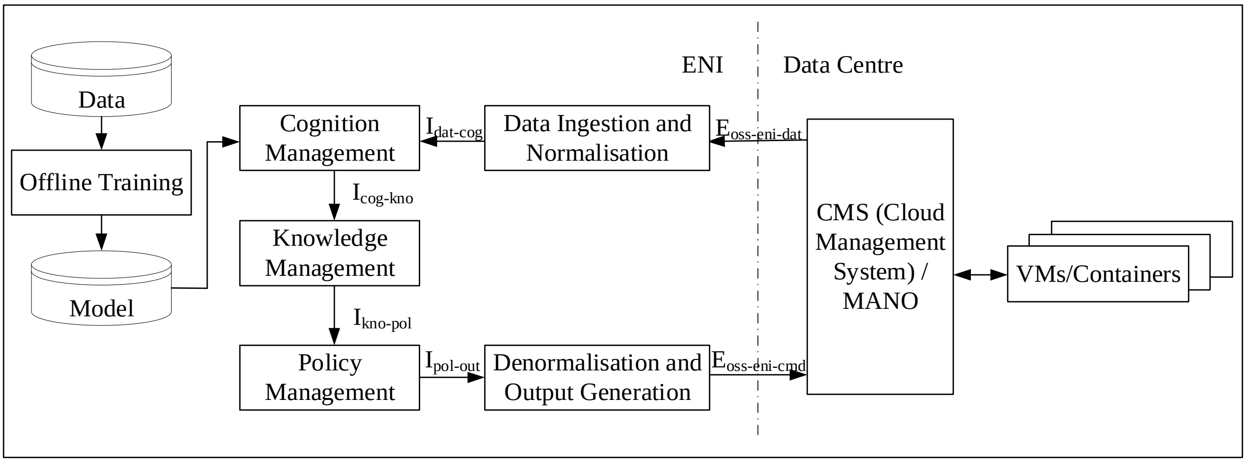


Figure 5-3: Mapping to ENI reference architecture

Data Ingestion and Normalization Functional Block converts the data collected from the DC servers into normalized form so that the ENI System can analyse and understand.

Cognition Management Functional Block evaluates the existing knowledge and performs inferences using the trained model and ingested data to make predictions about the future service requests, then determines if any operation should be taken to achieve the goal of energy saving.

Knowledge Management Functional Block stores the generated knowledge and defines a formal and consensual representation of knowledge so that the computer system could implement the machine learning algorithms and perform reasoning using the knowledge representation.

Policy Management Functional Block provides decisions about server consolidation and live migration to optimize energy utilization and guarantee the key performance indicators (KPIs).

Denormalization and Output Generation Functional Block converts information (recommendations and decisions) generated by the ENI system to a form that the CMS (Cloud Management System)/MANO can understand and execute.

5.2.3.3.2 Interfaces

$E_{\text{oss-eni-dat}}$ defines data exchange between the ENI System and the Data Centre (Assisted System). The Cloud Management System in the data centre collects data from virtual machines and send it to ENI system.

$E_{\text{oss-eni-cmd}}$ defines recommendations and/or commands and acknowledgements exchanged between the ENI System and the Data Centre (Assisted System). The ENI System provides decisions about server consolidation and live migration so that the Cloud Management System could take actions accordingly to save the energy and OPEX for DCs.

$I_{\text{dat-cog}}$ defines the internal interface between Data Ingestion and Normalization Functional Block and Cognition Management Functional Block, which passes the normalized data and information to the Cognition Management Functional Block to perform inferences and generate new knowledge.

$I_{\text{cog-kno}}$ defines the internal interface between Cognition Management Functional Block and Knowledge Management Functional Block, which passes the generated new knowledge (e.g. predictions of future service requirements) to the Knowledge Management Functional Block for storage.

$I_{\text{pol-dat}}$ defines the internal interface between Policy Management Functional Block and Denormalization and Output Generation Functional Block, which passes the recommendations/commands so that it can be converted to a form that CMS/MANO can execute and implement.

5.2.3.3.3 Flow of information

The flow of information is given in Figure 5-4.

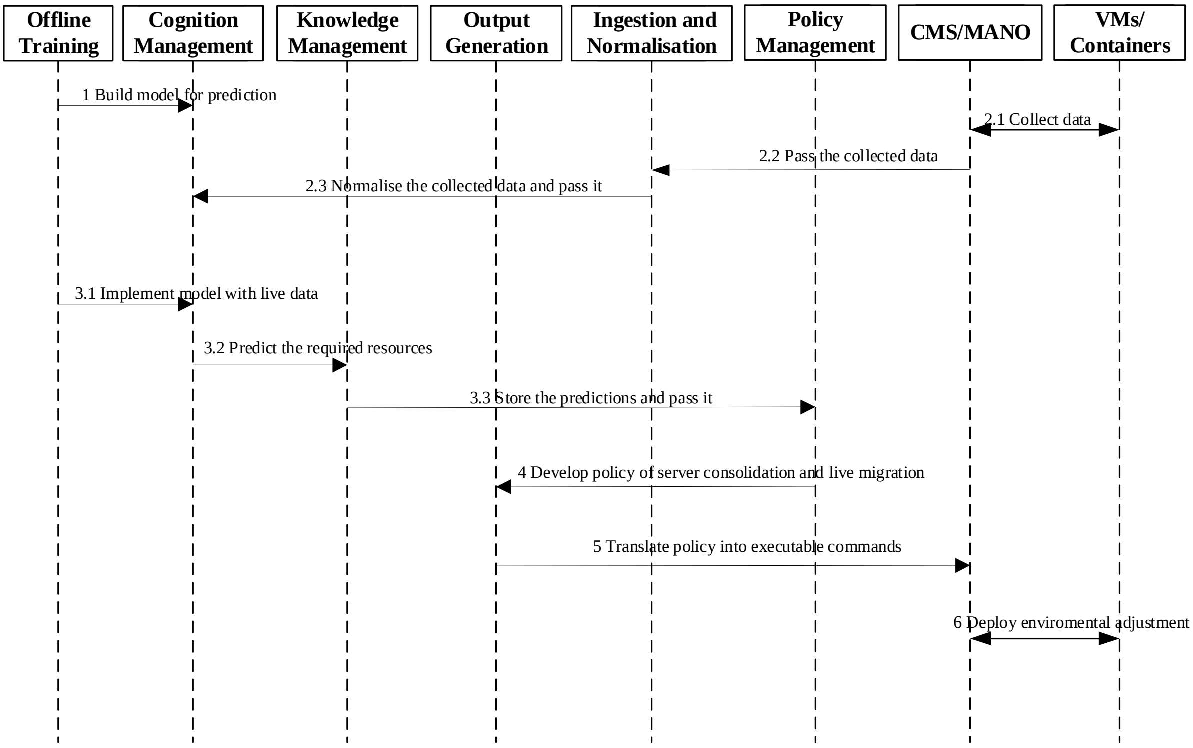

Offline Training
Cognition Management
Knowledge Management
Output Generation
Ingestion and Normalisation
Policy Management
CMS/MANO
VMs/ Containers
1 Build model for prediction
2.1 Collect data
2.2 Pass the collected data
2.3 Normalise the collected data and pass it
3.1 Implement model with live data
3.2 Predict the required resources
3.3 Store the predictions and pass it
4 Develop policy of server consolidation and live migration
5 Translate policy into executable commands
6 Deploy enviromental adjustment

Figure 5-4: Procedure for energy optimization using AI

Step 1: Choose the appropriate model and train the model with historical data

Step 2.1: The Cloud Management System collect data (e.g. CPU usage rate, storage usage rate, network throughput) from virtual machines/containers

Step 2.2: The collected data is sent back to the ENI system

Step 2.3: The Ingestion and Normalization Functional Block converts data from multiple input sources into a normalized form and feed the information to the pre-trained model

Step 3.1: Fine-tune and implement the model with live data

Step 3.2: The Cognition Management Functional Block analyses the existing knowledge and generate new knowledge (e.g. predictions about the required resources of the service)

Step 3.3: The Knowledge Management Functional Block stores the prediction and send it to the Policy Management Functional Block

Step 4: Develop policies about server consolidation and live migration according to the predictions of the future service flow

Step 5: Convert policy into a form that Cloud Management System could understand and take actions

Step 6: The Cloud Management System implements the environmental adjustment on the virtual machines/containers

5.3 Network Operations

5.3.1 Use Case #2-1: Policy-driven IP managed networks

5.3.1.1 Use case context

There are some types of network nodes that need to allocate IP addresses to end users. Examples include Broadband Remote Access Server (BRAS), Dynamic Host Configuration Protocol (DHCP) server, and Carrier Grade Network Address Translation (CGN). Each of these network nodes needs to be configured with IP addresses (i.e. from an IP address pool), which they can use to allocate to the end users. Currently, the plan and configuration of IP address pools rely on manual configuration that is fundamentally static in nature.

5.3.1.2 Description of the use case

5.3.1.2.1 Overview

In a common scenario of Home Access, the client sends an access request to a BRAS. The BRAS picks one IP address from its pre-configured IP address pool and allocates that IP address to this client; this enables this client to access the network using this IP address. CGN translates private address into public address. CGNs are configured with several public IP address pools. When there is a need for a CGN to translate the private IP address of one session from the client side to a public IP address for network side, the CGN picks one public IP address in its pre-configured public IP address pools, replaces the private IP address by the selected public IP address, and records this mapping.

5.3.1.2.2 Motivation

The traditional IP management approach suffers from low utilization of IP addresses and poor sharing among equipment. Manual address allocation is cumbersome, and scripts are fragile and cannot adjust to dynamic network conditions. There are several disadvantages:

- Currently certain operators do not have sufficient IP address resources, especially for IPv4.
- IP address resource utilization ratio is low in general: some network nodes have low utilization ratio of internal addresses; some devices suffer from tidal effect (i.e. high in peak period and low in idle period).
- Address resources are not shared among equipment, which leads to inefficiencies in deployment.

In this use case, the ENI System learns the pattern of user sessions, which consumes the IP addresses, and classifies the users accordingly. The ENI System generates IP address pool configuration policies and IP address allocation policies to improve the efficiency of the utilization of IP addresses.

Policy enables more intelligent usage of address pools and automates the address allocation. With policy-driven network resource optimization and network resource monitoring, it is possible to automatically adjust address allocation on different equipment using policies. Such policies may consider factors such as demand on address, utilization ratio, address usage lifecycle, and constraints (e.g. the rejection rate of a BRAS or CGN, or thresholds that apply to address utilization). This allows more intelligent usage of address pools and automates the address allocation process, where improved operator experience can be expected. It also ensures more consistent operation of address allocation, which also improves the operator experience.

Such a use case is illustrated in Figure 5-5.

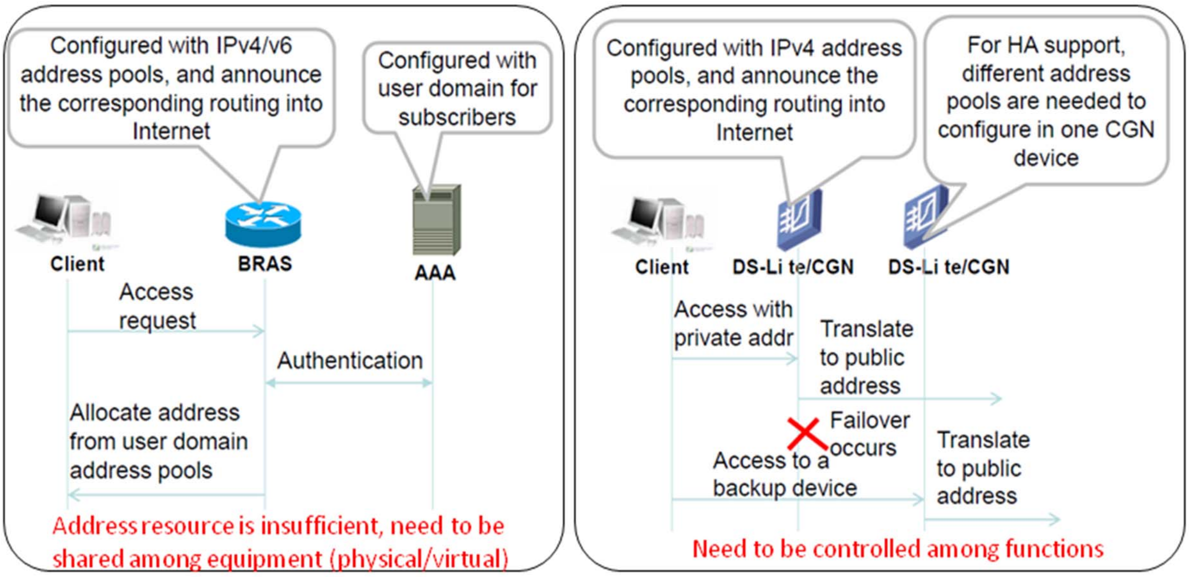

Configured with IPv4/v6 address pools, and announce the corresponding routing into Internet
Configured with user domain for subscribers
Client
BRAS
AAA
Access request
Authentication
Allocate address from user domain address pools
Address resource is insufficient, need to be shared among equipment (physical/virtual)
Configured with IPv4 address pools, and announce the corresponding routing into Internet
For HA support, different address pools are needed to configure in one CGN device
Client
DS-Li te/CGN
DS-Li te/CGN
Access with private addr
Translate to public address
Failover occurs
Access to a backup device
Translate to public address
Need to be controlled among functions

Figure 5-5: Current Problems in Operator IP Managed Networks

5.3.1.2.3 Actors and Roles

- ENI System with the IP address allocation algorithm system and data collection system.
- Network Functions which need IP address pool configuration (e.g. vBRAS).
- Network administrator.

Stakeholders managing the above:

- Operators.

5.3.1.2.4 Initial context configuration

- The network administrator's inputs the policies to configure the number and size of IP address blocks.
- One vBRAS is configured with a predefined number of IP address blocks, where each block contains a predefined number of IP addresses.

vBRAS allocates an IP address to the users randomly; current solutions suffer from many IP addresses in each IP address block not being used, with at least one IP address in use.

5.3.1.2.5 Triggering conditions

- Trigger 1 for IP address allocation policy adjustment: when a user's IP address usage does not align with the current allocation policy more than a predefined number of times in one measurement time period, the ENI system will adjust the IP address allocation policy according to the latest information from the user.

- Trigger 2 for IP address allocation policy adjustment: when one or more users change their behaviour, the ENI system will those users based on an appropriate classification or clustering algorithm, and adjust the IP address allocation policy accordingly.

5.3.1.2.6 Operational flow of actions

1) The ENI system collects and stores the information of the users' usage of IP addresses, in a normalized format with user ID, location number, daily IP address usage time, holiday IP address usage, weekdays and weekends IP address usage, etc.

2) The ENI system uses one or more classification or clustering algorithms to build an appropriate model. Users are labelled based on their behaviour characteristics by using an appropriate algorithm, according to their historical and contextual information (e.g. location information, time of attachment and detachment, types of applications used, and amount of data transferred).

3) The ENI system modifies policies to re-configure the number and size of IP address blocks to be allocated to each user group, as well as the IP address allocation mechanisms.

4) IP address blocks and IP address allocation policies are sent to the BRAS for processing:

   a) When a user attaches to the BRAS, the BRAS allocates an IP address to the user in his/her corresponding IP address block, according to the IP address allocation policy and the user information including his/her equipment identifier.

   b) When the current usage of one IP address block reaches a threshold, the BRAS will select another IP address block with the same characteristics for further IP address allocation to the same type of users.

   c) If all IP addresses in an IP address block are not in use, and the IP addresses are not kept for redundancy purposes, this IP address block will be recycled.

5) When triggered, the ENI system will regroup the users and adjust the IP address allocation policy accordingly.

6) When a user attached to the BRAS requesting for an IP address, the BRAS will select a most frequently used IP address block among the ones mapping to the user label, and allocate an IP address in this block to the user.

5.3.1.2.7 Post-conditions

All current users have the minimum number of IP addresses allocated or reserved. IP address pools are optimized.

5.3.2 Use Case #2-2: Radio Coverage and capacity optimization

5.3.2.1 Use case context

Coverage and capacity optimization (CCO) is one of the typical operational tasks of the radio access network (RAN). CCO aims to provide the required capacity in the targeted coverage areas, to minimize the interference and maintain an acceptable quality of service in an autonomous way. To achieve these targets, antenna power and configuration (pilot power, antenna down tilt, antenna azimuth, or massive MIMO pattern in 5G) play a critical role, as they affect the direction of the antenna radiation pattern, therefore can be used to improve the received signal strength in the own cell as well as to reduce the interference to neighbouring cells.

The CCO task also exists in enterprise wireless local area network (WLAN) scenario. In enterprise WLAN, an access point (AP) controller sets multiple APs' RF parameters (e.g. channel frequency, bandwidth, power) to provide full coverage and minimize the inter-cell interference (namely dynamic channel allocation and transmit power control).

5.3.2.2 Description of the use case

5.3.2.2.1 Overview

CCO allows the system to periodically adapt to the changes in traffic (i.e. load and location) and the radio environment by adjusting the key radio frequency (RF) parameters (e.g. antenna configuration and power). For the online CCO task, it is not possible to find definite function to map between the RF parameters and the target coverage and capacity performance. The main reason is that the set of configurable RF parameters is multi-dimensional, and each RF parameter has wide range of values, leading to very large number of possible options.

5.3.2.2.2 Motivation

Performing exhaustive search to find optimal RF parameter combination and associated value can be extremely complex. Today's network lacks efficient way of find the optimal combination of RF parameters for the changing network environment. An intelligent entity (e.g. ENI system) can leverage machine learning to analyse and learn what the proper action is for each current network state (e.g. current RF parameters, user equipment (UE) location, traffic load, Spectrum allocation, etc.). Based on the learnt model (which can be continuously optimized), the ENI system can then instruct the operations system (OS) the base station the proper action to adjust the RF parameters for optimizing coverage and capacity.

In WLAN scenario, the ever-changing radio environment (e.g. external AP interference and non-Wi-Fi-type interference) requires the system to adjust their RF parameters to achieve best performance. Using collected RF parameters, signal strength and throughput data, an intelligent entity (e.g. ENI system) can use machine learning to learn the mapping relationship, and instruct the AP controller to set proper RF parameters for those managed APs to optimize coverage and capacity.

The use case is illustrated in Figure 5-6.

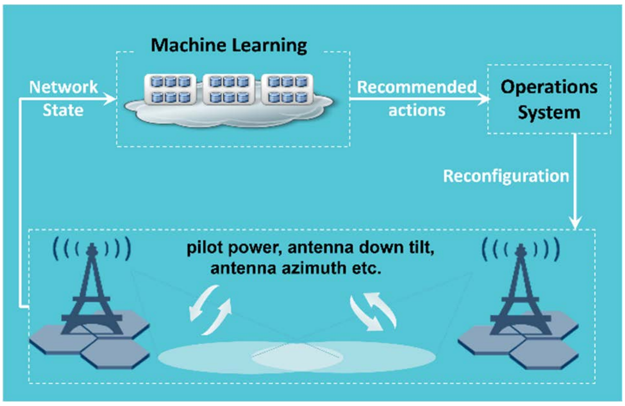


Figure 5-6: Coverage and Capacity Optimization

5.3.2.2.3 Actors and Roles

- Operator: defines the target coverage and capacity performance (e.g. maximize the traffic and Transmission Control Protocol (TCP) load) of managed areas.
- ENI Engine: collects and analyses the state and performance of radio access network, dynamically determines what RF parameters should be configured according to them.
- Operations System: adjusts RF parameters according to the policies generated by ENI system.

5.3.2.2.4 Initial context configuration

- The configurations of RF parameters are fixed.
- The ENI system is learning how to configure the RF parameters in order to achieve the target coverage and capacity in certain network state through its machine learning capacities.

5.3.2.2.5 Triggering conditions

Current RF parameters configurations do not meet the target coverage and capacity performance.

5.3.2.2.6 Operational flow of actions

1) Operator pre-configures the target coverage and capacity performance.

2) ENI system collects and analyses the radio environment information to be aware of the state and performance of current network.

3) ENI system determines the RF parameters configuration according to the current network state and target coverage and capacity performance.

4) Operations system reconfigures the RF parameters according to the output of ENI system.

5.3.2.2.7 Post-conditions

- The RF parameters dynamically adjust according to the changing radio environment.

- The target coverage and capacity performance is met.

5.3.2.3 Mapping to ENI reference architecture

5.3.2.3.1 Functional blocks

The mapping to ENI architecture for radio coverage and optimization using AI is shown in Figure 5-7.

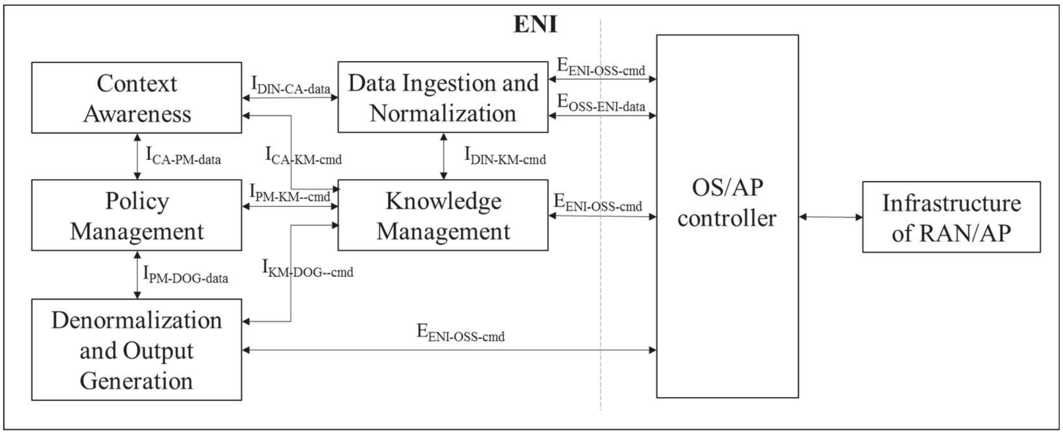


Figure 5-7: Mapping to ENI reference architecture

The knowledge management functional block holds the goals of the operator, the OS expert knowledge such as the rules of RF reconfiguration, the source data which should be collected for the CCO, the method of data process, the details of models that used for sensing the real-time state of RAN, the rules of model optimization and updating, and the policy learned by the ENI system.

The data ingestion and normalization function block transfers the raw data provided by the OS/AP controller into a form that can be understood by the ENI system.

The context awareness functional block analyses the real-time coverage and capacity performance of RAN by using AI models, meanwhile, it retrains and updates the model periodically.

The policy management functional block determines the actions should be taken, such as increase the downtilt angle by one degree. Based on the performance after the action taken by OS/AP controller, the block determines the next action should be taken. If the performance got worse, the block determines to roll back the configuration, and if the state got better, the block determines to future adjust the configuration. Finally, the coverage and capacity performance met the goals, and the block learned the optimal action should be taken under certain state.

The denormalization and output generation functional block converts recommendations generated by the ENI system, to a form that the OS/AP controller can understand.

5.3.2.3.2 Interfaces

$E_{OSS\text{-}ENI\text{-}data}$ defines data exchanged between ENI and the OS/AP controller. The OS/AP controller collects data from RAN infrastructure or AP and send it to ENI system. The data includes history traffic statistics, measurement report and configuration data, etc.

$E_{OSS\text{-}ENI\text{-}cmd}$ defines recommendations and/or commands and acknowledgements exchanged between ENI and the OS/AP controller. The commands include the downtilt angle parameter should be configured provided by ENI system, the request for RAN/AP infrastructure data sent from ENI system and the expert knowledge provided by OS/AP controller.

$I_{DIN\text{-}KM\text{-}cmd}$ defined recommendations and/or commands and acknowledgements exchanged between knowledge management functional block and data ingestion and normalization functional block. The recommendations include the data need to collect and the method of data ingestion and normalization provided by knowledge management functional block, which includes feature extraction and feature engineering, etc.

$I_{CA\text{-}KM\text{-}cmd}$ defined recommendations and/or commands and acknowledgements exchanged between knowledge management functional block and context awareness functional block. The commands include the model information used for context awareness, and the rules for model retraining and updating provides by knowledge management functional block, and the updated model information provided by context awareness functional block.

$I_{PM\text{-}KM\text{-}cmd}$ defined recommendations and/or commands and acknowledgements exchanged between knowledge management functional block and policy management functional block. The commands includes the RF configure rules provided by OS/AP controller and the new rules learned by ENI system.

$I_{KM\text{-}DOG\text{-}cmd}$ defined recommendations and/or commands and acknowledgements exchanged between knowledge management functional block and denormalization and output generation functional block. The commands include methods of transfer the command generated by ENI to the form that OS/AP controller can understand.

$I_{DIN\text{-}CA\text{-}data}$ defines data exchanged between data ingestion and normalization functional block and context awareness functional block.

$I_{CA\text{-}PM\text{-}data}$ defines data exchanged between context awareness functional block and policy management functional block, which includes the real-time coverage and capacity performance.

$I_{PM\text{-}DOG\text{-}data}$ defines data exchanged between policy management functional block and denormalization and output generation functional block, which include the recommend action generated by policy management.

5.3.2.3.3 Flow of information

The flow of information for this use case is given in Figure 5-8.

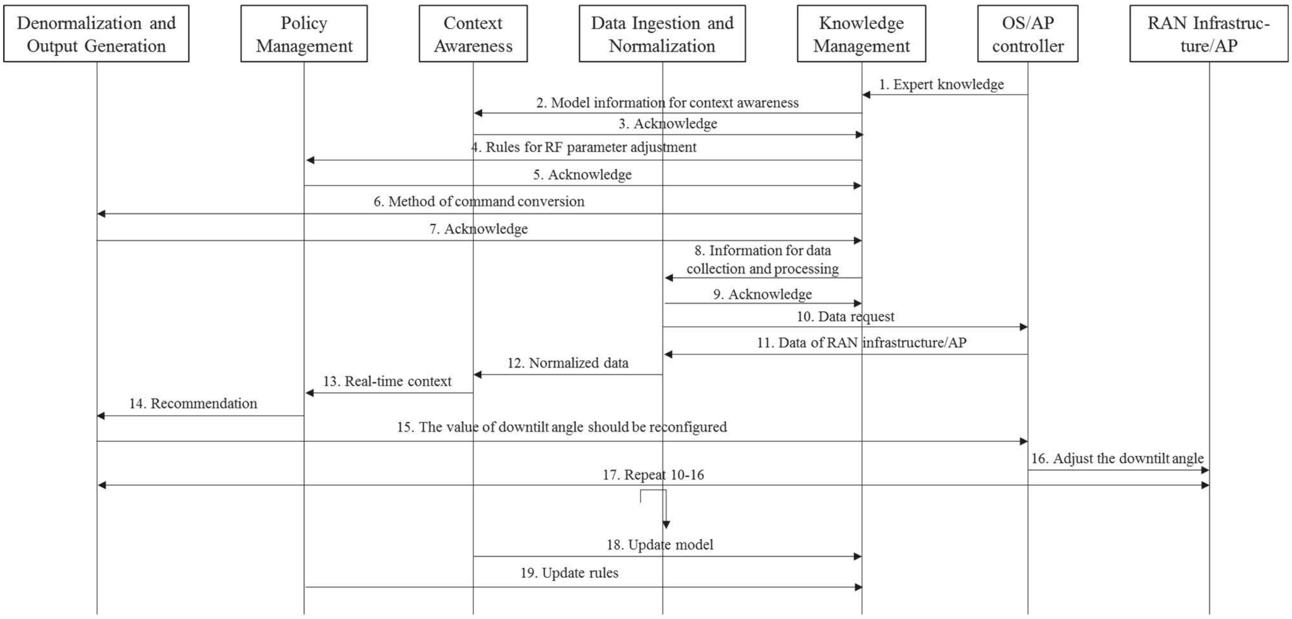

Denormalization and Output Generation
Policy Management
Context Awareness
Data Ingestion and Normalization
Knowledge Management
OS/AP controller
RAN Infrastructure/AP
1. Expert knowledge
2. Model information for context awareness
3. Acknowledge
4. Rules for RF parameter adjustment
5. Acknowledge
6. Method of command conversion
7. Acknowledge
8. Information for data collection and processing
9. Acknowledge
10. Data request
11. Data of RAN infrastructure/AP
12. Normalized data
13. Real-time context
14. Recommendation
15. The value of downtilt angle should be reconfigured
16. Adjust the downtilt angle
17. Repeat 10-16
18. Update model
19. Update rules

Figure 5-8: Procedure for coverage and capacity optimization

1. OS/AP controller transfers expert knowledge to knowledge management functional block.

2. Knowledge management functional block transfers model information for context awareness to context awareness functional block.

3. Context awareness functional block transfers acknowledge to knowledge management functional block.

4. Knowledge management functional block transfers rules for RF parameter adjustment to policy management functional block.

5. Policy management functional block transfers acknowledge to knowledge management functional block.

6. Knowledge management functional block transfers method of command conversion to denormalization and output generation functional block.

7. Denormalization and output generation functional block transfers acknowledge to knowledge management functional block.

8. Knowledge management functional block transfers information for data collection and processing to data ingestion and normalization functional block including what data needs to be collected and the method for data ingestion and normalization.

9. Data ingestion and normalization functional block transfers acknowledge to knowledge management functional block.

10. Data ingestion and normalization functional block transfers request for RAN infrastructure/AP data to OS/AP controller.

11. OS/AP controller transfers the collected data based on the requirement of step10 to data ingestion and normalization functional block, such as history traffic statistics, measurement report and configuration data.

12. Data ingestion and normalization functional block processes the raw data and then transfers the processed data to context-awareness functional block.

13. Context-awareness functional block gets the real-time coverage and capacity performance to policy management functional block.

14. Policy management functional block determines how to adjust the downtilt angle according to the state of network, and then transfers the recommendation to denormalization and output generation functional block.

15. Denormalization and output generation functional block gets the command that OS/AP controller can understand, and then transfer it to OS/AP controller.

16. OS/AP controller adjusts the configuration of RAN infrastructure/AP.

17. ENI system continue monitors the mobile network performance and adjusts the RF parameters.

18. Context-awareness functional block retraining and updating model, and then replace the model saved in knowledge management functional block.

19. Policy management functional block learned the optimal action should be taken under certain state, and then transfers the knowledge to knowledge management functional block.

5.3.3 Use Case #2-3: Intelligent Software Rollouts

5.3.3.1 Use Case context

Physical resources such as routers, during their lifetime, need to have their firmware updated, not only for the support of new services or functionalities, but also to fix existent impairments. In some cases a firmware rollout can take several months to plan and enforce.

Indeed, updating a physical resource firmware constitutes a particularly delicate use case since it involves service disruption, potential bugs on the new version or in the worst case scenario the need to use workforce for equipment replacement. Thus, operators are very cautious when they need to perform a firmware rollout for a given resource, usually by dividing the complete process in different phases, either by geographical locations or different classes of clients.

With the arrival of new paradigms such as NFV or Mobile Edge Computing (MEC) into the marketplace, this problem can become even worst as more (virtual) software-based resources are being dealt with and there is less time between releases.

5.3.3.2 Description of the Use Case

5.3.3.2.1 Overview

As just stated above, this rollout Use Case may become even worst when dealing with (virtual) software-based resources, in particular if dynamic on boarding of VNFs or of other type of applications is supported, in which case automatized and intelligent software rollout becomes vital for operators. With dynamic on boarding, common in DevOps environments, automatic tests to benchmark and building of a profile for a given application is possible and recommended. The subject of performing tests to benchmark network functions is very relevant for network operators and is a common procedure with their physical counterparts.

The flow of actions for both physical and virtualized is similar and should take into account the best practises from Cloud Computing and DevOps. However, since the rollout of virtualized equipment is considered to be more challenging due to the fact that the number of updates for software-based components is performed much more times, this type of update will be the only one considered in the present Use Case.

The current Use Case is further described by the following set of components and features.

5.3.3.2.2 Motivation

By making use of the ENI System, operators can define different policies for different types of rollouts and for different types of resources. One example could be the definition of a hierarchy of parameters for phasing out the rollout, e.g. client class, geographical location, or time of the day. In addition, and also taking dynamic on boarding and DevOps environments into consideration, different types of policies can be defined by using the ENI System, such as:

- Development, e.g. tests should provide a correlation between network function performance (throughput, jitter, delay) and resource utilization (CPU, RAM, I/O).
- Update schedule, e.g. for enterprise customers schedule updates outside business hours.
- Update procedures, e.g. create backup of current versions of software instances when updating instances from platinum level services in order to prevent service disruption in case of occurrence of significant errors.

- Failure procedures, e.g. considering two types of errors where the response would be defined by policies:

  i) critical errors, which make the ENI System stop the update movement process, and initiate the rollback to an already updated instance; and

  ii) minor errors, which makes the ENI System retry the update.

NOTE: In this Use Case, only type ii) errors will be considered.

Thus, the use of AI methods becomes more important when moving software from testing to production by using automatized procedures.

5.3.3.2.3 Actors and Roles

The presence of the following actors/entities as well as their associated roles are envisaged in the current Use Case:

- Customers/clients: the operators themselves.
- Network Administrator: entity/person responsible for the initial policy design that encompasses the definition of different activities during rollouts.
- Network Infrastructure: infrastructure that includes resources that are meant to be upgraded or, in the worst case, replaced.
- ENI System: system solution that makes use of AI methods when upgrading or moving software from testing to production, and that enables the use of policies to govern updates to software instances. This solution also participates in software tests and builds a profile with information, e.g. correlation between network function performance (throughput, jitter, delay) and resource utilization (CPU, RAM, I/O), that can be used to improve fulfilment and assurance procedures.
- OSS/BSS: operational and business systems that belong to the management system of network operators. In this case they are providing, among others, monitoring, actuation, internal records of very different items that may range from products to resources, as well as other business interfaces dedicated to external entities.

5.3.3.2.4 Initial context configuration

The network is operating in perfect conditions with all its components in good shape. Moreover, the network operator already has a development environment that is specified to mimic the production environment. This development environment is used to run automatized tests in order to validate new software versions and build the respective software profile, where the series of tests are defined by network operator policies. Finally, the move of software from development to a production environment is also conditioned by network operator policies, thus governing the phased deployment of the new version.

5.3.3.2.5 Triggering conditions

A new software version of a virtual component is released by the vendor and is on boarded on a network operator infrastructure. The upload of a new software version to software repository triggers the start of automatic tests pre-defined by policies also previously enforced in the ENI System.

5.3.3.2.6 Operational flow of actions

The following sequence of actions may be identified:

1) A new software version is instantiated on the network operator development environment.

2) Within the new environment, the software is subject to a series of tests determined by pre-defined policies in the ENI System, which results, will be used to create a software profile.

3) During the tests, the ENI System starts analysing the behaviour of all the collected data and compares it with the profile of previous versions of software.

4) Since the results of the tests are conformant to previous versions, the ENI System is in position to allow the triggers for moving the new version from the development to the production environment.

5) The ENI System takes into account the pre-defined operator policies for the new software rollout and performs the scheduling of updates for the software instances.

6) The ENI System triggers the movement of the new version from the development to the production environment.

7) During the update, all platinum SLA customers of software instances are using a redundant software instance to avoid any service disruption.

8) Some instances monitoring data may detect an inconsistency with the application profile indicating a problem with the update. Since it is considered a minor error, the ENI System retries the update on failed instances.

9) At the end of the process, the ENI System may notify relevant software components, e.g. OSS/BSS, that the software rollout has been carried out successfully.

5.3.3.2.7 Post-conditions

The new software version has been updated on all deployed instances and inventories. The network and corresponding services are running steady.

5.3.4 Use Case #2-4: Intelligent Fronthaul Management and Orchestration

5.3.4.1 Use Case context

Centralized radio access network (C-RAN) has been extensively considered for emerging and future cellular networks. In C-RAN, centralized RAN functions are located in an entity termed as Centralized Baseband Unit (BBU), and the remainder of radio access connectivity between the UE and the network are handled by Remote Radio Units (RRUs). Such an architecture enables functional split between BBU and RRUs. For example, RF and some physical layer (PHY) level functionalities can be handled at RRUs, while the rest can be moved to centralized BBUs.

Such functional split versus classical Distributed RAN (D-RAN) brings several advantages including accelerated network deployment on RRU side, reduced operating costs (although Capital Expenditure can be high in short term), support for richer multi-node network cooperation and coordination (e.g. on Coordinated Multi-Point systems or Carrier Aggregation) and improved network performance, in particular at the cell edge.

To support such functional split, a Common Public Radio Interface (CPRI) has been proposed to support Fronthaul connectivity, which is the connection between BBU and RRUs. However, CPRI is believed to require strict high bandwidth, low delay, tight synchronization and additional transmission equipment partly attributed to its Point-to-Point connectivity paradigm.

To address the above issues, next generation Fronthaul interface (NGFI) including envisioned new variants of CPRI (eCPRI) target redefining interface flexibility and network functional split between remote and centralized units. Such an interface enables statistical multiplexing on Fronthaul bandwidth, decoupling interface traffic from some RF-level attributes (e.g. number of antennas) and results in more flexible remote unit connectivity to a centralized unit.

In line with recent advancement on NGFI, RRUs are divided into clusters; each cluster may possess one logical entity termed as Remote Aggregation Unit (RAU) that can be physically located as part of one of RRUs per cluster or as a separate individual entity. The RAU is in charge of radio resource management per cluster.

As the functional split can dynamically switch between remote RAU and the centralized entity, the new centralized entity is redefined as Radio Cloud Centre (RCC) to convey multitude of functionalities beyond conventional BBU. RAU and RCC may also refer to the general remote and central entities in any of the next generation Fronthaul technologies.

5.3.4.2 Description of the use case

5.3.4.2.1 Overview

Flexible NGFI opens a new network design paradigm where nodes connectivity between centralized and remote units transforms from Point-to-Point or Point-to-Multi Point into Many-to-Many connectivity comprising hybrid of wired and wireless solutions. In other words, a multi-tier shared network forms the Fronthaul where the slicing of network resources between centralized and remote units can be dynamically tuned in an on-demand fashion.

The new use case is concerned with applying AI technologies, and the resulting interfaces and network components, at the next generation flexible Fronthaul, to facilitate the flexible and dynamic slicing of network resources at the Fronthaul.

5.3.4.2.2 Motivation

Slicing of network resource (especially in a dynamic and flexible manner) at the Fronthaul between remote and centralized units can be complex, as it is affected by multiple factors and their changing contexts - factors such as, the clustering on the remote units (the size, how are they clustered, etc.), the functional split between remote and centralized entities and dimensionality of solution space on network resources to be reserved on the Fronthaul (power, processing capability, radio resources, buffering memory, route to be selected across multiple Fronthaul nodes, etc.).

The application of AI under such context will bring efficient optimization framework, balancing the multiple aspects considered on network resources slicing mentioned above. It will also enable flexible and dynamic resource slicing and functional split at the Fronthaul, considering the changing contexts of the network, such as the changing traffic demand, at the RAU and RCC. As an example of such an application, through load estimation and prediction using the state-of-the-art AI algorithms, the Fronthaul management and orchestration can also be designed in an 'on-demand' manner.

Figure 5-9 shows the concept of the proposed Fronthaul use case.

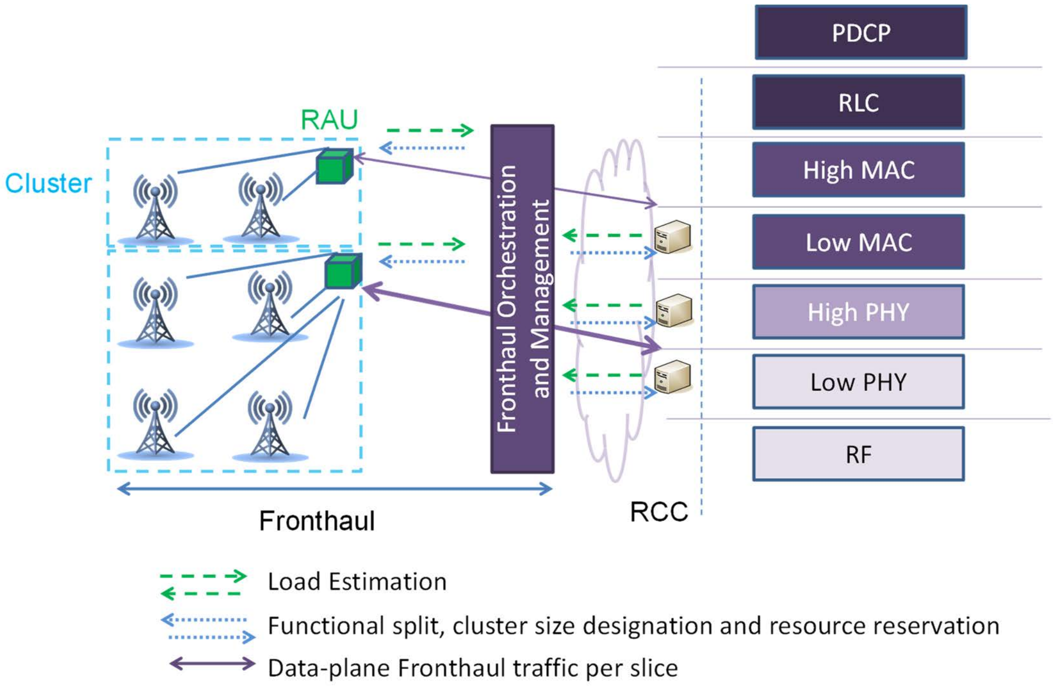

PDCP
RLC
High MAC
Low MAC
High PHY
Low PHY
RF
RAU
Cluster
Fronthaul Orchestration and Management
Fronthaul
RCC
Load Estimation
Functional split, cluster size designation and resource reservation
Data-plane Fronthaul traffic per slice

Figure 5-9: Concept of the proposed Fronthaul use case

5.3.4.2.3 Actors and Roles

- Operator: Provides interfaces to convey signalling on load estimation, service level agreements, slice-level requirements and traffic profiles (if any) to the ENI System.

- ENI System: Collects load estimation data (traffic demand, current configuration) from different RAU/RCC units; determines optimal Fronthaul parameters (e.g. functional split, clustering size per RAU); and reserves Fronthaul network resources accordingly between different RAU and RCC units.

- RAU/RCC units: Provide relevant input data (e.g. on load estimation) via Operator's interfaces to the ENI System and readjust their Fronthaul parameters (e.g. functional split, clustering size per RAU) according to output data from ENI System.

5.3.4.2.4 Initial context configuration

Network is configured with a default set of parameters on, e.g. a target Fronthaul KPI, the functional split, the cluster size, and the corresponding reservation of Fronthaul resources. The default parameters can be set by the network Operator.

5.3.4.2.5 Triggering conditions

When the Fronthaul KPI is below the target.

5.3.4.2.6 Operational flow of actions

1) ENI system collects Fronthaul parameters, current configurations and past and current traffic from different RAU/RCC Units.

2) ENI system makes decision on the Fronthaul parameters and feedbacks the parameters to the RAU/RCC units. Such decision can be made based on experiential learning of the ENI system given the current context.

3) RAU/RCC units receive and readjust their respective Fronthaul parameters.

4) ENI system reserves Fronthaul network resources. Fronthaul operation KPI is fed back to ENI system.

5) When triggered, the ENI system will reconfigure the parameters and reallocate the resources.

5.3.4.2.7 Post-conditions

Target Fronthaul KPI is guaranteed.

5.3.5 Use Case #2-5: Elastic Resource Management and Orchestration

5.3.5.1 Use case context

Vertical markets and industries are addressing a large diversity of heterogeneous services, use cases, and applications in 5G cellular networks. It is currently common understanding that for networks to be able to satisfy those needs, a flexible, adaptable, and programmable architecture based on network slicing is required. Moreover, a softwarization and cloudification of the communications networks is already happening, where network functions (NFs) are transformed from monolithic pieces of equipment to programs running over a shared pool of computational and communication resources. However, this novel architecture paradigm requires new solutions to exploit its inherent flexibility.

This use case addresses mechanisms to exploit the abovementioned flexibility and the concept of resource elasticity, which is a key means to provide an efficient management and orchestration of the computational resources of virtualized and cloudified networks. Elasticity can thus be understood as the ability to gracefully adapt to load changes in an automatic manner such that at each point in time the available resources match the demand as closely and efficiently as possible. These automation mechanisms can greatly benefit from the employment of AI techniques in general and the integration of an ENI system in particular, which would allow optimized decisions to be made based on real data.

5.3.5.2 Description of the use case

5.3.5.2.1 Overview

An elastic management and orchestration of resources can be achieved in different ways. Three different set of elasticity mechanisms can be differentiated, each of them addressing a specific challenge in the overall use case context:

- The computational aspects of network functions have not been taken into account in their original design, hence computationally elastic VNFs can be redesigned to account for those in their operation.

- Flexible mechanisms for orchestration and placement of NFs across central and edge clouds should be designed, considering source and destination hosts resources, migration costs and services' requirements. In particular, latency requirement are a key driver for placement of VNFs at the edge.

- Slicing multiplexing gains due to the sharing of the infrastructure and physical resources need to be fully exploited. Moreover, an efficient network management has to capitalize on the possibility of sharing and re-using the same virtual resources for network slices with similar or identical requirements and shared VNFs.

The three above described challenges are the target of the proposed elastic management and orchestration of resources. To that aim, AI and the ENI system may play an important role as a tool to enhance the performance of elasticity algorithms. Prominent examples of performance-boosting capabilities that could be provided by the ENI system are the following:

i) speeding the service deployment process by realizing an AI-based, automatic, accurate, and reliable mapping from service requests to network slice instantiations;

ii) identification of similarities (in terms of requirements or shared VNFs) across slices to facilitate resource sharing, thus increasing the system resource utilization efficiency;

iii) learning and profiling the computational utilization patterns of VNFs, thus relating performance and resource availability;

iv) traffic prediction models for proactive resource allocation and relocation;

v) optimized VNF migration mechanisms for orchestration using multiple resource utilization data (CPU, RAM, storage, bandwidth), and vi) optimized elastic resource provisioning to network slices based on data analytics. In the following, details are provided to the description of this use case.

5.3.5.2.2 Motivation

As previously mentioned, there is a need to design mechanisms that allow the network infrastructure to become flexible enough to host the heterogeneous set of verticals that 5G is meant to address, where the flexibility enabled by this use case is indeed beneficial not only for wireless networks, but also for fixed networks. An elastic resource management and orchestration increases the flexibility of the network by allowing a very efficient utilization of the resources that gracefully adapts its behaviour to the load and the available resources at every time. Furthermore, networks and network slices get currently over-dimensioned in their computational capabilities for cases of peak load. With this use case, a more autonomous and intelligent self-dimensioning of the network is targeted, along with a smart redistribution of the computational resources.

5.3.5.2.3 Actors and Roles

The AI-assisted "elastic" network management and orchestration is enabled by a predisposition to elasticity of the whole network infrastructure that provides end-to-end services through network slicing. However, this predisposition can be achieved with the standard 3GPP and ETSI NFV architecture, where management and orchestration functionalities of several architectural elements would be enhanced with elastic capabilities. In particular, the following architectural elements and elements play an active role in the current Use Case:

- Management and Orchestration System: it is composed of the functions from different network, technology, and administration domains (such as 3GPP public mobile network management, ETSI ISG NFV Orchestration) that manage network slices and related communications services across multiple management and orchestration domains in a seamless manner.

- Network Slice Management Function (NSMF): it is part of the Management and Orchestration System, e.g. 3GPP public mobile network management [i.6], or it is an external entity in systems compliant with ETSI ISG NFV Orchestration [i.7]). NSMF would use AI to extend the 3GPP NSMF/NSSMF functionalities, in order to support the elastic intra-slice (or cross-domain) orchestration and the elastic cross-slice orchestration. The former deals with the orchestration of the different VNFs part of the same slice across multiple domains, while the latter addresses the joint orchestration of the multiple slices deployed on a common architecture. The NSMF also includes functions related to performance monitoring, measurement, and alarm. It is also in charge of defining and instantiating elastic slices, creating first the slice blueprint based on the service-related resource requirements and then defining the appropriated Network Slice Instance.

- Elastic Slice: a set of VNFs and the associated resources to support a mobile service with elastic (non-stringent) requirements that admit graceful performance degradation. This allows e.g. more flexibility in the allocation of resources and in the deployment of the associated VNFs.

- Elastic VNFs: they can be (re-)designed with elastic principles in mind such that the computational resources available for its execution are taken into account, or its temporal and/or spatial interdependencies with other VNFs are mitigated.
- ENI System: system solution that provides a set of AI methods (e.g. supervised/unsupervised and reinforcement learning schemes) to the Elastic Network Slice Management Function.

5.3.5.2.4 Initial context configuration

Consumer-facing service descriptions are mapped to network slice "blueprints". Based on the slice blueprints, a running network slice instance (NSI) is selected or created. Once the NSI deployed, the AI schemes can be used to predict network loads, estimate resource usages, and react accordingly by activating elastic Cross-slice (or Intra-slice) Orchestrator functions in order to optimize the resource usage across slices and prevent system faults.

5.3.5.2.5 Triggering conditions

The ENI System may recommend or enforce the application of one or more algorithms for an elastic (re-)orchestration of resources when at least one of the following events happens:

- A new service request arrives.
- The resource requirements of a new slice cannot be satisfied in the current system configuration.
- The amount of resources allocated to one instantiated slices exceeds a given "efficiency" threshold.
- The requirements of running services change (or is predicted to change) and become substantially more stringent.
- A risk of imminent resource shortage is detected.

5.3.5.2.6 Operational flow of actions

During the slice setup process, the ENI System may be used first to define the slice blueprint; then, based on the slice blueprint, to identify whether it exists one deployed NSI that can support the new service, with a minimum amount of additional resources. Based on this, the resource required are allocated, the slice is instantiated and managed during its lifecycle.

If there are not enough resources available prior to the slice instantiation or an alarm notifies congestions, the ENI System may be used to support the following "elastic" system adaptation functions:

1) Elasticity solutions at the VNF level: VNF computational resource scaling and graceful degradation of performance.
2) Elasticity solutions at the intra-slice level: migration of VNFs to different clouds, to create room for other VNFs with tighter (latency or computational) requirements or enhance the performance of the migrated VNFs.
3) Elasticity solutions at the cross-slice level: cross-slice resource management to maximize resource sharing and optimize the resource utilization efficiency.

The three (families of) elasticity functions mentioned above can be jointly executed and are not mutually exclusive. Nonetheless, in general, they act at different time scales and involve different hierarchical elements of the network architecture (e.g. cross-domain or per-domain).

5.3.5.2.7 Post-conditions

The elastic Network Slice Management Function entails an improvement in the exploitation of the network resources. On the one hand, less resources are employed to guarantee the same QoS. On the other hand, more service requests can be accepted and treated at the same time, improving the network efficiency and reducing redundancy in resource exploitation. Network slicing is re-organized still meeting non-elastic slice requirements.

5.3.5.3 Mapping to ENI reference architecture

5.3.5.3.1 Functional blocks

As discussed in [i.9], the ENI system can enhance the current operator policies for network management with AI. Specifically, this use case relates to the elastic management and orchestration procedures, as described above. The elastic (re-)orchestration algorithms needed to implement the aforementioned operations will reside in the management and orchestration functional blocks of the Assisted System. The ENI system continuously analyses the network performance and, whenever it consider it necessary, may trigger the elastic network management and orchestration mechanisms. The Assisted System receives inputs and instructions from the ENI system that are based on its learning and inferring capabilities. The communication between the elastic Management and Orchestration System of the network and the ENI system happens through the Reference Points specified in [i.9] and their relevant Interfaces. In the following, the Functional Blocks of the ENI architecture involved in this use case is described.

All the Functional Blocks of the ENI system described in [i.9] play a role in this use case, according to their defining functions and tasks. The work of the following blocks is of particular importance:

- Data Ingestion and Normalization Functional Block: it shall collect from all the layers of the network data on performance KPIs, resource availability and state, network slice instances, etc. The more efficiently these data are gathered, the more effective can be the optimization recommended or enforced by the ENI system.

- Context-Aware Management Functional Block: this block is continuously used to update the context in which the ENI system's decisions are made.

- Situation Awareness Functional Block: this block has the goal to estimate and evaluate how the ENI system's decisions impact the assisted network. It should prevent that poor decisions lead to violations of the requirements and SLAs of network slices.

- Cognition Framework Functional Block: this block concretely applies the AI algorithms that lead to an elastic management and orchestration of resources and to the optimization of the network activities. As also proposed in ETSI ISG ENI's PoC#2 "Elastic Network Slice Management", the different kinds of elasticity are enabled by the decisions taken by the Cognition Framework Functional Block, at different levels. More in detail:

    - **Intelligent admission control:** when a new service request arrives, the Operator Management Systems (OSS and BSS) should check the feasibility of serving an additional network slice in the system according to very different viewpoints, e.g.: the kind and amount of resources available in the system, the kind and amount of already provisioned slice in the system or the forecast of future load. After this evaluation, the intelligent admission control system shall take a decision on whether accept the slice (and thus re-orchestrate the network accordingly) or reject (and keep resources free). This approach fits very well with the role of the Cognition Framework Functional Block, which shall act to maximize (resp. minimize) the provider's revenues (in respective to network's health). That is, in a scenario in which network slices bring different revenues to the infrastructure provider, the system may learn to reject requests that are economically not efficient. On the other hand, if the decision has to be taken on a purely infrastructure occupation basis, the effort needed to re-orchestrate the network shall be taken into account. Thus, slices that may require a substantial re-orchestration of the network will be rejected.

    - **Elastic re-orchestration:** once slices are admitted in the system, they have to be served according to the required set of SLAs between the tenants and the operator. As one of the fundamental tasks of an ENI system is to be very efficient, the elastic re-orchestration function of the Cognition Framework Functional Block shall predict the future load of each network slice to efficiently compose the VNFs in order to minimize the resource utilization while keeping the KPIs stable. This is a forecasting problem that builds on the feature extraction of the traffic of each network slice. By building on this load forecast (it can be either short- or long-term) the elastic re-orchestration may trigger:

        i) the scaling of a specific VNF (up or down);

        ii) its relocation; or

        iii) the amendment of the Service Function Chain that compose a Network Slice (i.e. adding or removing one network function).

- **Elastic VNFs:** while the aforementioned items relate mostly with the management and orchestration of the system, there is a third dimension of elasticity which directly tackles the design of a VNFs. Analogously to the elastic re-orchestration, a VNF may re-shape its behaviour depending on i) the predicted load and ii) the amount of assigned (computational) resources. This can be applied to any kind of VNF in the system, but advantages will be higher with the most resource-consuming ones, like e.g. RAN- or routing-related VNFs. This functionality is highly intertwined with the elastic orchestration described above as:

  i) elastic orchestration decision shall be taken considering the "elasticity" of the VNFs; and

  ii) elastic operations shall be used only when a wrong resource assignment has been performed, so to efficiently avoid an abrupt interruption of the functionality and rather provide a graceful degradation of the performance until new resources are assigned.

5.3.5.3.2 Interfaces

According to [i.9], the elastic management and orchestration use case shall employ the following Reference Points and related interfaces:

- The Reference Point towards BSS (Ebss-eni-reg) to lead, perform, or optimize the slice admission control in the system.
- The Reference Point towards OSS (Eoss-eni-dat and Eoss-eni-cmd) to gather information from the OSS about the current status of the system and possibly enforce re-orchestration commands.
- The Reference Point towards the Orchestrator (Eor-eni-cmd and Eor-eni-cfg) to enforce lower-level re-orchestration procedures in the system.
- The Reference Point towards the Infrastructure (Einf-eni-dat and Einf-eni-cmd) to gather information about the current load of the system and perform lower-level configuration on the elastic VNFs.

5.3.5.3.3 Flow of information

The ENI system continuously gathers data and information through the abovementioned interfaces from the different layers of the network. These data are made available to the Context-Aware Management, the Situation-Awareness, and the Cognition Framework Functional Blocks by the Data Ingestion and Normalization and the Knowledge Representation Functional Blocks. The latter collect the information and make it "readable" to the other Functional Blocks of the ENI system. Then, the Cognition Framework Functional Block runs the AI algorithms that enable elasticity management and orchestration, taking as input the data coming from the network, the context analysis provided by the Context-Aware Management Functional Block. The decisions elaborated by the Cognition Framework Functional Block need to be compliant with the inputs or constraints that may come from the Situation Awareness and the Policy Management Functional Blocks. Once validated, those decisions are then translated by the Denormalization and Output Generation Functional Block into instructions for the Network Management and Orchestration System of the assisted network. This data and information exchange is illustrated as an example in Figure 5-10.

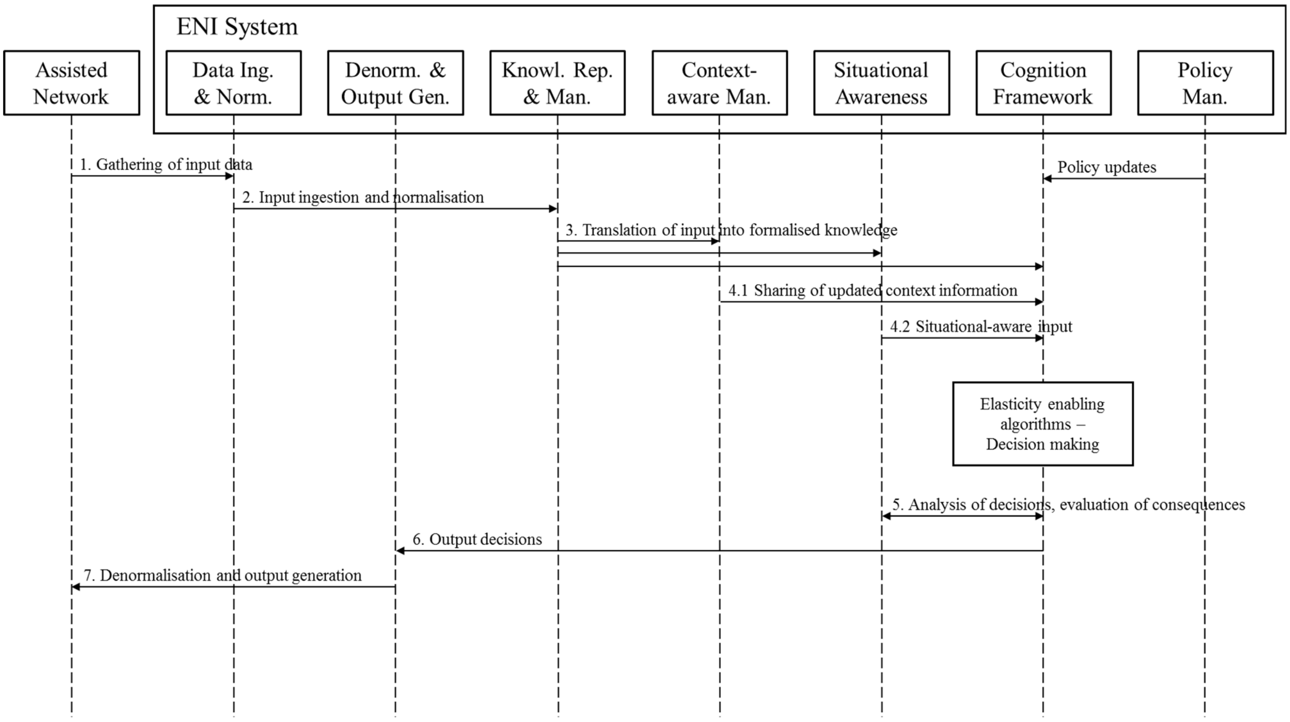

ENI System
Assisted Network
Data Ing. & Norm.
Denorm. & Output Gen.
Knowl. Rep. & Man.
Context-aware Man.
Situational Awareness
Cognition Framework
Policy Man.
1. Gathering of input data
Policy updates
2. Input ingestion and normalisation
3. Translation of input into formalised knowledge
4.1 Sharing of updated context information
4.2 Situational-aware input
Elasticity enabling algorithms – Decision making
5. Analysis of decisions, evaluation of consequences
6. Output decisions
7. Denormalisation and output generation

Figure 5-10: Example Flow of information among the ENI System's functional blocks and the Assisted System

The flow of information and the exchange of messages between the assisted network and the ENI system shall be as frequent as needed to guarantee the timeliness and effectiveness of the decisions recommended or enforced, but their frequency can vary according to the network load and the speed or significance of the changes in measured or inferred context. The delivery of instructions and recommendations from the ENI system to the assisted Network Management and Orchestration System can happen at different "time granularities" for the different elasticity mechanisms. This can depend on the different complexities of the involved machine learning algorithms that run in the ENI system or on the urgency and "costs" (in term of complexity, time, impacted resources, etc.) related to the implementation of the decisions.

5.3.6 Use Case #2-6: Application Characteristic based Network Operation

5.3.6.1 Use case context

While Self-Organizing Network (SON) capabilities have advanced over the past decade, the work centres mainly on self-configuration, self-healing, and self-optimization of network performance. However, a network with good network performance KPI is not always a reliable indicator of the user's perception of the Quality of Experience - QoE gleaned from using services on that network. For example, non-network related performance measurements can also impact user QoE including service provisioning time, repair times, customer complaint resolution time, brand perception, and more. Moreover, there could be subtle combinations of real time KPI but that when averaged over a period of time may appear sufficient, but the interim variations can have a strong negative impact on user experience. The real impact has to be measured at an application performance level as this is what the user experiences first hand.

The use case proposes a study of how user perception of service quality QoE relates to expectations for application performance, and network performance.

The latter can be challenging to detect since there can be thousands of KPI per second in a real E2E network service flow for a single user, hence the motivation for AI/ML learning techniques that can process massive amounts of performance and management data and detect subtle patterns that can have a large impact on user experience.

5.3.6.2 Description of the use case

5.3.6.2.1 Overview

3GPP specifies use cases for SON, but:

- Has not yet any targeted solution to improve wireless performance and further user QoE.
- Has not proposed any feasible proposal for self-planning.
- Has not yet defined any practical algorithms to implement SON.

The use case proposes a study of how User perception of service quality QoE relates to expectations for application performance, and network performance.

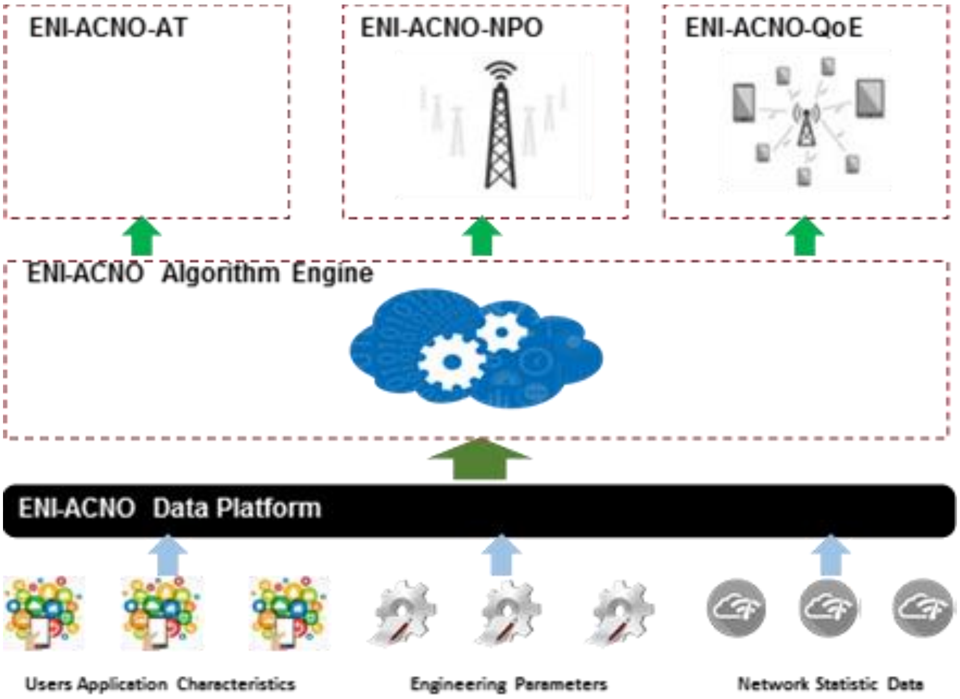


Figure 5-11: An illustration of the ENI-ACNO framework

5.3.6.2.2 Motivation

SON is driven by performance optimization rather than ultimately improving user QoE. The impact of application characteristics on network performance and further on QoE are not also considered in 3GPP SON. This work intends to fill that critical gap by developing and implementing an ENI-ACNO system (illustrated in Figure 5-11 above), that will optimize network performance and QoE.

5.3.6.2.3 Actors and Roles

- Mobile Network Operator Clients for the ENI-ACNO system
- Network Infrastructure:
  - 3G/4G/5G Wireless communication networks

- ENI-ACNO System Framework:
    - ENI-ACNO Data Platform collects network performance statistics and DPI statistics
    - ENI-ACNO Algorithm Engine profiles cell traffic and application characteristics through clustering and labelling analytics
    - ENI-ACNO-AT optimizes capacity and coverage through automatically tuning antenna azimuth and down tilt
    - ENI-ACNO-NPO optimizes the targeted cell KPIs through automatically tuning cell engineering parameters
    - ENI-ACNO-QoE improves user QoE through automatically tuning cell engineering parameters
- Operations System:
    - ENI-ACNO is the system to implement and execute the policies of AT, NPO, and QoE through automatically tuning the corresponding network parameters
    - ENI-ACNO engine prioritizes the targeted performance indicators to be optimized based upon cell profile

5.3.6.2.4 Initial context configuration

- The ENI-ACNO system studies cell traffic patterns and corresponding application QoE characteristics by utilizing clustering algorithms.
- The ENI-ACNO system determines the targeted and prioritized KPIs for optimization based upon the individual cell configuration (profile and label).
- The ENI-ACNO-AT learns how to optimize capacity and coverage through tuning cell engineering parameters, such as, but not limited to cell azimuth and down tilt.
- The ENI-ACNO-NPO learns how to optimize the targeted KPIs through tuning cell engineering parameters, such as received power, etc.
- The ENI-ACNO-QoE learns how to optimize user QoE through tuning cell engineering parameters which impact certain KPIs and also impact user perception of Quality.

5.3.6.2.5 Triggering conditions

- Application performance changes.
- Existing capacity and coverage cannot meet the capacity and coverage threshold.
- Existing network performance cannot meet the performance thresholds.
- Existing user QoE cannot meet the QoE threshold.

5.3.6.2.6 Operational flow of actions

- **Cell Labelling and Clustering**
    - Customers label each cell cluster according to their application characteristics, and determines the relevance/priority of targeted KPIs for each cell.
- **Cell Application Characteristic Profiling**
    - ENI-ACNO system collects and analyses the application characteristics information indicating the application usage pattern of each cell in the network.

- **Targeted KPIs Identification and Prioritization**
  - ENI-ACNO system labels each cell according to its application characteristics and prioritizes the targeted KPI to be optimized.
- **Cause Effect Deriving**
  - ENI-ACNO-AT system derives the relationship between the engineering parameters and the cell capacity and coverage performance.
  - ENI-ACNO-OPN system derives the relationship between the engineering parameters and network performance indicators.
  - ENI-ACNO-QoE system derives the relationship between the engineering parameters and user QoE.
- **Engineering Parameter Tuning**
  - ENI-ACNO-AT adjusts the engineering parameters (such as cell down tilt, azimuth) to optimize the capacity and coverage.
  - ENI-ACNO-OPN adjusts the engineering parameters (such as received power) to optimize the network performance.
  - ENI-ACNO-OPN adjusts the engineering parameters (such as received power) to optimize the user QoE.

5.3.6.2.7 Post-conditions

- The engineering parameters are dynamically tuned according to the behaviour change of application characteristics.
- After the engineering parameters are tuned:
  - Capacity and Coverage will be optimized
  - Targeted KPIs will be optimized
  - User QoE will be improved

5.3.6.3 Mapping to ENI reference architecture

5.3.6.3.1 Functional blocks

The functional blocks for application characteristic based network operation using AI is shown in Figure 5-12.

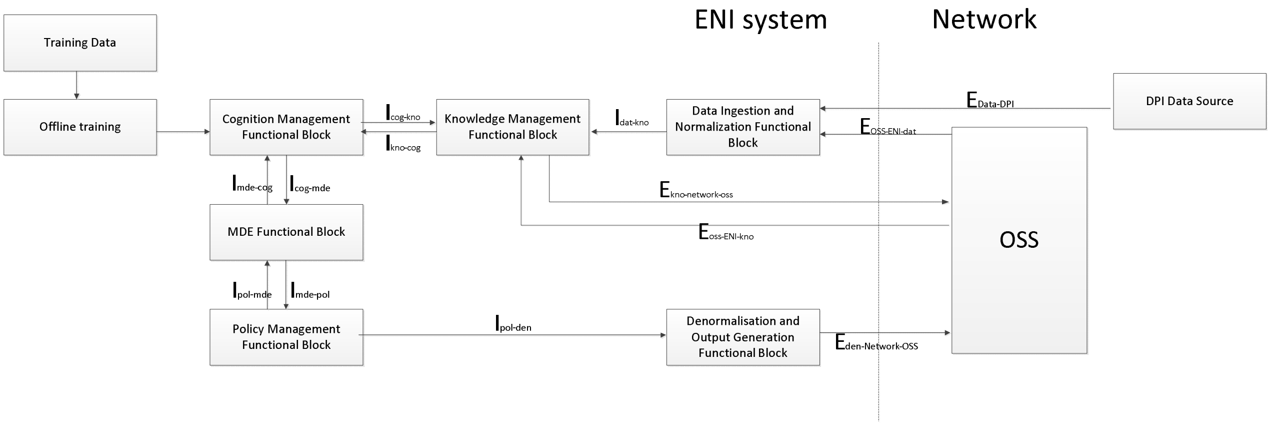


Figure 5-12: Mapping to ENI reference architecture for Application Characteristic based Network Operation

For training data, it can be pre-prepared one or the one from OSS and DPI data Source.

The "Data ingestion and Normalization Functional block" collects the network parameters like network performance from OSS and DPI data from "DPI Data Source". These data are filtered and normalized accordingly.

The "Knowledge Management Functional Block" generates inferences and passes that as well as data to "Cognition Management Functional Block". This functional block would also evaluate performance of model in "Cognition Management Functional Block". When model performance deteriorates, an updated feature set (including feature name and parameter) without invalid features is generated based on a feature set corresponding to the current data pattern (e.g. distribution and statistic characteristics, etc.) and a score table of current features of model (including feature name, validity score and validity flag), in which the validity score of a feature is negatively related to correlation with other features. And the feature list and this score table of features will be exchanged with OSS for review, modification and confirmation.

The "Cognition Management Functional Block" utilizes data and inferences to generate predictions based on modelling provided by offline training. For the modelling in "Cognition Management Functional block", it will be iterated periodically based on the data from DPI Data Source and OSS as well as updated features and parameters generated by "Knowledge Management Functional Block".

The "MDE Functional Block" translates the predictions into the form that is understandable by "Policy Management Functional Block". More usages of "MDE functional block" will be further investigated.

The "Policy Management Functional Block" makes the polices and inputs the decisions to "Denormalization and Output Generation Functional Block", which generates execution command to OSS.

5.3.6.3.2 Reference Point

$E_{OSS\text{-}ENI\text{-}dat}$ defines data exchange between OSS and ENI system.

$E_{Data\text{-}DPI}$ defines data exchange between DPI data source (e.g. certain entity in core network or data centre) and ENI system.

Reference Point $I_{dat\text{-}kno}$ defines internal Reference Point between "Data ingestion and Normalization Functional block" and "Knowledge Management Functional Block".

Reference Point $I_{kon\text{-}cog}$ and $I_{cog\text{-}kon}$ define internal Reference Point between "Knowledge Management Functional Block" and "Cognition Management Functional Block".

Reference Point $I_{mde\text{-}cog}$ and $I_{cog\text{-}mde}$ define internal Reference Point between "Cognition Management Functional Block" and "MDE Functional Block".$I_{mde\text{-}pol}$ and $I_{pol\text{-}mde}$ define internal Reference Point between "MDE Functional Block" and "Policy Management Functional Block".

$I_{pol\text{-}den}$ defines data exchange between "Policy Management Functional Block" and "Denormalization Functional and Output Generation Block".

$E_{den\text{-}Network\text{-}OSS}$ defines data exchange between "Denormalization Functional and Output Generation Block" and OSS.

$E_{kno\text{-}network\text{-}oss}$ and $E_{oss\text{-}ENI\text{-}kno}$ define data exchange between "Knowledge Management Functional Block" and OSS.

5.3.6.3.3 Flow of information

The flow of information is shown in following Figure 5-13.

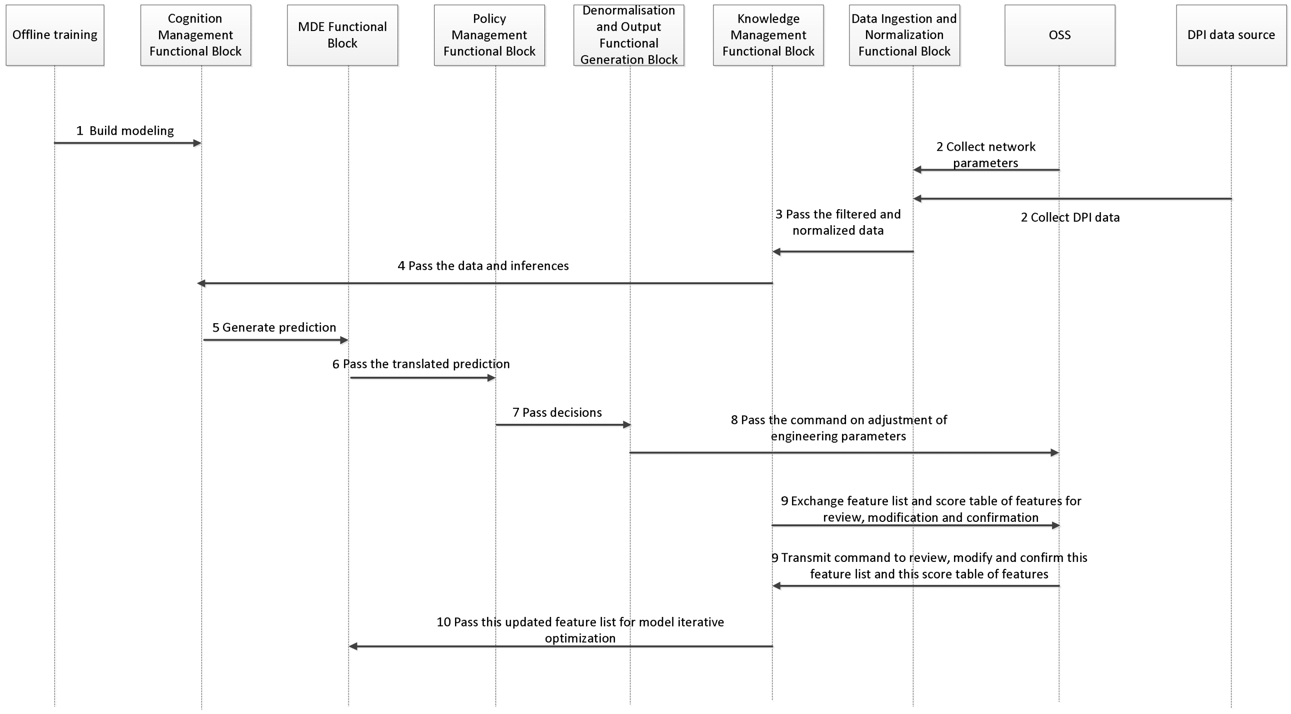

Offline training
Cognition Management Functional Block
MDE Functional Block
Policy Management Functional Block
Denormalisation and Output Functional Generation Block
Knowledge Management Functional Block
Data Ingestion and Normalization Functional Block
OSS
DPI data source
1 Build modeling
2 Collect network parameters
2 Collect DPI data
3 Pass the filtered and normalized data
4 Pass the data and inferences
5 Generate prediction
6 Pass the translated prediction
7 Pass decisions
8 Pass the command on adjustment of engineering parameters
9 Exchange feature list and score table of features for review, modification and confirmation
9 Transmit command to review, modify and confirm this feature list and this score table of features
10 Pass this updated feature list for model iterative optimization

Figure 5-13: Flow of information for Application Characteristic based Network Operation

Step 1 is about building modelling via offline training.

Step 2 is that "Data ingestion and Normalization Functional block" collects data like network parameters from OSS and DPI data from DPI data source like certain entity in core network or data centre.

Step 3 is that the filtered and normalized data is passed to "Knowledge Management Functional Block", which will pass the data and inferences in step 4.

In step 5, "Cognition Management Functional Block" generates predictions and pass it to "MDE functional block", which is a model-driven functional block based on latest architecture.

In step 6, the translated predictions that are understandable by "Policy Management Functional Block" are passed.

In step 7, the decisions are passed to "Denormalization Functional and Output Generation Block".

In step 8, command on adjustment of engineering parameters is passed to OSS.

In step 9, "Knowledge Management Functional Block" evaluates model performance and sends this feature list and this score table of features to OSS for review, modification and confirmation. Meanwhile, the command for review, modification and confirmation is transmitted from OSS to "Knowledge Management Functional Block".

In step 10, the updated feature set is delivered to "Cognition Management functional block" for model iterative optimization.

5.3.7 Use Case #2-7: AI enabled network traffic classification

5.3.7.1 Use case context

Network traffic classification plays an important role in network operation and management. Based on traffic classification techniques (e.g. port-based technique, payload-based technique), unknown traffic is categorized into a number of classes at the level of protocol (e.g. HTTP, SIP), application (e.g. video, voice, download, instant messaging, online games, and virtual reality), etc.

Network traffic classification is regarded as a fundamental work in 5G techniques (e.g. network slicing, network orchestration, and user plane policy rules). It is very essential for network operators to classify network traffic. On one hand, by processing different traffic classes respectively, network traffic classification supports numerous network closed-loop control activities in terms of network security, traffic engineering, Quality of Service (QoS) and etc. On the other hand, by providing traffic change or distribution information at network or service level, to support policy-making processes. Besides, classification results sever as the guideline for Operations Support System (OSS) and traffic forecast.

In addition, nowadays more and more traffic in the network are encapsulated in encrypted ways. HTTP over TLS/SSL (well known as HTTPs) and many other private encrypted protocols used by 3rd parties hide key features of Application Level in flows, posing great challenges to network traffic classification's effect. So a more intelligent and innovative mechanism needs to be introduced into future network to face such situation.

5.3.7.2 Description of the use case

5.3.7.2.1 Overview

In this use case, one or more machine learning classifiers are trained and applied in the ENI system to classify real-time network traffic. In the model training phase, designated network traffic is captured from network interfaces according to predefined rules, which is labelled to a corresponding class automatically. The raw bit streams or extracted features (e.g. port, packet length, inter-packet arrival time, and session time) compose to the training data set. Then target model is trained and generated based on machine learning algorithms (e.g. Random Forests, Convolutional Neural Network, and Recurrent Neural Network). In the inference phase, the well-trained modules are implemented in ENI system and play the role of network traffic classification. The model parameters are dynamically adjusted according to the accuracy of classification results. Besides, based on the incremental learning, the model is further trained to adapt to application updating. It does not retrain the model.

5.3.7.2.2 Motivation

Network traffic classification can be achieved by various methods, such as port-based technique and payload-based technique. However, with the growth in the diversity of applications, traffic volume and the proportion of encapsulated traffic, traditional methods mentioned above are inefficient and even fail to classify network traffic.

- Port-based technique: determining application layer protocol by a firstly registered port in Internet Assign Number Authority (IANA). This method is efficient and low resource-consuming, but not reliable as many application use dynamic port numbers (e.g. P2P).
- Payload-based technique: Deep Packet Inspection (DPI) is widely used in telecom networks, which inspects characteristic signatures in the packet payload to identify an application traffic. The method has high accuracy, but has high complexity and labour cost for matching the signature strings in packet Characteristic signatures shall be kept up to date, as the applications change very frequently. In addition, it cannot be used for encapsulated traffic.

In this use case, AI enabled Network Traffic Classifier (AI-NTC) is proposed in the ENI system, which deals with the situation of non-available payloads or dynamic ports.

AI has achieved a better-than-human recognition rate in the field of image classification. Therefore, the AI-NTC transforms bit streams or extracted features into images and models the network traffic classification as the 'traffic image' classification. That is, by collecting data at the transport level (instead of application level) and applying machine learning algorithms, the ENI system is able to achieve accurate classification result with a relatively low overhead.

Such a scenario is illustrated in Figure 5-14.

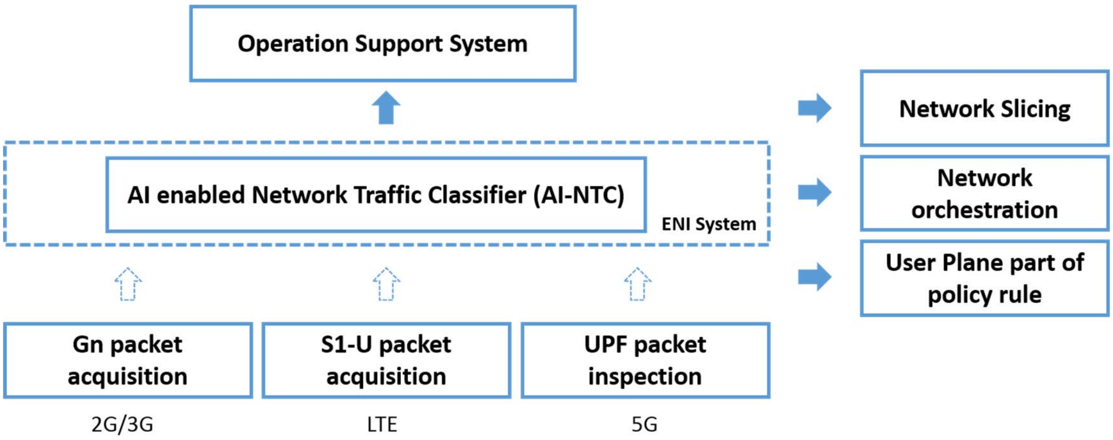


Figure 5-14: Scenario of AI enabled network traffic classification

5.3.7.2.3 Actors and Roles

- Network Administrations: entity/person who defines target traffic classes and usage of classification results.
- Network Infrastructure: network resources for collecting target traffic data to ENI system, and management systems whose policies are based on traffic classification results.
- ENI System: system solution used to receive target traffic data from network infrastructure; to classify traffic data into target classes; to provide results to outer network systems in predefined formats (e.g. Ipflow protocol defined in IETF RFC 6645 [i.11], etc.).

5.3.7.2.4 Initial context configuration

- Mapping target classifications to predefined labels.
- Defining target feature types extracted from traffic datasets or raw bit streams.
- Defining classification results formats.
- AI-NTC in ENI system is enabled with AI capability, which is initialled to learn target traffic classification patterns through target features or bit streams in training data set automatically.
- Related management systems connects to ENI system for acquiring classification results.

5.3.7.2.5 Triggering conditions

AI-NTC provides the classification results to outer related management systems, which check the change of the traffic classification distribution pattern. When the pattern changes drastically (for instance) and the current network resource usage cannot meet the requirement of this new pattern. Some network re-organization actions would happen step by step (e.g. allocating more UPFs, assigning different uplink-classifier policies for steering some parts of the traffic).

5.3.7.2.6 Operational flow of actions

ENI system with AI-NTC is intended to enhance the capability of traffic classification and optimize the network management based on the real time classification results of traffic. This kind of mechanism is realized by introducing the new AI capability (e.g. normally based on the deep-learning algorithm and architecture), with the following flow of activities:

1) ENI system initiates a process to collect training traffic (can be extracted from the running network or off-line uploading) with specific classification labels, and uses the labelled data to train the classification module. This process can be on-line all the time or for some time periods predefined.

2) ENI system initiates another process to infer the target traffic's classes by loading the well-trained module mentioned in 1). This process can be realized in real time.

3) ENI system arranges the classification results and provides them to outer management systems.

4) The related system receives the classification results and adjusts the policies according to the predefined rules.

5.3.7.2.7 Post-conditions

- The network resources such as network slices are re-allocated and re-adjusted according to the real-time traffic classification results. The changed network is well optimized based on that current traffic pattern.

- The module parameters are dynamically adjusted according to the classification result's evaluation from the network, by means of reinforcement learning.

The model can adapt to application updates by means of incremental learning.

5.3.7.3 Mapping to ENI reference architecture

5.3.7.3.1 Functional blocks

The mapping to ENI reference architecture for the use case of 'AI enabled network traffic classification' is illustrated as Figure 5-15.

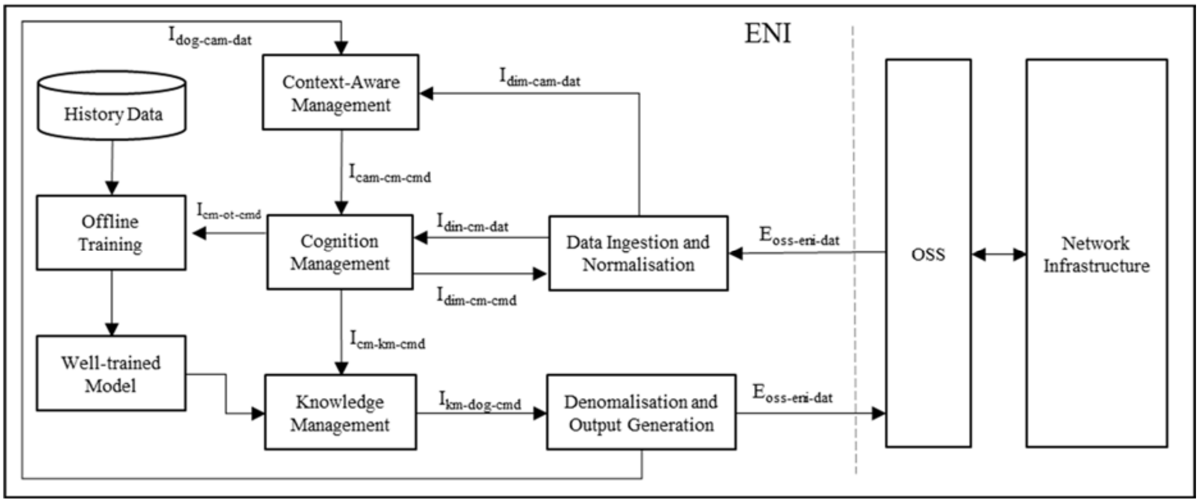

ENI
$I_{dog\text{-}cam\text{-}dat}$
Context-Aware Management
$I_{dim\text{-}cam\text{-}dat}$
History Data
$I_{cam\text{-}cm\text{-}cmd}$
Offline Training
$I_{cm\text{-}ot\text{-}cmd}$
Cognition Management
$I_{din\text{-}cm\text{-}dat}$
Data Ingestion and Normalisation
$E_{oss\text{-}eni\text{-}dat}$
OSS
Network Infrastructure
$I_{dim\text{-}cm\text{-}cmd}$
$I_{cm\text{-}km\text{-}cmd}$
Well-trained Model
Knowledge Management
$I_{km\text{-}dog\text{-}cmd}$
Denomalisation and Output Generation
$E_{oss\text{-}eni\text{-}dat}$

Figure 5-15: Mapping to ENI reference architecture

Data Ingestion and Normalization Functional Block collects the traffic data and extracted features based on the classification tasks and then normalized them to a common format.

Context-Aware Management Functional Block defines target traffic classes corresponding to personalized and customized requirements. For example, traffic is categorized into video application class or webpage application class in some cases. This also gathers environment information such as application updating, classification accuracy and so on, then determines whether the classification model should be adjusted.

Cognition Management Functional Block defines which features should be extracted according to target traffic classes provided by context. The functional block understands ingested data and transfers it to Knowledge Management Functional Block. This block also understands context to generate new knowledge.

Knowledge Management Functional Block stores the machine learning model and performs inferences by using the well-trained model and extracted features to get classification results. This block also stores the generated knowledge and defines a formal and consensual representation of knowledge so that the computer system could implement the machine learning algorithms and perform reasoning using the knowledge representation.

Denormalization and Output Generation Functional Block translates from a normalized classification label to a form that the OSS can understand and then transfers it to OSS. According to the real-time traffic classification results, OSS can take some actions to optimize network performance.

5.3.7.3.2 Interfaces

$E_{oss-eni-dat}$ defines data exchange between the ENI System and the OSS (Assisted System). The OSS collects traffic data from network infrastructure that belongs to different domains (e.g. RAN/Fixed Access, Transport, and Core) and sends it to ENI system.

$E_{eni-oss-dat}$ defines data exchange between the ENI System and the OSS. The ENI System provides real-time traffic classification results so that the OSS could take actions accordingly, e.g. checking the change of the traffic classification distribution pattern.

$I_{din-cm-dat}$ defined data exchange between Data Ingestion and Normalization Functional Block and Cognition Management Functional Block. Data Ingestion and Normalization Functional Block passes the normalized data to Cognition Management Functional Block to understand and analyse.

$I_{din-cm-cmd}$ defined recommendations and/or commands between Data Ingestion and Normalization Functional Block and Cognition Management Functional Block. Cognition Management Functional Block passes the feature extraction rules to Data Ingestion and Normalization Functional Block.

$I_{cam-cm-cmd}$ defined recommendations and/or commands exchanged between Context-aware Management Functional Block and Cognition Management Functional Block. The commands include traffic classification information and context information used for model updating.

$I_{cm-km-cmd}$ defined recommendations and/or commands exchanged between Cognition Management Functional Block and Knowledge Management Functional Block. Cognition Management Functional Block informs Knowledge Management Functional Block to perform inferences or generate new knowledge for storage.

$I_{km-dog-cmd}$ defined data exchange between Knowledge Management Functional Block and Denormalization and Output Generation Functional Block. The data includes the traffic classification labels inferred by ENI system.

$I_{cm-ot-cmd}$ defined recommendations and/or commands exchanged between Context-aware Management Functional Block and Offline Training Functional Block. Context-aware Management Functional Block informs Offline Training Functional Block to adjust module parameters or retrain the traffic classification model.

$I_{dog-cam-dat-}$defined data exchange between Context-aware Management Function Block and Denormalization and Output Generation Functional Block. Context-aware Management Functional Block receives the classification results and the classification accuracy.

$I_{dim-cam-dat-}$ defined data exchange between Data Ingestion and Normalization Functional Block and Context-aware Management Function Block. Data Ingestion and Normalization Functional Block passes the information of application updating.

5.3.7.3.3 Flow of information

The flow of information is given in Figure 5-16.

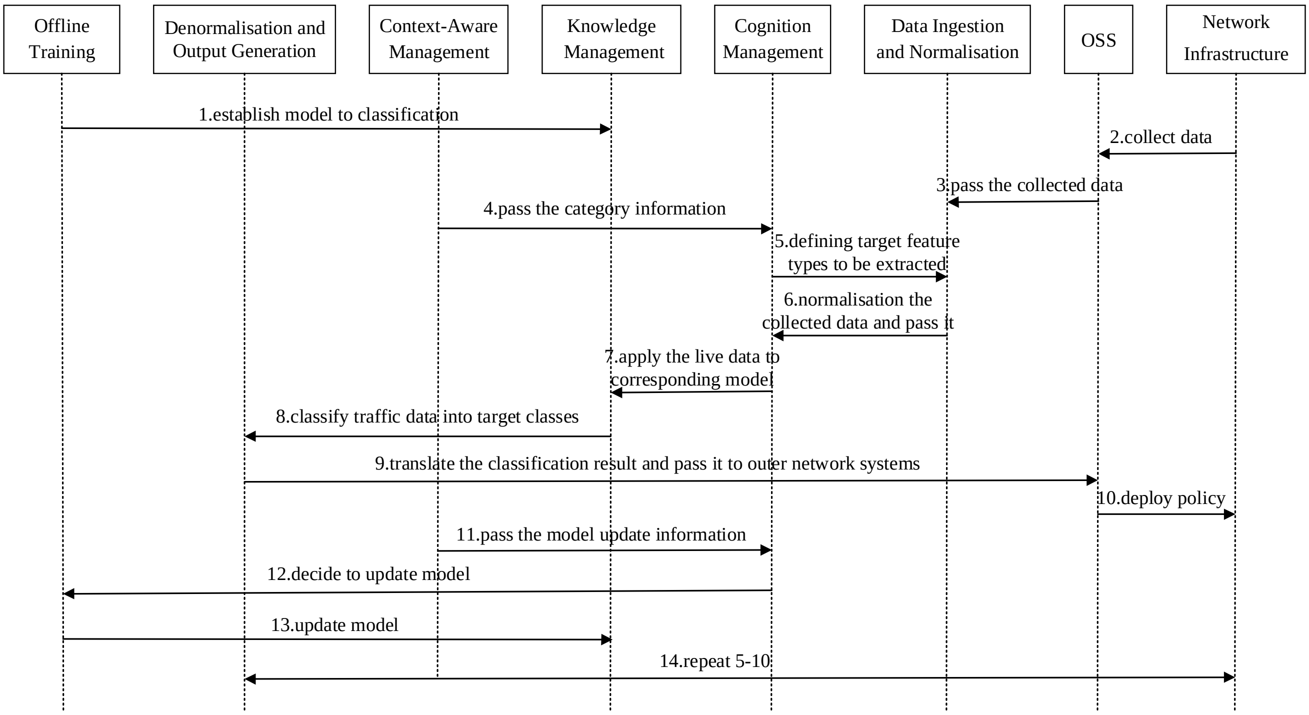

Offline Training
Denormalisation and Output Generation
Context-Aware Management
Knowledge Management
Cognition Management
Data Ingestion and Normalisation
OSS
Network Infrastructure
1.establish model to classification
2.collect data
3.pass the collected data
4.pass the category information
5.defining target feature types to be extracted
6.normalisation the collected data and pass it
7.apply the live data to corresponding model
8.classify traffic data into target classes
9.translate the classification result and pass it to outer network systems
10.deploy policy
11.pass the model update information
12.decide to update model
13.update model
14.repeat 5-10

Figure 5-16: Scenario of AI enabled network traffic classification

1. Training the traffic classification model with historical data.

2. The OSS collects unknown traffic data from network infrastructure.

3. The collected data is sent to the ENI system.

4. Context-Aware Management Functional Block transfers traffic category information to Cognition Management Functional Block.

5. Cognition Management Functional Block defines feature extraction rules and sends them to Data Ingestion and Normalization Functional Block.

6. Data Ingestion and Normalization Functional Block processes the raw data and then transfers the processed data to Cognition Management Functional Block.

7. Cognition Management Functional Block understands and transfers the normalized data to Knowledge Management Functional Block.

8. Knowledge Management Functional Block feeds the information to the pre-trained model and performs inferences.

9. Denormalization and Output Generation Functional Block gets the traffic classification results and converts them to a form that OSS can understand, and then transfers it to OSS.

10. OSS adjusts the configuration of network infrastructure.

11. Context-Aware Management Functional Block transfers model updating information to Cognition Management Functional Block.

12. Cognition Management Functional Block informs Offline Training Functional Block to retrain and update model.

13. Offline Training Functional Block provides Knowledge Management Functional Block with the updated model and then Knowledge Management Functional Block replaces the saved model.

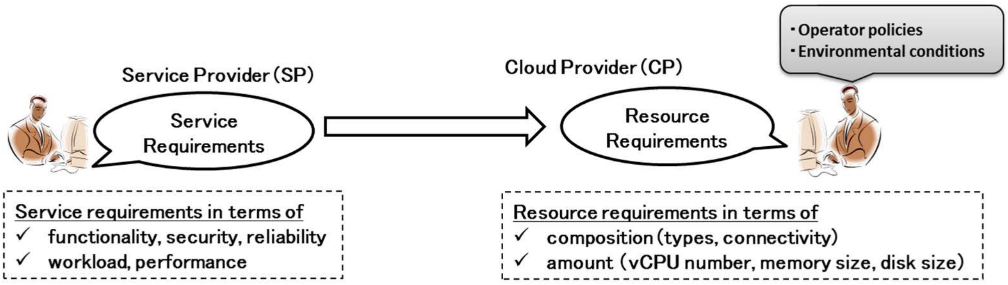
14. Repeat step 5-10.

5.3.8 Use Case #2-8: Automatic service and resource design framework for cloud service

5.3.8.1 Use case context

Cloud service based on Virtualization technology enables prompt on-demand realization of various functions on the virtualized platform. An increasing number of Service Providers (SP) such as ecommerce companies, science institutes, are implementing various kinds of services and functions, e.g. web service, machine learning in the cloud environment provided by the Cloud Provider (CP). As shown in Figure 5-17 the SP is concerned about the service requirements, such as the functionality of the service, the levels of security and reliability, and the ability to handle workloads. In contrast, CP needs to know the composition of resources and amount of resources to be allocated when fulfilling the service orders. The resource composition describes the types of resources and the connectivity between them. For instance, resource composition for a basic web service is web server instance connected to a database instance. Resource amount is the amount of vCPU and memory, disk to allocate to each instances aforementioned. Therefore, in accordance with the service requirements, cloud resource composition and amount need to be designed in various phases of cloud service delivery. Currently the design heavily relies on the human decision-making process either on the SP side or the CP side. Practices to design the cloud resource include:

- Self-service approach. The SP is provided with a management interface to manage cloud resources and needs to decide their resource requirements. The approach requires the SP to have IT expertise and may put up barriers to SPs wishing to enter the market.
- Cloud-consultant approach. The SP is assisted by cloud consultants from the CSP who collect the service requirements from SP and decide resource details accordingly. The approach demands the CSP spend high operating expenditure (OPEX) and leads to them needing a relatively long time to deliver services.



・Operator policies
・Environmental conditions
Service Provider (SP)
Cloud Provider (CP)
Service Requirements
Resource Requirements
Service requirements in terms of
✓ functionality, security, reliability
✓ workload, performance
Resource requirements in terms of
✓ composition (types, connectivity)
✓ amount (vCPU number, memory size, disk size)

Figure 5-17: Design cloud resources in accordance with service requirements

5.3.8.2 Description of the use case

5.3.8.2.1 Overview

As addressed above, the current approaches to design the cloud resources lead to high human cost on the SP or CP side. This use case aims to reduce the cost by automating the design process leveraging the previous human design/operation knowledge. The automatic cloud resource design framework is expected to be able to design cloud resource in accordance with service requirements in the following 3 aspects as shown in Figure 5-18:

- Service requirement analysis: SPs describe the Service Requirements through various channels, e.g. natural language and GUI. The system needs a requirements analysis entity that is responsible for analysing Service Requirements captured in various channels e.g. natural language or APIs, and categorizing the requirements into atom requirements, i.e. requirements regarding functionality, security, reliability, performance, etc. In the case of service requirements in natural language, the analysis function is realized based on the natural language processing model, which is generated from machine learning of human analysis results data set.
- Resource composition design: This entity takes the result of the requirements analysis, and is able to customize the resource composition automatically in accordance with the atom service requirements using customize rules. These customize rules define the actions to be taken, i.e. how to derive resource composition given different functionality, security and reliability requirements. The rules to customize the service composition may be statically generated by the operator or dynamically based on the machine learning of previous composition design examples.

- Resource amount design: This entity decides the resource amount needed to satisfy performance requirements. Besides the workload and the performance requirements, environmental conditions and operator policies need to be considered to decide the amount of computation resources allocated to e.g. virtual machines (VMs). In this context, environmental conditions may include the static conditions, e.g. central processing unit (CPU) clocks and memory architecture of the host, and dynamic conditions, e.g. resource utilization ratio of the physical host to which the VM is allocated. In addition, operator policies are statically set by the SP or CP and restrict the operation state within a desired range, e.g. an operator policy restricts the computation resource usage of a VM to 50 % - 90 % to prevent resource underuse or overuse. The model used to calculate the resource amount is generated from the log data including the resource amount configuration data, performance data, environmental data and operational state data. These data can be obtained by utilizing the ENI system external RPs including or-eni-cfg, app-eni-ctx, inf-eni-dat.

The resource design result includes the resource composition and the resource amount. This information is fed back to the SP engineer and to the CP operator for evaluating the design result. On the other hand, the resource management and orchestration system is enforced with the design result to implement the service requested at the upper level. Finally, the Infrastructure system, which is a NFVI-like System composed by APPs, VMs and overall HW, is also enforced through the, inf-eni-dat RPs with the design result to implement the service requested at the lower level. On its turn, this NFVI-like System feeds back the Resource Design System, though the inf-eni-dat RP, with raw data for evaluation as part of the monitoring process.

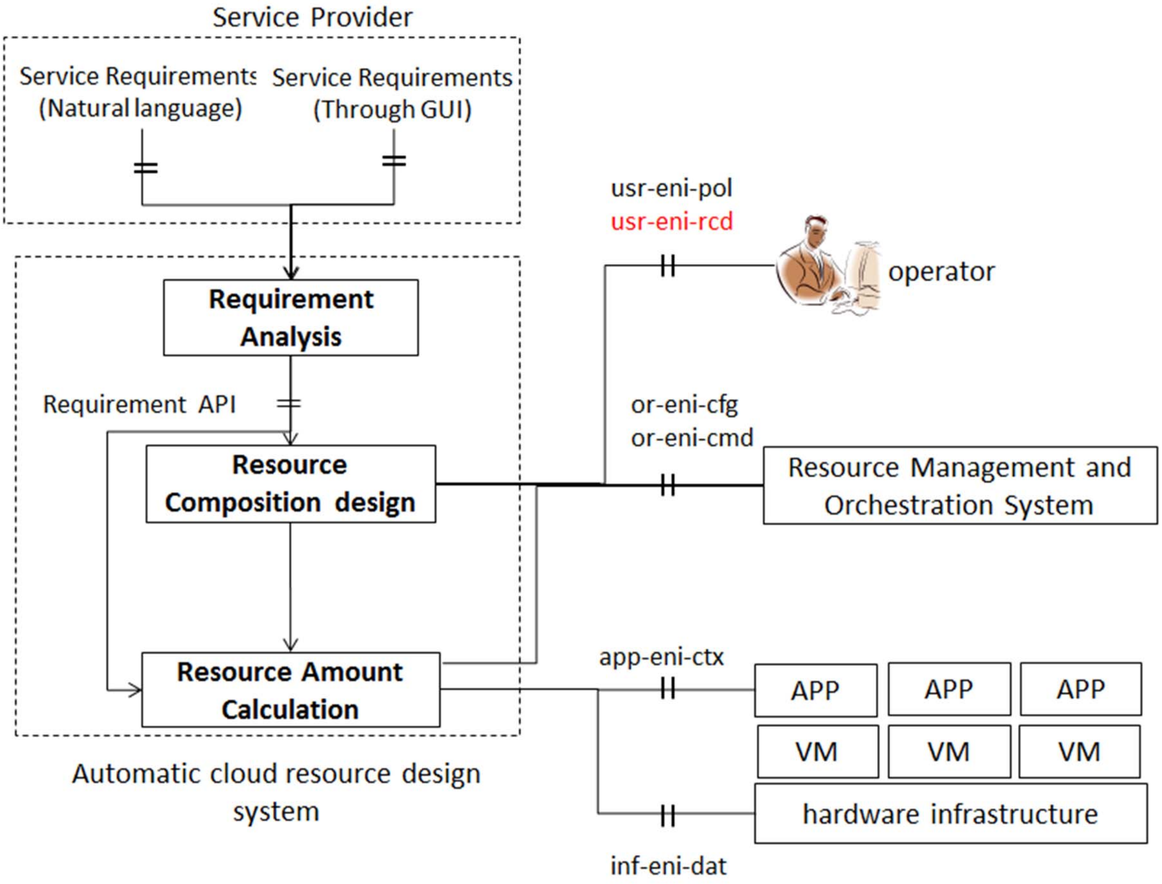

Service Provider
Service Requirements (Natural language)
Service Requirements (Through GUI)
usr-eni-pol
usr-eni-rcd
operator
Requirement Analysis
Requirement API
or-eni-cfg
or-eni-cmd
Resource Composition design
Resource Management and Orchestration System
Resource Amount Calculation
app-eni-ctx
APP
APP
APP
VM
VM
VM
hardware infrastructure
Automatic cloud resource design system
inf-eni-dat

Figure 5-18: Automatic Cloud Resource Design System

5.3.8.2.2 Motivation

Automatic design of cloud resource benefits in the consultation, design, and operation phases of cloud service delivery. For example, in the consultation phase, for an SP who plans to migrate services implemented in an on-premises environment into a cloud environment, it can be used to show the performance that can be achieved after the migration and the needed cloud resources and cost. In the design phase, it enables the automatic design of resources, thus service design time and human labour reduction can be expected. In the operation phase, it is able to adjust the resource composition and amount in accordance to the changes, thus ensuring the continuous satisfaction of service requirements, which contributes to higher customer satisfaction.

5.3.8.2.3 Actors and Roles

SP: it requests a cloud service to the design system, and use the cloud service to implement specific functions and services. Therefore, it is more concerned on the service requirements (functionality, security, performance, etc.).

Engineer of SP: it is the Assisted System (party) of Automatic Cloud Resource Design System. The design result produced by the Automatic Cloud Resource Design System is fed back to it, and it is able to accept, decline or edit the design result and instruct the Resource management and orchestration Assisted System to orchestrate the resource based on the design results. It is able to add/edit/delete the operator policies used in the system directly through usr-eni-pol RP.

CP: it provides cloud service to the SP. To implement the cloud service, it needs the information about the composition of cloud resources, and the amount of computation resources (vCPU, memory, and disk).

Operator of CP: it is the Assisted System (party) of Automatic Cloud Resource Design System. The design result produced by the Automatic Cloud Resource Design System is fed back to it, and it is able to accept, decline or edit the design result and instruct the Resource management and orchestration Assisted System to orchestrate the resource based on the design results. It is able to add/edit/delete the operator policies used in the system directly through usr-eni-pol RP.

Automatic Cloud Resource Design System: it consists of three main entities i.e. service requirements analysis, resource composition design, and resource amount design, as defined in the section on overview of the Use Case.

Resource management and orchestration system: it is the Assisted System of Automatic Cloud Resource Design System. It is responsible for the orchestration, configuration, and activation and monitoring of the cloud resources.

NFVI-like infrastructure: it is the Assisted System composed by APPs, VMs, and other hardware usually associated with a virtualized network infrastructure.

5.3.8.2.4 Initial context configuration

The automatic cloud resource design system exposes interfaces for the SP and the CSP operator from the three function blocks. The SP or CSP can use the system to do the requirement analysis, resource composition design and resource amount design.

5.3.8.2.5 Triggering conditions

- Consultation: The SP engineer uses the Automatic Cloud Resource Design System to estimate the needed resource before requesting the cloud service.

- Design: The SP places a service request and, to meet the service requirements, the Automatic Cloud Resource Design System is utilized to decide the resource needed.

- Operation: After the service delivery, changes in workload, environment etc., may be detected that may lead to service requirements violation. Based on the monitored data that is fed back from the NFVI-like infrastructure System, the Automatic Cloud Resource Design System is able to redesign resource composition and the resource amount to ensure the service requirements are satisfied again.

5.3.8.2.6 Operational flow of actions

1) The models used respectively in the three function blocks in the automatic cloud resource design system are trained based on previous design experiences and operation log data as mentioned before.

2) The service requirement analysis function parses the service requirements into atom standard requirements.

3) The resource composition design function customizes the resource composition according to the atom requirements.

4) The resource amount calculation function decides the resource amount in accordance with the workload, performance requirement based on current environmental conditions and operation policies.

5) The operator optimizes the design result if the service requirements are not satisfied.

6) The system models are retained based on the new human design results and monitoring data.

At the beginning, action 1 is taken to prepare the automatic cloud resource design system.

In the consultancy phase, the automatic cloud resource design system works in recommendation mode. The SP inputs the service requirements in any available formats. Actions 2-4 are taken and the output of the automatic cloud resource design system is feedback to the SP. The SP may also input partial requirements, in this case, the system takes one or multiple actions in action 2-4 and feedback the results to the SP.

In the design phase, the automatic cloud resource design system works in recommendation or management mode. Actions 2-4 are taken. The results are passed to the CSP operator and resource management and orchestration system. If the requirements are not met, action 5 is taken.

In the operation phase, the automatic cloud resource design system works in recommendation or management mode. The system collects the monitoring data of service, and if there are changes in workload, environmental conditions, and operation policies, the automatic cloud resource design system is used to adjust the resource.

The system takes action 6 to update the system model automatically or under the instruction.

5.3.8.2.7 Post-conditions

In various phases of cloud service delivery, precise and seamless resource design in accordance with service requirements in a much shorter time compared to human design. The human cost to design the resource is reduced to a large extent. Meanwhile, the accuracy of resource decision is improved by the continuous data collection and training of models.

5.3.9 Use Case #2-9: Intelligent time synchronization of network

5.3.9.1 Use case context

With wide application of information technology and Internet, many industries and fields are rapidly putting the business on the internet. For example, automation and networking have been realized in aviation, finance, railway transportation, medical and other systems, so there shall be a coordinated and unified time to ensure these systems can work together. At the same time, data traceability and analysis, data interaction of information system such as Internet of things, cloud computing and big data are all based on accurate time. Therefore, it is more and more important to get a unified standard time.

High-precision time synchronization is one of the key requirements of 5G network. The application scenarios of time synchronization is very extensive. In some scenarios, time synchronization is needed on a large scale while the accuracy of time synchronization is not highly requested. However, the time synchronization is strictly required which should reach nanosecond in some application scenarios, such as military command system, financial transaction system. Therefore, it is significant to improve the accuracy of time synchronization.

For these reasons, an intelligent time synchronization of network scheme is proposed, which can effectively improve accuracy of time synchronization. The prediction model can accurately predict time offset and time skew rate by using AI technology. According to predict results, the system can adjust its clock time to reduce the deviation from the standard time.

5.3.9.2 Description of the use case

5.3.9.2.1 Overview

In this use case, intelligent time synchronization of network is based on AI-based prediction module which can effectively improve accuracy of time synchronization. This intelligent time synchronization can be used in many scenarios such as fault location, transport intelligent dispatching, financial transaction system and public security management. The intelligent system is divided to three modules:

- Data acquisition module that used to obtain observation data (e.g. clock time, time offset, frequency offset, time skew rate, and channel environment) and accuracy requirement according to predefined rules.

- Control module that used to decide which machine learning algorithm to invoke according to accuracy requirement.

- AI-based prediction module that used to predict the time offset and time skew rate.

In the model training phase, the prediction models are trained on the basis of appropriate machine learning and observation data. Then the trained prediction model estimates the time offset and time skew rate. In the inference phase, AI-based prediction module implemented in the ENI system. The ENI system provides the predicted results to the external system. The external system dynamically adjusts the local clock time to reduce the deviation from the standard time, and achieves accurate time synchronization. In addition, the parameters of model can be dynamically adjusted according to the prediction results.

5.3.9.2.2 Motivation

Time synchronization requires that the time offset between system time and standard time be limited to a small range. There is a deviation when standard time is distributed to the subordinate node by using wired or wireless network, namely time delay which including signal processing delay, signal transmission delay, medium access delay, noise interference delay, etc. Therefore, the system time is not synchronized with standard time, or time synchronization accuracy is not ideal.

Time synchronization can be achieved by various methods, such as time synchronization based on the Precision Time Protocol (PTP). However, with the increasing demand on time synchronization, the traditional methods cannot meet the accuracy requirements of time synchronization in some scenarios (e.g. traffic management system, fault location, and financial transaction system).

Time synchronization based on PTP: master clock and slave clock add time stamps at the network link layer in order to accurately record the time when the message is received or sent. The master clock and slave clock obtain the timestamps by exchanging message through four times, thereby achieving calculation of the time offset. However, the time offset is inaccurate for PTP ignores the asymmetry of channel. In addition, PTP requires hardware support which costs a lot.

In this use case, an AI-based time synchronization is proposed in the ENI system. Appropriate algorithms in machine learning is introduced to estimate the time offset and time skew rate according to different accuracy demands on time synchronization, thus accuracy of time synchronization is improved.

The intelligent time synchronization of network is illustrated in Figure 5-19.

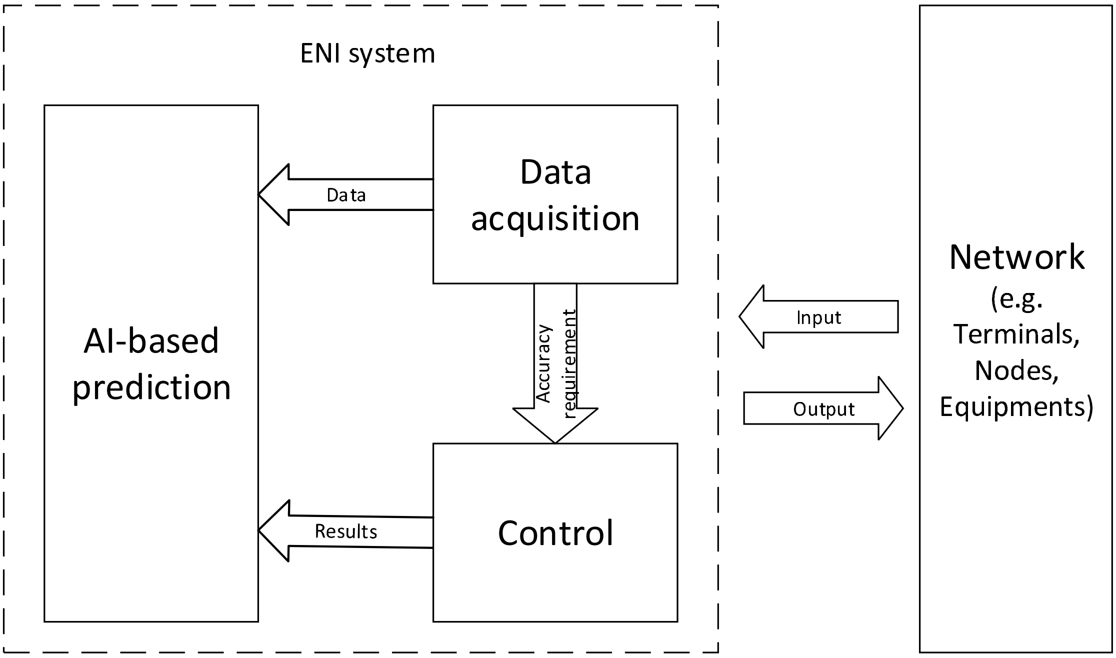


Figure 5-19: Intelligent time synchronization of network

5.3.9.2.3 Actors and Roles

- ENI System: system solution used to receive observation data from network; to predict the time offset and time skew rate according to the observation data; to provide results to outer network systems. (Input: observation data (e.g. clock time, time offset, frequency offset, time skew rate, and channel environment) and accuracy requirement. Output: the prediction results).

- Network: entity that provide the observation data and accuracy requirement.

5.3.9.2.4 Initial context configuration

- Define the accuracy level of time synchronization.
- Observation data obtains on the basis of predefined rules, and then match the accuracy requirement with the accuracy level.
- Time synchronization in ENI system is enabled with AI capability, which is initialled to learn target prediction through observation data.

5.3.9.2.5 Triggering conditions

In this use case, when the predicted rules cannot hit the accuracy of time synchronization or with the approaching predicted period, the process of predicting the time offset and time skew rate will be initiated. In the meantime, the parameters of the prediction model will be dynamically adjusted according to the prediction results.

5.3.9.2.6 Operational flow of actions

In the ENI system, AI-based time synchronization is intended to improve the accuracy of time synchronization by introducing the new AI capability. The time offset and time skew rate can be predicted by using machine learning method, with the following flow of activities:

1) The ENI system initiates a process to obtain observation data (e.g. clock time, time offset, frequency offset, time skew rate, and channel environment) by using predefined rules.

2) Determine which machine learning algorithm to call according to the accuracy level.

3) The ENI system initiates a process to train AI prediction model on basis of observation data.

4) The trained AI prediction model estimates the time offset and the time skew rate.

5) The ENI system provides the predicted results to the external system.

6) In order to reduce the deviation between local time and standard time, the external system adjusts its clock time based on the received results and predefined rules.

5.3.9.2.7 Post-conditions

- The external system adjust its clock time on the basis of the prediction results provided by the ENI system. In this way, the deviation between the system time and the standard time is reduced, and the accuracy of time synchronization is improved.
- The prediction model dynamically adjusts the parameters of the model based on the prediction results.
- At the next prediction period, the prediction model will predict the time offset and time skew rate for the next state.
- The ENI system dynamically adjusts the period of time synchronization based on the feedback from the external system.

5.4 Service Orchestration and Management

5.4.1 Use Case #3-1: Context-aware VoLTE Service Experience Optimization

5.4.1.1 Use case context

As the mobile network evolves to 4G and 5G, an all-IP network will provide high definition voice transmission. VoLTE, namely Voice over LTE, it is an IP data transmission technology, which does not need a 2G or 3G network. In VoLTE, all business bears on 4G network, and can realize the unification of data and voice services using the same network. As a result, the 4G and 5G networks not only provide high-speed data services, but also provide high-quality audio and video services, the latter achieved via the use of the VoLTE technology.

5.4.1.2 Description of the use case

5.4.1.2.1 Overview

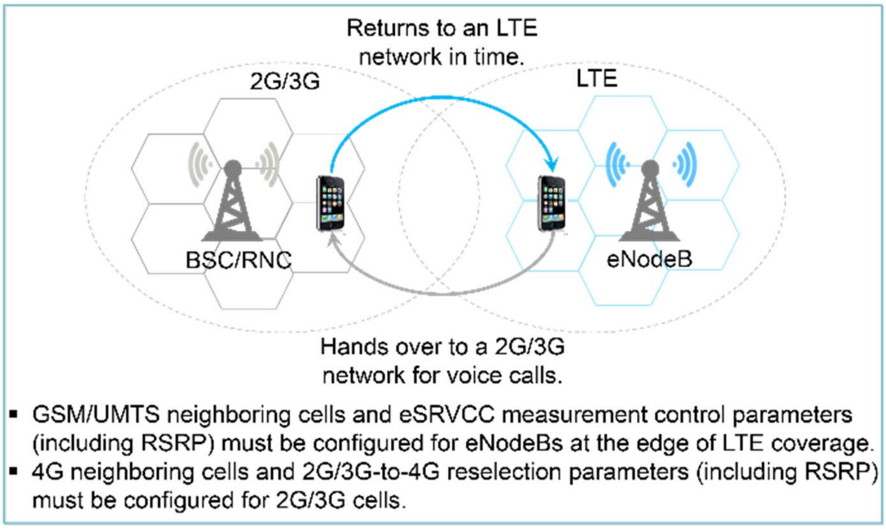
Conventionally, operators rely on field or drive tests to determine the Reference Signal Received Power (RSRP) for smooth VoLTE service experience. However, such tests are not adequate and efficient enough to support increased quality and capacity demands, because it is difficult for a human expert to do thorough tests everywhere and every day, and such tests are also error prone. Moreover, VoLTE RSRP is configured in a RAN statically, which consequently results in VoLTE call drop or handover to 2/3G unnecessarily.

VoLTE operation requires the RSRP to be adaptively configured to meet the changing context.

5.4.1.2.2 Motivation

It has been observed that the RSRP configuration is relevant to many factors, such as mobile terminal type, user location, voice codec, traffic load, time of day, etc. These factors may change frequently, which makes it very difficult to find a deterministic function to model this dynamism. Therefore, an intelligent entity (i.e. the ENI system) can be used to collect the relevant data, use one or more AI mechanisms to analyse the data, and then predict the proper RSRP. When an ENI system sends the predicted RSRP to operations system, the VoLTE RSRP will be adjusted according to the different VoLTE service information. Furthermore, RAN monitors the fulfilment of the QoS requirements. If the QoS requirements are no longer fulfilled, a notification is sent to the ENI System (and/or OSS), which will take actions to either adjust to lower QoS requirements or to terminate the service. With this control loop enabled by ENI system, the VoLTE service experience can be optimized adaptively and responsively in contrast to time-consuming and inefficient manual field tests.

A figure illustrating the Interoperability between VoLTE and 2G and 3G is given in Figure 5-20.


Returns to an LTE network in time.
2G/3G
LTE
BSC/RNC
eNodeB
Hands over to a 2G/3G network for voice calls.

- GSM/UMTS neighboring cells and eSRVCC measurement control parameters (including RSRP) must be configured for eNodeBs at the edge of LTE coverage.
- 4G neighboring cells and 2G/3G-to-4G reselection parameters (including RSRP) must be configured for 2G/3G cells.

Figure 5-20: Interoperability between VoLTE and 2G and 3G

5.4.1.2.3 Actors and Roles

- Radio Access Network: monitors whether the QoS requirements are met and notifies the ENI system.
- ENI Engine: collects and analyses VoLTE service information and contextual data, and dynamically determines if the RSRP should be reconfigured.
- Operations System: as defined by ETSI TS 132 101 [i.5], adjusts VoLTE RSRP according to the policies generated by the ENI system.
- Network Administrator: responsible for configuring the network.

5.4.1.2.4 Initial context configuration

- The VoLTE RSRP was configured in RAN statically.
- The ENI system has learned how to configure the RSRP in order to ensure the VoLTE service experience.

5.4.1.2.5 Triggering conditions

The current VoLTE RSRP does not meet the VoLTE continuity coverage requirement.

5.4.1.2.6 Operational flow of actions

1) ENI system collects and analyses VoLTE service information (and any other necessary information, such as contextual data).

2) ENI system determines what the RSRP should be according to the current VoLTE service information.

3) Operations system reconfigures the RSRP according to the output of the ENI system.

4) RAN monitors whether the QoS requirements are met; if not, it notifies the ENI system.

5) ENI system recommends appropriate changes (e.g. rollback the RSRP, or make other configuration changes) to meet the QoS requirements.

6) Operations system implements recommended changes.

5.4.1.2.7 Post-conditions

- The VoLTE RSRP dynamically adjusts according to the changing network environment.
- The VoLTE service experience was optimized adaptively.

5.4.1.3 Mapping to ENI reference architecture

5.4.1.3.1 Functional blocks

The functional blocks for context-aware VoLTE Service experience using AI is shown in Figure 5-21.

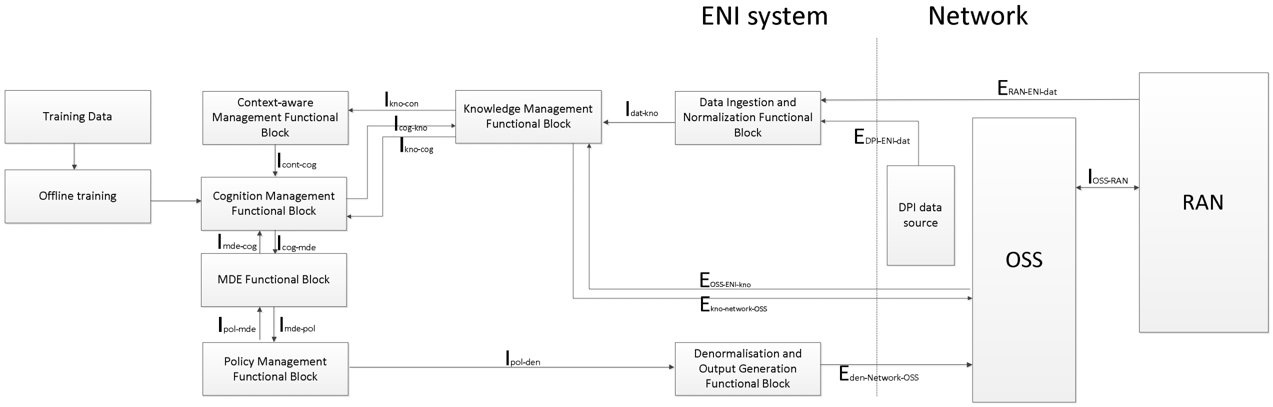


Figure 5-21: Mapping to ENI reference architecture for Context-aware VoLTE Service Experience

For training data, it can be pre-prepared one or the one from RAN and DPI data Source.

The "Data ingestion and Normalization Functional block" collects DPI data from DPI data source (e.g. core network or data centre) and radio parameters from RAN. These data are filtered and normalized accordingly.

The "Knowledge Management Functional Block" is used to store knowledges and generate inferences. There may be some interactions between "Knowledge Management Functional Block" and OSS on feature set update which will be used for modelling iteration in "Cognition Management Functional Block".

The "Context-aware Management Functional Block" is used to generate context information of users.

The "Cognition Management Functional Block" utilizes data as well as inferences to generate predictions based on modelling provided by offline training. For the modelling in "Cognition Management Functional block", it will be iterated periodically based on the data from DPI Data Source and RAN.

The "Policy Management Functional Block" makes the polices and inputs decisions to "Denormalization and Output Generation Functional Block", which generates execution command to operating system like OSS. Next, OSS would send the command to RAN for adjusting radio parameters like transmission power and so on.

5.4.1.3.2 Reference Points

$E_{RAN\text{-}ENI\text{-}dat}$ defines data exchange between RAN and ENI system.

$E_{Data\text{-}DPI}$ defines data exchange between DPI data source (e.g. certain entity in core network, data centre) and ENI system.

$I_{dat\text{-}kno}$ defines internal Reference Point between "Data ingestion and Normalization Functional block" and "Knowledge Management Functional Block".

$I_{kno\text{-}con}$ defines internal Reference Point between "Knowledge Management Functional Block" and "Context-aware Management Functional Block".

$I_{cont\text{-}cog}$ defines internal Reference Point between "Context-aware Management Functional Block" and "Cognition Management Functional Block".

$I_{cog\text{-}kno}$ and $I_{kno\text{-}cog}$ define internal Reference Point between "Cognition Management Functional Block" and "Knowledge Management Functional Block".

$I_{cog\text{-}mde}$ and $I_{mde\text{-}cog}$ define internal Reference Point between "MDE Functional Block" and "Cognition Management Functional Block".

$I_{pol\text{-}mde}$ and $I_{mde\text{-}pol}$ define internal Reference Point between "MDE Functional Block" and "Policy Management Functional Block".

$I_{pol\text{-}den}$ defines internal Reference Point between "Policy Management Functional Block" and "Denormalization and Output Generation Functional Block".

$E_{\text{den-network-OSS}}$ defines data exchange between "Denormalization and Output Functional Generation Block" and "OSS".

$E_{\text{kno-network-oss}}$ and $E_{\text{oss-ENI-kno}}$ define data exchange between "Knowledge Management Functional Block" and OSS.

5.4.1.3.3 Flow of information

The flow of information is shown in Figure 5-22.

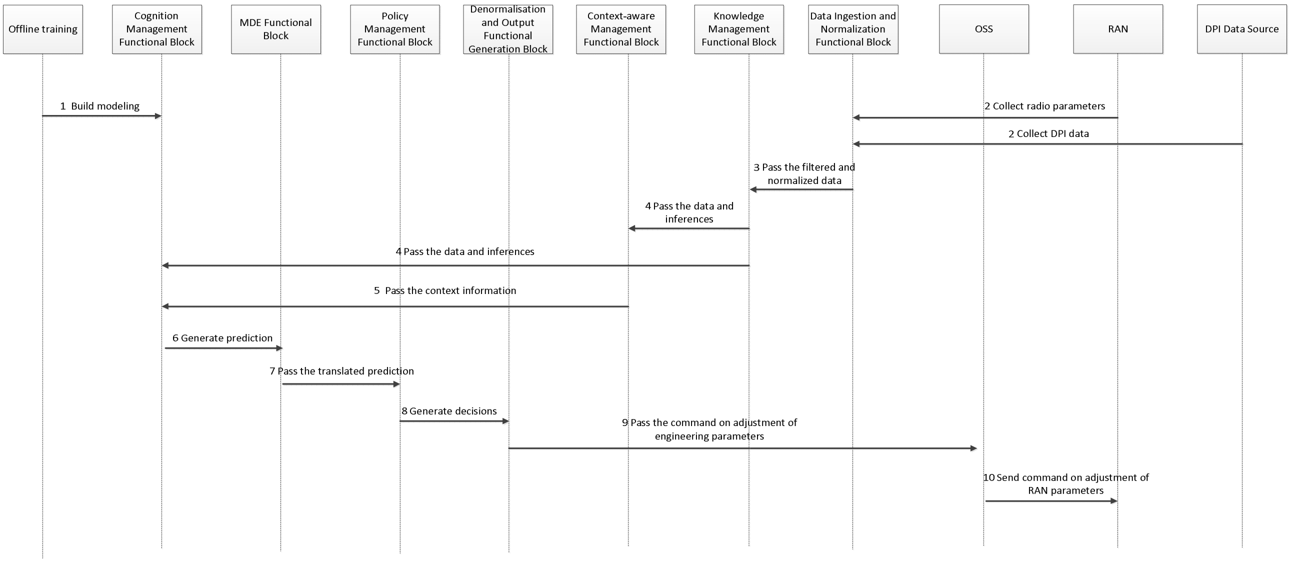

Offline training
Cognition Management Functional Block
MDE Functional Block
Policy Management Functional Block
Denormalisation and Output Functional Generation Block
Context-aware Management Functional Block
Knowledge Management Functional Block
Data Ingestion and Normalization Functional Block
OSS
RAN
DPI Data Source
1 Build modeling
2 Collect radio parameters
2 Collect DPI data
3 Pass the filtered and normalized data
4 Pass the data and inferences
4 Pass the data and inferences
5 Pass the context information
6 Generate prediction
7 Pass the translated prediction
8 Generate decisions
9 Pass the command on adjustment of engineering parameters
10 Send command on adjustment of RAN parameters

Figure 5-22: Flow of information for Context-aware VoLTE Service Experience

Step 1 is about building modelling via offline training.

Step 2 is that "Data ingestion and Normalization Functional block" collects DPI data from DPI data source and radio parameters from RAN.

Step 3 is that the filtered and normalized data is passed to "Knowledge Management Functional Block".

In Step 4, the data and inferences are passed to "Context-aware Management Functional Block" and "Cognition Management Functional Block".

In step 5, the context information generated from "Context-aware Management Functional Block" is passed to "Cognition Management Functional Block".

In step 6, the prediction is generated in "Cognition Management Functional Block" and passed to "MDE Functional Block".

In step 7, the translated predictions are generated and passed to "Policy Management Functional Block".

In step 8, "Policy Management Functional Block" generates decisions to "Denormalization Functional" and "Output Generation Block".

In step 9, command on adjustment of engineering parameters is passed to operating system like OSS.

In step 10, command on adjustment of RAN parameters like transmission power is passed to RAN.

There would be some steps additionally on model iteration between "Knowledge Management Functional Block" and "Cognition Management Functional Block".

5.4.2 Use Case #3-2: Intelligent network slicing management

5.4.2.1 Use case context

The concept of network slicing has been introduced by the NGMN 5G whitepaper [i.2], which enables multiple logical self-contained networks to use a common physical infrastructure platform, enabling a flexible stakeholder ecosystem that allows technical and business innovation integrating network and cloud resources into a programmable, software-oriented network environment. From the perspective of 3GPP [i.3], network slicing enables operators to create networks customized to provide optimized solutions for different market scenarios which demands diverse requirements, e.g. in the areas of functionality, performance and isolation.

In particular, network slicing can be used to support very diverse requirements imposed by different type of services, e.g. IoT services as well as 5G services as eMBB/mMTC/uRLLC in a single physical network (universal and heterogeneous) by using flexibility and scalability.

5.4.2.2 Description of the use case

5.4.2.2.1 Overview

The realization of the network slice concept is accomplished using network slice instances (NSIs), see Figure 5-23. An NSI is an instance of a logical representation of Network Function(s) and corresponding resource requirements necessary to provide the required end to end (E2E) telecommunication services and network capabilities. An NSI typically covers multiple technical domains, which includes terminal, access network, transport network and core network, as well as DC domain that can host third-party applications from vertical industries.

In the early stage of network slicing deployment, there could be only a few NSIs. The deployment may occur in a semi-automatic mode. As the number of NSIs increases and scenarios, such as, dynamic instantiation of NSIs or runtime adaptation of the deployed NSI emerge, more advanced technologies will be desired to support network slicing and its further evolution, see e.g. [i.3]. Specifically, management functions could become real-time, implying that the difference between management and control will gradually disappear. Some management functions will be tightly integrated with the NSIs as well as the network infrastructure.

This use case is applying the ENI system to enhance and optimize the network slice management and control operations.

Other possible scenarios are network slicing where an operator can dynamically change a given slice resource reservation, considering that each slice may be assigned for a specific type of service or service class. Moreover, hybrid scenarios, where network slicing and resource sharing are applied at the same time are also envisaged. These scenarios are not addressed in this release.

5.4.2.2.2 Motivation

In current networks, technical domains are normally coordinated via centralized network management system. In 5G, performing real-time cross-domain coordination through distributed lower layer such as control plane would be possible, with potentially unified control logic of different domains.

Advanced automation and AI algorithms can be applied in a unified, "holistic" network manner, which could be scalable and flexible, and which might then achieve runtime deployment and adaptation of NSIs.

In the context of ENI, the ENI system can be used to enhance and optimize the network slice management and control operations.

Figure 5-23 shows an example of two network slice instances that are being created using the 3GPP 5G infrastructure. Please note that Figure 5-23 shows functions used in Network Slicing for the interaction purposes of ENI. In a production environment other capabilities are available.

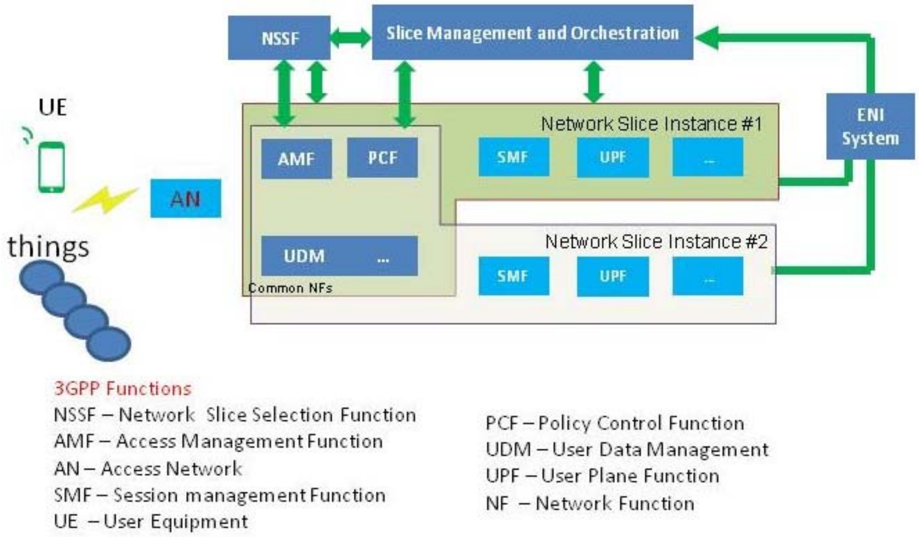

NSSF
Slice Management and Orchestration
UE
ENI System
Network Slice Instance #1
AMF
PCF
SMF
UPF
...
AN
things
Network Slice Instance #2
UDM
...
Common NFs
SMF
UPF
...
3GPP Functions
NSSF – Network Slice Selection Function
AMF – Access Management Function
AN – Access Network
SMF – Session management Function
UE – User Equipment
PCF – Policy Control Function
UDM – User Data Management
UPF – User Plane Function
NF – Network Function

Figure 5-23: Example of a network slicing management and orchestration scenario

5.4.2.2.3 Actors and Roles

- Slice Management and Orchestration: entity that manages and orchestrates the life cycle of slices; Note that this entity is administrated by a network operator.
- Network Infrastructure: infrastructure used to create and maintain the slice.
- ENI system: system solution used to assist and optimize the operation of the Slice Management and Orchestration entity.
- IoT devices: devices that can be represented as Things and can take part in the operation of the end to end slice.
- UE: any device, e.g. Smart phone that is using the 3GPP cellular technology and can take part in the operation of the end to end slice.

5.4.2.2.4 Initial context configuration

The slices are created and configured; The ENI system through AI and machine learning capabilities is learning the configuration of the applied slices and as well the traffic patterns used by each of these slices; moreover, the ENI system measures the utilization of the network and other relevant parameters that define the satisfactorily operation of each slice and that needs to conform to the Service and Network KPIs requested by the operators.

5.4.2.2.5 Triggering conditions

Triggering conditions that may affect resources in transport network plane include but not limited to, the following situations, e.g.: input traffic adjustment, network health deterioration, routing path switching, bandwidth overutilization, packet loss rate increase, latency increate.

When associated with assisted networks that use network slicing, the ENI system should actively and passively monitor related triggering conditions for anomaly detection (fault, error and unusual behaviour), prevention and fast recovery. Therefore, the ENI system should be distributed in the whole life-cycle management of transport network slicing, including probes deployment, status collection and analysis, decision making, verification and delivery.

When the ENI system concludes that the measured parameters associated with the operation of each slice do not conform to the Service and Network KPIs requested by the operators, then the ENI system notifies the Slice Management and Orchestration entity about this event.

5.4.2.2.6 Operational flow of actions

ENI system applying analysis and machine learning technologies can be used to enhance and optimize the network slice management and control operations and to assist the Slice Management and Orchestration entity to resolve any abnormal operation of each slice; some of the ENI system activities are listed below:

1) ENI system analyses the collected data associated to e.g. network topology, network traffic load, service characteristic, user location and movement, VNF type and placement constraints, infrastructure capability and resource usage, etc.

2) While performing the monitor process, the ENI system is aware of the existence of triggering conditions for e.g. network health, bandwidth utilization, loss rate, latency, and performs anomaly (fault, error and unusual behaviour) demarcation.

3) ENI system produces a proper context aware policy to indicate to the network slice management entity when, where and how to place or adjust the network slice instance (e.g. reconfiguration, scale-in, scale-out, change the template of the network slice instance), including the network slice functions and their configurations, in order to achieve an optimized resource utilization according to the possible change of service requirements and/or the network environment.

4) ENI system provides possible root causes analysis report.

5.4.2.2.7 Post-conditions

The abnormal operation of the slice is resolved and the slice performs and conforms according to the Service and Network KPIs requested by Operators.

5.4.3 Use Case #3-3: Intelligent carrier-managed SD-WAN

5.4.3.1 Use case context

Software-Defined Wide Area Network (SD-WAN) is an approach of designing and deploying an enterprise WAN that uses SDN to determine the most effective way to route traffic to remote locations. SD-WAN allows enterprises to reduce the cost of expensive leased Multi-Protocol Label Switching (MPLS) circuits by sending lower priority, less-sensitive data over cheaper public Internet connections, as well as by reserving private links for mission-critical or latency-sensitive traffic like VoIP.

With the carrier managed SD-WAN service, enterprises can free up from network management and monitoring, and focus more on the business itself.

5.4.3.2 Description of the use case

5.4.3.2.1 Overview

The enterprise has a hybrid Wireless Access Network (WAN), which includes the high quality private MPLS circuit, as well as economic public Internet access and wireless access for last resort. The devices at the edge of the enterprise network are managed by a network supervisory controller. Furthermore:

- Each enterprise can have customized SD-WAN services through the web portal via an Application Programming Interface (API). Enterprises customize their own networked experience based on their business needs, such as cost priority, quality priority, or cost-effective priority. Similarly, enterprises may customize their communication service levels assurance requirements based on the needs demanded by their enterprise communications applications. Therefore, the network supervisory controller may adapt intelligent policies according to the preceding enterprise needs.

- Different WAN traffic will be handled by the intelligent WAN policies, in order to make the usage of bandwidth resource more efficient. These policies steer traffic depending on WAN resource capabilities and according to application performance demands. This dynamic process saves time and optimizes resources.

- As conditions on the network change, the network supervisory controller may detect them through monitoring mechanisms, and adapt traffic routing through intelligent WAN policies, which automatically prioritize critical applications while dynamically suppressing non-critical ones.

As the logic of switching across different circuits becomes more complex, network intelligence can help Network Administrators to better manage the SD-WAN service.

The current Use Case is further described by the following set of components and features.

5.4.3.2.2 Motivation

The main advantage of using the ENI System is that, through the use of AI and context-awareness, it can monitor the network and help enterprises to optimize their services and resources, hence allowing enterprises to focus more on their businesses.

A further advantage of applying the ENI System to SD-WAN services is that it can expose an Intent based interface that allows enterprises to customize their service using natural language with a terminology that is familiar to them.

EXAMPLES: Such Intent policies in SD-WAN could be:

- "All personal devices will access Internet using the Overlay connection";
- "All traffic belonging to Users in Administrative group will go through the firewall";
- "All policies applied to personal devices will also apply to guest devices";
- "John is part of the Administrative group";
- "Personal devices cannot access Facebook more than one hour per day".

Additionally, the ENI System may also use AI methods in order to optimize the service and suggest policies adaptations to Network Administrators, e.g.:

- Scenario 1: Guest devices have been using a large part of the bandwidth with video streaming. The ENI System may trigger an alarm identifying the need to adapt an existing or create a new one to lower the priority for guest devices.
- Scenario 2: Every last day of the month, company A backups all data to a server in the central office. The ENI System could suggest a periodic rule so that all traffic coming from registered devices and destined to the backup server bypasses the firewall (preventing unnecessary use of resources and speeding up the backup process).

Figure 5-24 provides a pictorial representation of the use case described.

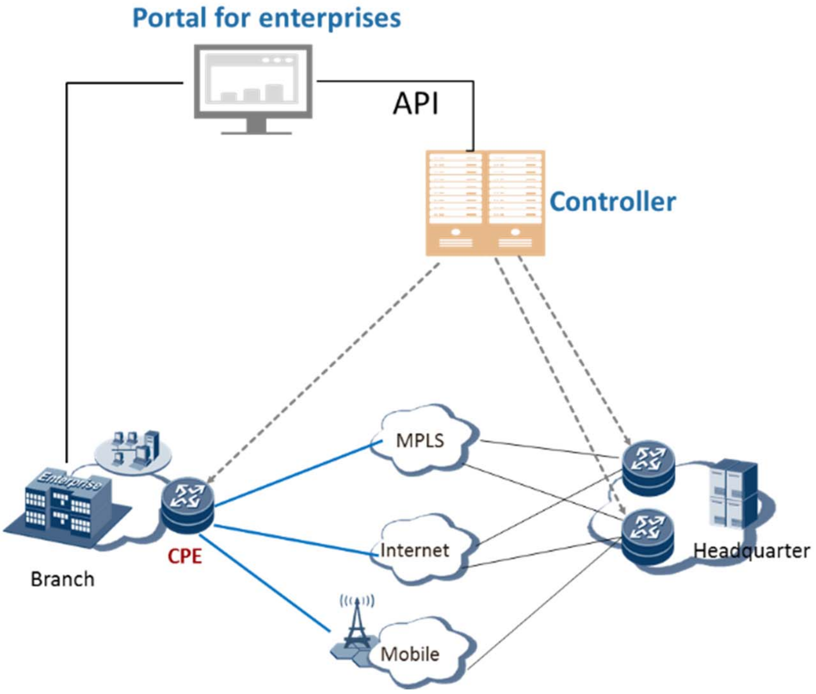


Figure 5-24: Intelligent carrier managed SD-WAN service

5.4.3.2.3 Actors and Roles

The presence of the following actors/entities as well as their associated roles are envisaged in the current Use Case:

- Carrier: Entity that provides MPLS/Internet connection to enterprises, as well as access to an web portal via one API in order to allow the enterprise to customize the services/connections.
- Network Administrator: Entity/person that configures the network topology and builds the WAN policies.
- ENI System: Entity that receives the intent policies from Network Administrators, monitors the services/connections and translates the intent policies into instructions/configurations to be enforced and executed by network devices.

Additionally, the ENI System may also optimize the services/connections and suggest policies, e.g. under the scenarios above identified.

5.4.3.2.4 Initial context configuration

The Network Administrator, according to enterprise's needs, configures the services/connections policies, e.g. Intent policies, for each application, which requires a large amount of backup work.

5.4.3.2.5 Triggering conditions

A new traffic pattern is detected by the ENI System as a new application and due to the collected information it is recognized as a potential new social network. The new traffic pattern is causing some optimization problems with Internet access.

5.4.3.2.6 Operational flow of actions

The following sequence of actions may be identified after the occurrence of the trigger:

1) The ENI System requests a confirmation from the Network Administrator to treat the new application as a social network.

2) The Network Administrator confirms that the new application is indeed a social network.

3) The ENI System identifies related policies and analyses past history looking for:

   a) Policy violations before the identification of the new application.

   b) Impact of the new application on network optimization.

4) The ENI System suggests the Network Administrator some changes to existing policies:

   a) Alter existing policies e.g.: "Personal devices cannot access Facebook more than one hour per day" to "Personal devices cannot access social networks more than one hour per day", in order to include both Facebook and the new application; or

   b) Add new policy e.g.: "All Social Network traffic have the lowest priority when accessing the Internet" to mitigate the access problems cause in the network by the new application when accessing the Internet.

5) The Network Administrator acknowledges the new policies and confirms the changes to the ENI System.

6) After confirmation, the ENI System provokes the enforcement of the new changes in policies by configuring appropriate network components.

5.4.3.2.7 Post-conditions

The new application is categorized and customized configuration is registered in the portal. The network traffic generated by different applications is routed to different paths according to the services/connections policies and application configuration.

After the changes to the network policies and devices configuration, the SD-WAN service is running under optimal conditions.

5.4.4 Use Case #3-4: Intelligent caching based on prediction of content popularity

5.4.4.1 Use case context

With the development of communication technologies, more and more user terminals will access the mobile cellular network, which will lead to explosive growth of network data traffic and content diversity. How to ensure that user terminals can be fast, effective and secure getting what users want is a big challenge for mobile networks. In today's cellular networks, a large amount of mobile traffic is generated by social and mobile applications. A significant portion of such traffic consists of popular contents that are repeatedly transmitted to mobile users and unnecessarily consume extra backhaul bandwidth and resource of radio access networks. Therefore, mobile caching (MC) has received significant attentions as an efficient approach to boost spectral efficiency and reduce backhaul load of mobile networks by bringing contents near mobile users. It is very important to improve efficiency of caching content management in MC due to limited cache storage resources within mobile cellular networks. The cache decision by content popularity prediction can improve the cache hit ratio of cached content, reduce the backhaul bandwidth cost of cellular networks, and reduce the content access delay of mobile users.

5.4.4.2 Description of the use case

5.4.4.2.1 Overview

In this use case, the AI-based MC is implemented in a mobile edge cache equipment, which includes but is not limited to a core network gateway, a base station, a user equipment. The AI-based MC (AI-MC) is divided into four modules: AI prediction module cache decision module; control module; and information collection module.

The prediction module, which is aligned with the knowledge & model function block in ENI architecture [i.9], is used to predict the popular content. It is divided into two sub-modules: a global AI prediction module and a local AI prediction module. The results of the local prediction module represent the content that the local user prefers, while the prediction results of global prediction module represent the content that the whole network user likes. In general, the prediction results of the two sub-modules are different.

The cache decision module, which is aligned with the knowledge & model function block in ENI architecture [i.9], will integrate the results from the two sub-modules, which form the AI prediction module, to make the final decision of which contents should be cached. The result of the cache decision module is defined as a content list sorted by the popularity of the content. The content recorded in the list will suggest to be cached

The control module, which is used to initialize other module at the beginning of the work, is responsible for the controlling of the work process of AI-MC, such as determining the work cycle. And it is aligned with the knowledge & model function block in ENI architecture [i.9].

The information collection module, which is aligned with the data ingestion function block in ENI architecture [i.9] is used to collect some key information which includes but is not limited to the hit ratio, the delay, which defined as the time since the user initiated the request until the response was received; the classification of content; the user behaviour, and to pre-process the information that has been collected to make the dataset.

The AI-MC provides the ability to cache popular content in the cache equipment in advance, which can significantly reduce network traffic and improve the user experience, like reducing the content acquisition latency.

5.4.4.2.2 Motivation

In order to improve user experience and reduce access latency in a congested network environment, caching has emerged. The current caching technology, which only follows the simplest replacement principle and does not perform caching operations on popular content, is not effective. This caching technology only improve network performance in a limited way. In view of the existing problems of existing caching technology, an AI-MC is proposed which includes four modules.

In this use case, The AI-MC, which are characterized by the artificial intelligence prediction modules and other supporting modules, are used in the ENI system. Artificial intelligence technology can dramatically improve the accuracy of popular content prediction. The AI-MC predicts popular content by loading trained model and collected information then caches popular content in advance, which plays a key role in reducing latency and improving network stability.

The system can be described in Figure 5-25.

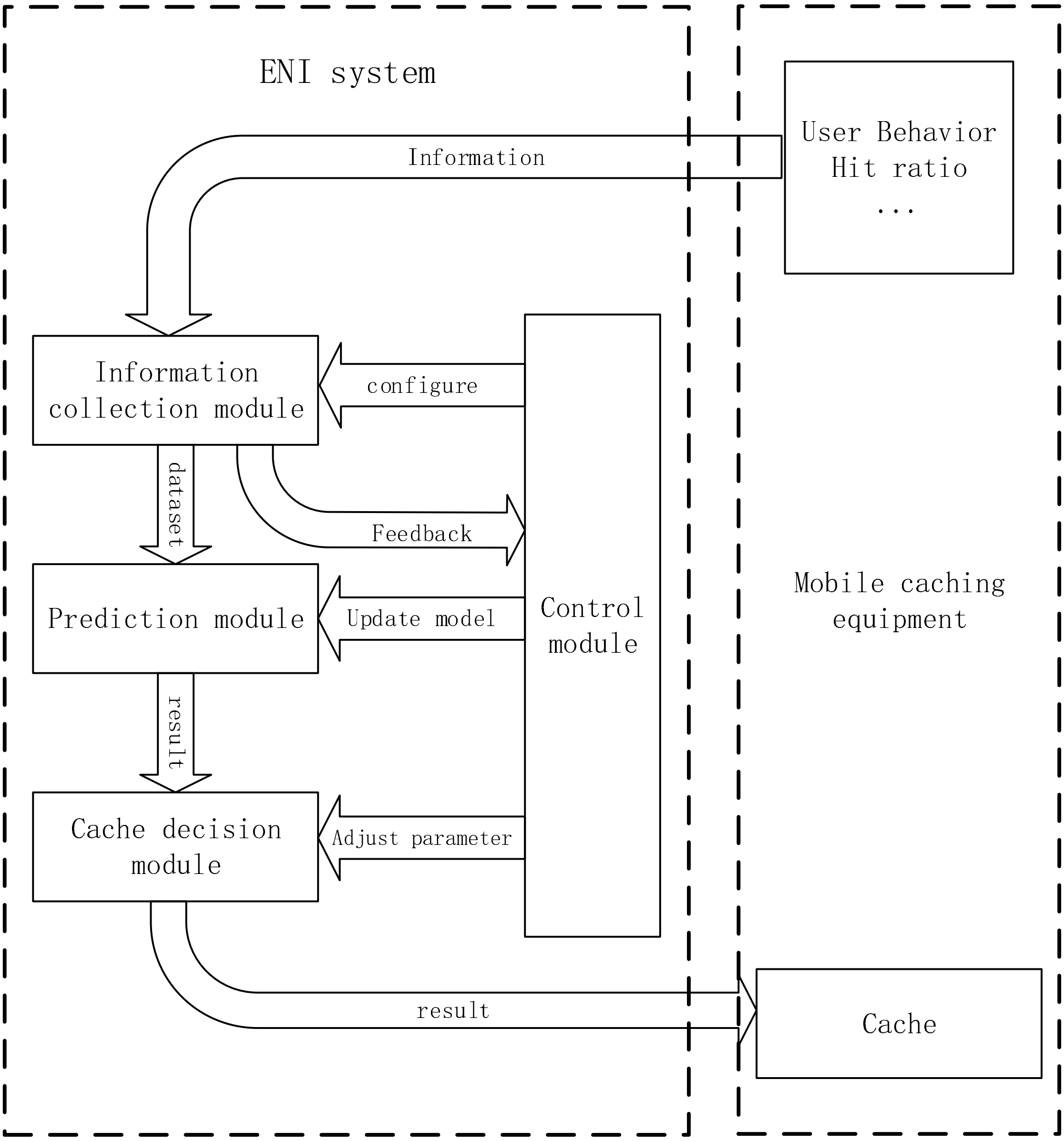


Figure 5-25: AI-based mobile caching

5.4.4.2.3 Actors and Roles

- ENI System: System solution used to receive target information such as the user behaviour from MC equipment; to provide results to external system in predefined formats.
- AI-MC: entity/person who derives results of popular content which is suggested to be cached.
- MC equipment: A device used to provide the necessary information such as users behaviour, hit ratio; to provide the storage resources.

5.4.4.2.4 Initial context configuration

- Mapping target content to predefined labels.
- Defining the prediction results formats.
- ENI system initials AI-MC.
- AI-MC in ENI system is enabled with AI capability, which is initialled to predict popular content through well-processed dataset.

- Related management systems connect to ENI system for acquiring results

5.4.4.2.5 Triggering conditions

In this use case, the cache hit ratio will be monitored by mobile edge cache equipment, when the cache hit ratio is lower than the predetermined value, or the prediction period is temporary, a new work process will be started under the control of the control module. In addition, the parameters of the AI prediction module will be adjusted by the control module based on the indicators.

5.4.4.2.6 Operational flow of actions

ENI system with AI-MC is intended to enhance the capability of mobile edge caching and optimize the content acquisition latency based on the prediction results of the popular content. This kind of mechanism is realized by introducing the new AI capability (e.g. normally based on the deep-learning algorithm and architecture), with the following flow of activities.

1) The ENI system initials a process to receive train data from database and pre-processes the data to make dataset.

2) The ENI system initials a process to train the AI prediction module according to the dataset which have been well-processed in 1).

3) The ENI system presents the prediction results by loading the AI prediction module.

4) The ENI system combines the result of the two prediction modules to derive the popular content.

5) The external system commanded to initial a caching process to cache the popular content which has not been cached and replace the cached content that is not marked as popular content in the final results.

5.4.4.2.7 Post-conditions

- In a work cycle, the ENI system needs to get metrics such as cache hit ratio, also needs to receive the user behaviour information.

- The control module updates the model parameters of the AI prediction module based on the information collected by the information collection module.

- The control module determines the caching period based on the configuration presented by the ENI system.

- The external system updates their cached content according to the result presented by ENI system.

5.5 Assurance

5.5.1 Use Case #4-1: Network fault identification and prediction

5.5.1.1 Use case context

For a network device or a network service, performance and other problems generally exist before the equipment/service fails. It is important to proactively identify and forecast status of a device/service that is not performing as expected in order for network operation and maintenance management to be able to repair the service before customer requirements are violated. Such identification and predication will need network information to be collected.

This use case takes wavelength division service as an example, which collects information such as FEC_bef, input optical power, laser bias current, and other key factors that can be selected. The information collected can be used to keep track of wavelength division service over time and calculate the device statistics data in a specific time period such as average device downtime in the specified time window. These statistics data can be further used to detect wavelength division service anomaly or improve the accuracy rate for wavelength division KPI anomaly detection.

5.5.1.2 Description of the use case

5.5.1.2.1 Overview

The development of artificial intelligence and big data technologies has brought new chance to the network operation and maintenance management. Big data technology can be applied to analyse huge data generated from network operation and maintenance management, and deep learning method can be used to construct the Intelligent Network Failure Prevention (INFP) system, which can be a sub-system of intelligent analysing and prediction in the ENI system. INFP system can help operators to reduce the OPEX in the network and promote service quality.

5.5.1.2.2 Motivation

Network failure can result in service disruption. The passive strategy is inefficient, and easily lead to long service interruption. By actively learning the health status of history data and intelligent partitioning the current service performance online, AI can be utilized to identify the potential sub-health services and rank these services according to health level. Taking again the example of wavelength division service, one minute rapid optical layer failure location can be achieved. Such a scenario is depicted in Figure 5-26.

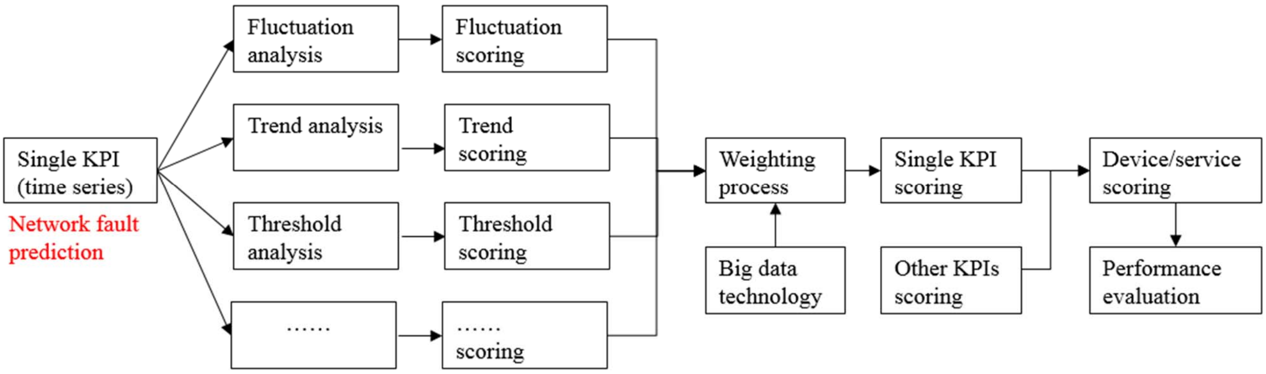


Figure 5-26: Network fault predication

5.5.1.2.3 Actors and Roles

- Network Administrators: define threshold for fault prediction.
- Network Performance Analysts: analyses fault prediction report.
- ENI system: collects and analyses data then predicts possible fault and produces detailed report.

5.5.1.2.4 Initial context configuration

Network time series data analysis comprises methods for analysing time series data in order to extract meaningful statistics and other characteristics of the data. Network performance changes over time. ENI system gathers data about the network and monitors the health status of the network.

For network equipment performance evaluation, multiple features are usually extracted from KPI data, such as fluctuation, trend, threshold, etc., and used as the key factors for anomaly analysis.

5.5.1.2.5 Triggering conditions

ENI system detects that network performance degrades below a threshold or the trends of metrics and statistics indicate that a fault may happen.

5.5.1.2.6 Operational flow of actions

1) ENI system calculates the network health indicator and predicts a possible fault, which may happen in future.

2) ENI system outputs detailed information about the fault (e.g. fault probability and fault coverage.

5.5.1.2.7 Post-conditions

Possible fault is identified and reported.

Service provided to customer is verified to be operating correctly.

5.5.2 Use Case #4-2: Assurance of Service Requirements

5.5.2.1 Use Case context

Nowadays, specific industries such as banking, energy or railroads use dedicated network infrastructures because it is the only way they can guarantee their specific requirements are met. These network infrastructures have huge costs in terms of planning and management, and take several months to be deployed. However, these industries would prefer to have Network Operators deploying and managing these private networks because that is not part of their core business. Moreover, during their lifetime operation, any change to the network infrastructure, no matter how small it is, is a cumbersome task due to the inherent complexity.

To overcome this scenario, Network Operators can replace these dedicated networks by other virtualized solutions where pinning of virtual resources may be made to physical ones. In that virtualized context, one of those solutions is the use of the network slicing feature, if slices are capable of meeting the requested strict requirements. In addition, the use of proper resource allocation techniques would also help to solve the situation. However, this approach is very difficult to comply with, because current resource allocation techniques are not able to provide the required performance and assurance with context-awareness capabilities.

5.5.2.2 Description of the Use Case

5.5.2.2.1 Overview

When combining network slices regarding a solution to the situation by making use of the slice/service prioritization and resource allocation concepts, it has to be considered that the dissemination of network slicing across network infrastructures will impact the virtual resource reservation and sharing since, unlike logical resources, physical resources cannot scale or migrate on demand.

NOTE: A network slice may encompass one or more than one services (mapping 1:1 or 1:n). When the mapping is 1:1, prioritization may either be designated by slice prioritization or by service prioritization.

In this Use Case, only a 1:1 mapping between slices and services is considered, as such slice prioritization will be used.

Considering that each slice may be assigned for a specific type of service or service class, Network Operators will need to enhance their operational systems with the necessary carrier grade assurance capabilities to guarantee the continuous delivery of services characterized by strict requirements. These capabilities are needed to resolve resource allocation conflicts between competing network slices deployed on top of a shared infrastructure in an efficient and dynamic manner. In a shared infrastructure, being aware of a constantly changing context is vital for triggering a set of actions in a timely manner, e.g. scaling resources in order to meet network slice requirements or increase the priority for specific network slices.

A network domain may run several network slices where one of them provides an infrastructure to a specific industry, e.g. an Energy Provider company. This Energy Provider uses the network slices for vital applications that enable them to operate their core business. Because these applications have very strict network requirements, the Network Operator and the Energy Provider establish a customized SLA.

The current Use Case is further described by the following set of components and features.

5.5.2.2.2 Motivation

By using AI and appropriate policies Network Operators will be able to predict potential hazardous situations, where two or more slices are competing for the same resources, and employ preventive measures, e.g. by using resource reservation. In other cases, more specifically when it is not possible to predict a certain scenario in advance, actions still need to be made at runtime in an autonomous way, e.g. by increasing the priority of a given network slice over neighbouring slices. Figure 5-27 provides a pictorial representation of the use case just described where the sequence of flow actions representing the resource behaviour in a specific section of an operator network infrastructure is depicted. It is meant to illustrate how an AI-based system is continuously monitoring and is able to predict fault starvation scenarios to trigger the most appropriate and optimal responses for mitigation e.g. slice prioritization enforcement.

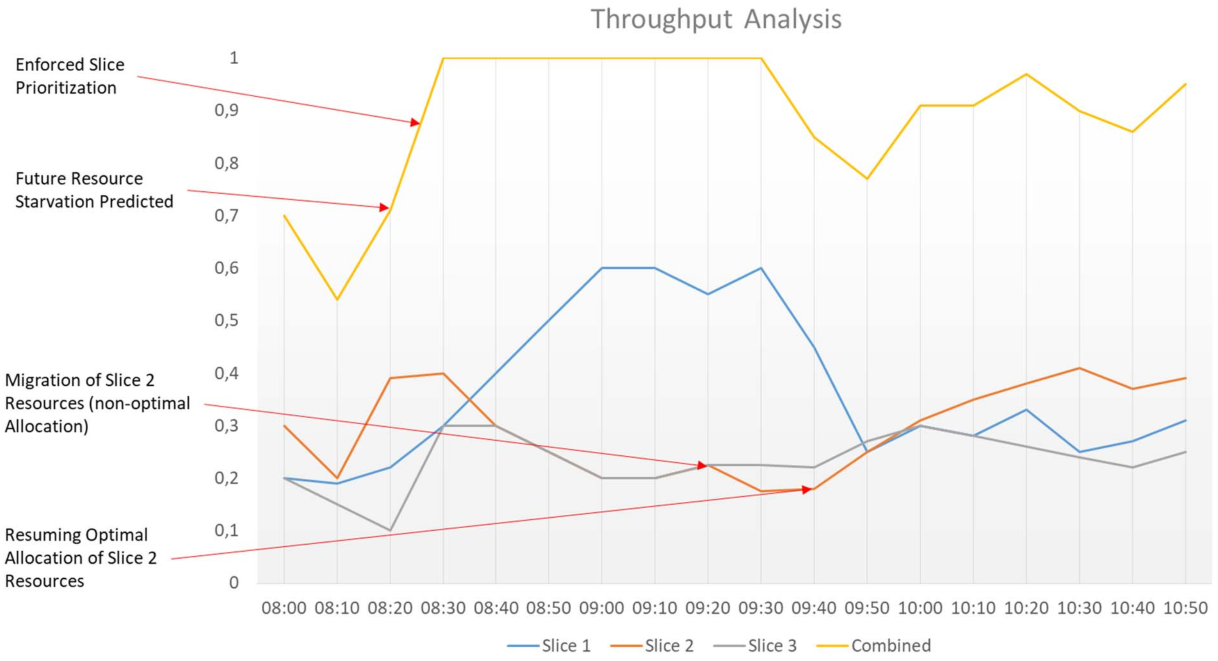


Figure 5-27: High priority Network Slice Assurance

5.5.2.2.3 Actors and Roles

The presence of the following actors/entities as well as their associated roles are envisaged in the current Use Case:

- **Customers/Clients:** specific vertical industries such as banking, energy or railroads that use dedicated network infrastructures.

NOTE: These entities, Customers/Clients, could also be considered as Assisted Systems.

- **Operations Support System/Business Support System (OSS/BSS):** Operational and Business Assisted Systems that belong to the management system of network operators. In this case, they enforce the strict operational and business requirements, which will be translated into slice policies that shall be accomplished in order to fulfil the agreed SLA between the Network Operator and the verticals, i.e. the Energy Provider.

- **Slice Management and Orchestration:** Assisted System administrated by a network operator that manages and orchestrates the life cycle of network slices.

- **Network Operator:** Owner of the Network Infrastructure that is used to provide services to customers/clients.

- **(Shared) Network Infrastructure:** (Shared) Infrastructure Assisted System used to create network slices and maintain their requirements, which delivers raw data to the ENI System during the training phase.

- **Network Slice Instance:** Representation of a network virtual component, which encompasses network functions, capabilities and resources, dynamically provisioned and assigned to a particular type of service or service class that are used by specific industries.

- **ENI System:** System solution used to predict or detect requirements change also involving possible competition for the same shared resources as well as to enforce slice prioritization.

5.5.2.2.4 Initial context configuration

The requirements agreed with customers, formally contracted in SLAs, are communicated by the OSS/BSS Assisted Systems to the Slice Management and Orchestration Assisted System as well as to the ENI System. Furthermore, the ENI System is also notified about the associated policies. At the same time, the Network Slices instances associated to each dedicated network are created and configured accordingly by the OSS/BSS Assisted Systems. All services are running with optimal resource allocation.

The ENI System is operating under normal conditions where raw data is being gathered and processed. Dedicated tools for knowledge extraction, filtering, and fusion are applied to different data sources in order to be combined, filtered, correlated, or otherwise processed to produce new knowledge. On the other hand, context-aware update is also taking place by monitoring the information about the characteristics and behaviour of the environment that ENI interacts with. This context-aware update also enables the ENI System to adapt its behaviour according to changes in context.

5.5.2.2.5 Triggering conditions

At a certain point in time, one slice reveals a considerable deviation from normal resource consumption patterns. Since this slice is deployed over a shared infrastructure where other slices and services are also provisioned, the abnormal behaviour may impact these other slices, including the one established for the Energy Provider, and eventually provoke the violation of the agreed SLA.

5.5.2.2.6 Operational flow of actions

The following sequence of high level actions may be identified after the occurrence of the trigger:

1) The ENI System makes a simulation for the consumption of resources according to the detected spike in a specific zone.

2) The ENI System projection, predicts that the spike will impact on the service delivery on what strict requirements is concerned hence violating past agreed SLAs.

3) In order to preserve the contracted SLAs, the ENI System enforces slice prioritization to guarantee that network slices with strict requirements continue to satisfy those SLAs, since the option of allocating more resources to those slices is not feasible due to a temporary lack of available resources on the above mentioned specific zone.

4) With the change on prioritization for the slices, one of lower priority starts suffering from resource starvation. Once detected by the ENI, in order to mitigate the impact, it makes use of dynamic resource allocation techniques and performs the migration of some resources to a temporary non-optimal location where, however, it is able to better accommodate the network slice.

5) After the spike in resource consumption disappears, if found as appropriate, the ENI System triggers the optimal allocation of resources to the network slice blueprint that was standing before the resource migration; afterwards all services are running with optimal resource allocation.

A more detailed flow diagram, showing an example of a possible solution for the deployment of this Use Case involving FBs of the ENI Reference Architecture, is depicted in clause 5.5.2.3.3 in order to complement the operational sequence of high level steps just described.

5.5.2.2.7 Post-conditions

The dedicated slices resume normal operation according to contracted SLAs. New knowledge as well as new context begin to be generated once again as normal operation activities are resumed.

5.5.2.3 Mapping to ENI reference architecture

5.5.2.3.1 Functional blocks

Table 5-2 contains an example of a set of Functional Blocks (FBs) belonging to the ENI Reference Architecture [i.9], which may be used on the implementation of a solution for the scenario described by this Use Case. For each FB, a brief description of the functionalities that may be accomplished by each one to address a particular feature of the Use Case is made.

NOTE: The applicability of each FB in terms of what is its role in the overall implementation of the Use Case can only be seen as an example based on the current ENI Reference Architecture [i.9].

Table 5-2: Mapping of ENI FBs to Use Case functionalities description

ENI Functional Blocks	Use case functionalities description
Data Ingestion and Normalization	The purpose of the Data Ingestion FB is to collect data from multiple input sources and implement common data processing techniques so that ingested data can be further processed and analysed by other ENI Functional Blocks. The purpose of the Normalization FB is to process and translate data received from the Data Ingestion Functional Block into a form that other ENI Functional Blocks can understand and use. For implementation purposes, both FBs can be combined, if desired. Upon receiving the uniform data internal format information, the other FBs are entitled to predict/detect abnormal events, i.e. traffic spike or potential resource starvation in the case of this UC.
Knowledge Management	The purpose of the Knowledge Management FB is to represent information about both the ENI system as well as the system being managed. It also enables machine learning and reasoning (e.g. by performing inference, correcting errors, and deriving new knowledge). The Knowledge Management framework may make use of dedicated tools for knowledge extraction, filtering, and fusion that may be required in environments where knowledge from different data sources needs to be combined, filtered, correlated, or otherwise processed in order to produce data, information, and knowledge. Upon prediction/detection, or becoming aware, of the occurrence of an event, i.e. a traffic spike or a potential starvation in the case of this UC, this FB may contribute to the delivery of a set of corrective action plans together with other FBs.
Context-Aware Management	The purpose of the Context-Aware Management FB is used to describe the state and environment in which a set of entities in the Assisted System exists or has existed. Context consists of measured and inferred knowledge, may change over time, and is continuously updated, hence help to adopt decisions to overcome hazardous situations. Its applicability to the present Use Case relies on the observation and update of the (adapted) behaviour of the network slices associated with the offer of services or type of services according to SLAs. Upon prediction/detection, or becoming aware, of the occurrence of an event, i.e. a traffic spike or a potential resource starvation in the case of this UC, this FB may contribute to the delivery of a set of corrective action plans together with other FBs.

ENI Functional Blocks	Use case functionalities description
Cognition Management	The purpose of the Cognition Management FB is to enable the ENI system to understand normalized ingested data and information, as well as the context that defines how those data were produced. Once that understanding is achieved, the Cognition Framework FB then evaluates the meaning of the data, and determines if any actions should be taken to ensure that the goals and objectives of the system will be met. Upon prediction/detection, or becoming aware, of the occurrence of an event, i.e. a traffic spike or a potential resource starvation in the case of this UC, this FB contributes to the delivery of a set of corrective action plans together with other FBs.
Situational Awareness	The purpose of the Situational-Awareness FB is to enable the ENI system to be aware of events and behaviour that are relevant to a set of entities in the environment of the Assisted System. This includes the ability to understand how information, events, and recommended commands provided by the ENI system will impact the management and operational goals and behaviour, both immediately and in the near future. The Situation Awareness FB is especially important in environments where the information flow is high, and poor decisions may lead to serious consequences (e.g. violation of SLAs). Upon prediction/detection, or becoming aware, of the occurrence of an event, i.e. a traffic spike or a potential resource starvation in the case of this UC, it works together with other FBs to produce a set of corrective action plans based on information, knowledge, and wisdom. In addition, assisted by the Model-Driven Engineering FB, it also performs the coordination of the analysis and decision making of verdicts, and chooses the optimal plan of action before delivering it to the Policy Management FB.
Model-Driven Engineering	The purpose of the Model Driven Engineering FB is to use a set of domain models that collectively abstract all important concepts for managing the behaviour of objects in the system(s) assisted by the ENI system. The MDE approach is meant to increase productivity by maximizing compatibility between Functional Blocks and systems through the reuse of standardized models. In this Use Case, it assists the Situation Awareness FB in the overall coordination of the analysis and decision making process.
Policy Management	The purpose of the Policy Management Functional Block is to provide decisions to ensure that the system goals and objectives are met. Policies are used to provide scalable and consistent decision-making and are generated from data and information received by the Knowledge Management and Processing set of Functional Blocks. Policies may represent recommendations or commands and are conveyed to the Assisted System or its Designated Entity through the Denormalization and Output Generation FBs, i.e. its applicability to the present Use Case relies on the execution of the action plan selected by the Situation Awareness.

ENI Functional Blocks	Use case functionalities description
Denormalization and Output Generation	The purpose of the Denormalization Functional Block is to process and translate data received from other Functional Blocks of the ENI system into a form that facilitates subsequent translation to a form that a set of targeted entities can understand. The purpose of the Output Generation Functional Blocks is to convert data received by the Denormalization Functional Block into a form that the Assisted System or its Designated Entity can understand. This may include defining an appropriate set of protocols, changing the encoding of the data, and other related functions. This FB can be combined with the Denormalization FB, if desired.
Lifecycle Management	N/A
Ancillary	N/A

5.5.2.3.2 Interfaces

Table 5-3 contains an example of a set of Reference Points (RPs) belonging to the ENI Reference Architecture [i.9], which may be used on the implementation of a solution for the scenario described by this Use Case. For each RP, a brief description of the functionalities that may be accomplished by each one to address a particular task of the Use Case is made.

NOTE: The applicability of each RP in terms of what is its role in the overall implementation of the Use Case can only be seen as an example, based on the current ENI Reference Architecture [i.9].

Table 5-3: Mapping of ENI RPs to Use Case functionalities description

ENI External Ref. Points	Use Case functionalities description (see note)
$E_{oss\text{-}eni\text{-}dat}$	Defines data and acknowledgements exchanged between ENI and the OSS-like Assisted System (or its OSS-like functionality), i.e. data sent to ENI and acknowledged by the Assisted System or by its Designated Entity.
$E_{oss\text{-}eni\text{-}cmd}$	Defines recommendations and/or commands exchanged between ENI and the OSS-like Assisted System (or its OSS-like functionality), i.e. recommendations and commands sent from ENI and acknowledged by the Assisted System or by its Designated Entity.
$E_{app\text{-}eni\text{-}ctx}$	Defines situation- and/or context-aware data and information and acknowledgements exchanged between ENI and the Slice Management FB of the Slice Management and Orchestrator Assisted System (considered as an application).
$E_{app\text{-}eni\text{-}kmo}$	Defines model and/or knowledge information and acknowledgements exchanged between ENI and the Slice Management FB of the Slice Management and Orchestrator Assisted System or by its Designated Entity.
$E_{app\text{-}eni\text{-}oth}$	Defines generic application data and acknowledgements exchanged between applications and ENI, that is neither situation - and/or context-aware data and also is not model or knowledge information.
$E_{bss\text{-}eni\text{-}dat}$	Defines data and acknowledgements exchanged between the BSS-like functionality and ENI, i.e. data sent to ENI and acknowledged by the Assisted System or by its Designated Entity.
$E_{bss\text{-}eni\text{-}cmd}$	Defines data and acknowledgements exchanged between the BSS-like functionality and ENI, i.e. recommendations and commands sent from ENI and acknowledged by the Assisted System.
$E_{or\text{-}eni\text{-}dat}$	Defines data and acknowledgements exchanged between ENI and the Slice Management FB of the Slice Management and Orchestrator, i.e. data sent to ENI and acknowledged by the Assisted System or by its Designated Entity.
$E_{or\text{-}eni\text{-}cmd}$	Defines commands and acknowledgements exchanged between ENI and the Slice Management FB of the Slice Management and Orchestrator, i.e. recommendations and commands sent from ENI and acknowledged by the Assisted System or by its Designated Entity.

ENI External Ref. Points	Use Case functionalities description (see note)
$E_{inf\text{-}eni\text{-}dat}$	Defines data and acknowledgements exchanged between the infrastructure and ENI, i.e. data sent to ENI and acknowledged by the Assisted System or by its Designated Entity.
$E_{inf\text{-}eni\text{-}cmd}$	Defines recommendations and/or commands, and acknowledgements, exchanged between the infrastructure and ENI, i.e. recommendations and commands sent from ENI and acknowledged by the Assisted System or by its Designated Entity.
NOTE: Functionalities description is an adaptation of the text provided in Table 6-4 of [i.9] taking into account the specific related functionalities for the present Use Case.	

5.5.2.3.3 Flow of information

Figure 5-28a depicted below illustrates a possible solution for the deployment of the Use Case that has been described in this clause, by making use of FBs that belong to the ENI Reference Architecture [i.9]. The functional operational procedures identified in clauses 5.5.2.2.4 to 5.5.2.2.7 are accomplished by the functionalities associated to each of the selected FBs. In addition, only the most relevant actions are shown as well as the exchange of messages between them.

In Figure 5-28a, the flow diagram is split into two parts because the deployment of the Use Case actually implies the detection and subsequent handling of two events: spike in resource consumption and resource starvation. The NOTES referred in the diagrams may be found in the step-by-step description that follows the first diagram.

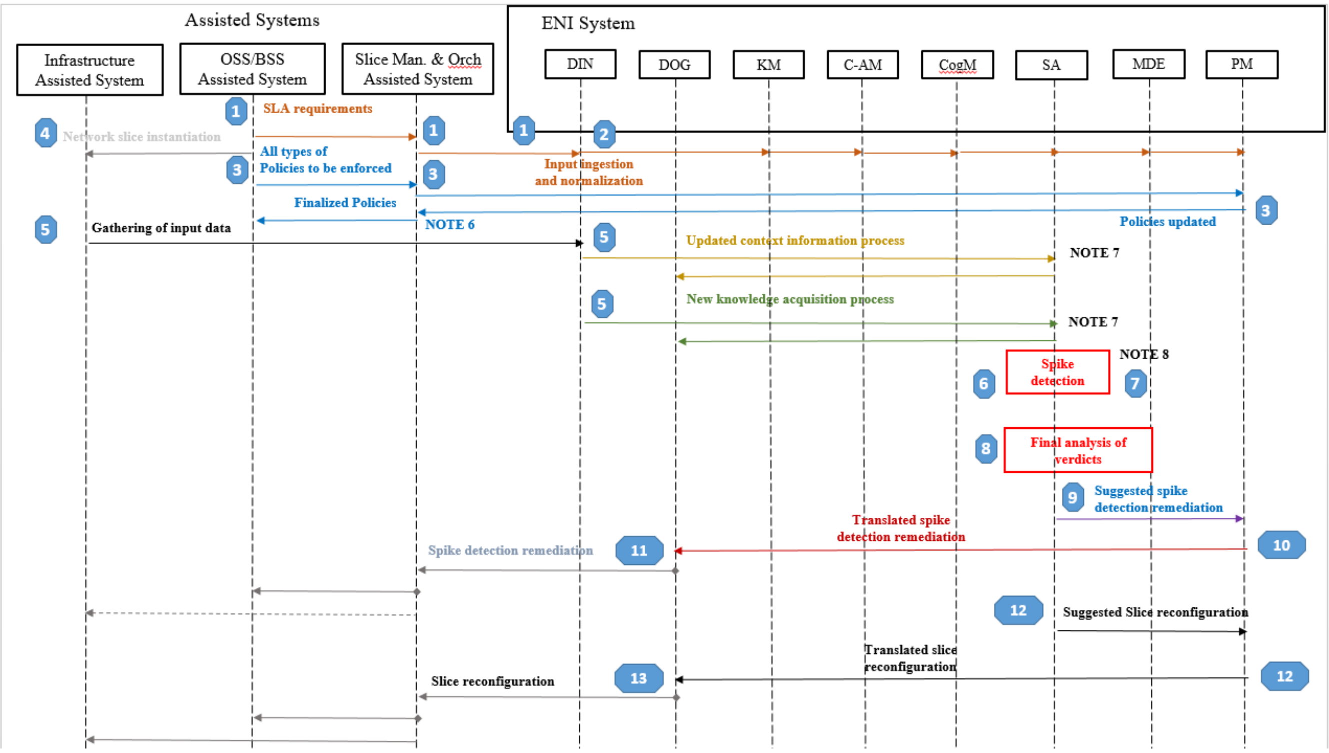

Assisted Systems
ENI System
Infrastructure Assisted System
OSS/BSS Assisted System
Slice Man. & Orch Assisted System
DIN
DOG
KM
C-AM
CogM
SA
MDE
PM
1
SLA requirements
4
Network slice instantiation
1
1
2
All types of Policies to be enforced
3
3
Input ingestion and normalization
Finalized Policies
3
Policies updated
5
Gathering of input data
NOTE 6
5
Updated context information process
NOTE 7
5
New knowledge acquisition process
NOTE 7
NOTE 8
Spike detection
6
7
8
Final analysis of verdicts
9
Suggested spike detection remediation
Translated spike detection remediation
Spike detection remediation
11
10
12
Suggested Slice reconfiguration
Translated slice reconfiguration
Slice reconfiguration
13
12

Figure 5-28a: Part 1: Spike detection and slice prioritization flow diagram

By using a step-by-step approach, the actions are described as follows:

1. The scenario for this Use Case starts with the formal settle down of SLAs with the customers, which implies the fulfilment of agreed strict requirements. These have to be communicated by the OSS/BSS Assisted Systems to the Slice Management and Orchestration Assisted System as well as to the ENI System.

2. The SLA requirements are received by the Data Ingestion and Normalization FBs, which perform its conversion into a normalized form in such way that can then be analysed and understood by the other FBs of the ENI system for further processing.

3. Furthermore, the ENI System shall also become aware of the possible associated policies, which may be enforced in all these FBs subsequently by the OSS/BSS Assisted System.

NOTE 1: The OSS/BSS Assisted System is entitled to enforce not only these associated policies but also any other type of slice policies at any time, see step 3 in Figure 5-28a.

4. At the same time, the Network Slices instances associated with each dedicated network are created and configured accordingly by the OSS/BSS Assisted System in the Infrastructure Assisted System. After that, all services are running with optimal resource allocation.

5. The ENI System enters then in its normal operation where raw data starts being gathered and processed after the conversion made by the Data Ingestion and Normalization FBs. The overall behaviour of the FBs, in this phase, is indicated in Table 5-2.

6. At a certain point in time, one slice reveals a deviation from normal resource consumption patterns and, since it is deployed over a shared Infrastructure Assisted System, where other slices and services are also provisioned, the abnormal behaviour may impact those other slices. Any events (e.g. alarms) or data that indicate this abnormal behaviour are received by the DIN FB. It may use various processes, including simulating the consumption of resources corresponding to this abnormal behaviour. After prediction/detection of the occurrence, other FBs, e.g. KM, C-AM, CogM, SA and MDE, become aware of the situation by extracting the information from the semantic bus. The SA FB stores all the parameters and associated configurations that make part of the network slice blueprint that was standing before the detection of the spike.

7. Upon taking care of the abnormal occurrence, each one of the involved FBs, including the Situational Awareness and the Model-Driven Engineering FBs, work together and may use a variety of algorithms to process data, and generate information, knowledge, and wisdom in order to prevent network slices from violating the agreed SLAs. In this way, they reach verdicts represented by action plans to be taken upon. Since for the case of the present UC, the option of allocating more resources to those slices is not feasible due to the lack of available resources on the affected specific zone, slice prioritization shall be used instead.

NOTE 2: A process involving back and forth messages, between all the concerned FBs and the MDE FB, may take place at this point. However, in order to simplify the drawing, it is not depicted.

8. Final analysis of verdicts conflicts, and evaluation of consequences, is coordinated by the Situational Awareness FB, which is in charge of the harmonization of the output decisions. In the execution of this task, it is assisted by the Model-Driven Engineering FB, and chooses the optimal plan of action avoiding the possibility of different plans having conflicting actions.

9. Once performed the evaluation of verdicts, and the selection of the ultimate action plan, the Situational Awareness FB sends the output data to the Policy Management FB.

10. The responsibility of the PM is to take the verdict (i.e. the final single action plan selected by the SA FB) and translate it into a set of Policies. After that, the PM FB forwards the (internally normalized) data to the Denormalization and Output Generation FBs.

11. In its turn, before sending the result to the external Assisted Systems, the Denormalization and Output Generation FBs process the output data received from the Policy Management FB, and translate it to an external format understandable by them.

12. Finally, the Situation Awareness FB, which had stored the parameters and associated configurations that make part of the network slice blueprint, initiates the reset to the conditions that applied when the spike was detected by sending an appropriate message to the Policy Management FB. In its turn, the PM FB translates it into a set of policies and sends the result to the Denormalization and Output Generation FB.

NOTE 3: The task described in step above is not usually performed since, typically, in network management, the action plan changes configurations in order to correct the problem, which implies returning an optimized state. After applying the plan of actions, the original state makes no sense anymore, however, the reset is performed in this Use Case just to show in the flow diagram the way it is performed.

13. Once received the message, the Denormalization and Output Generation FB performs the translation of its contents to a format that can be understand by the external Assisted Systems.

As a result of the projection, a resource starvation is also predicted on a specific zone of the infrastructure that is supporting several network slices.

The handling of this situation is shown in the second part of the flow diagram as depicted in Figure 5-28b.

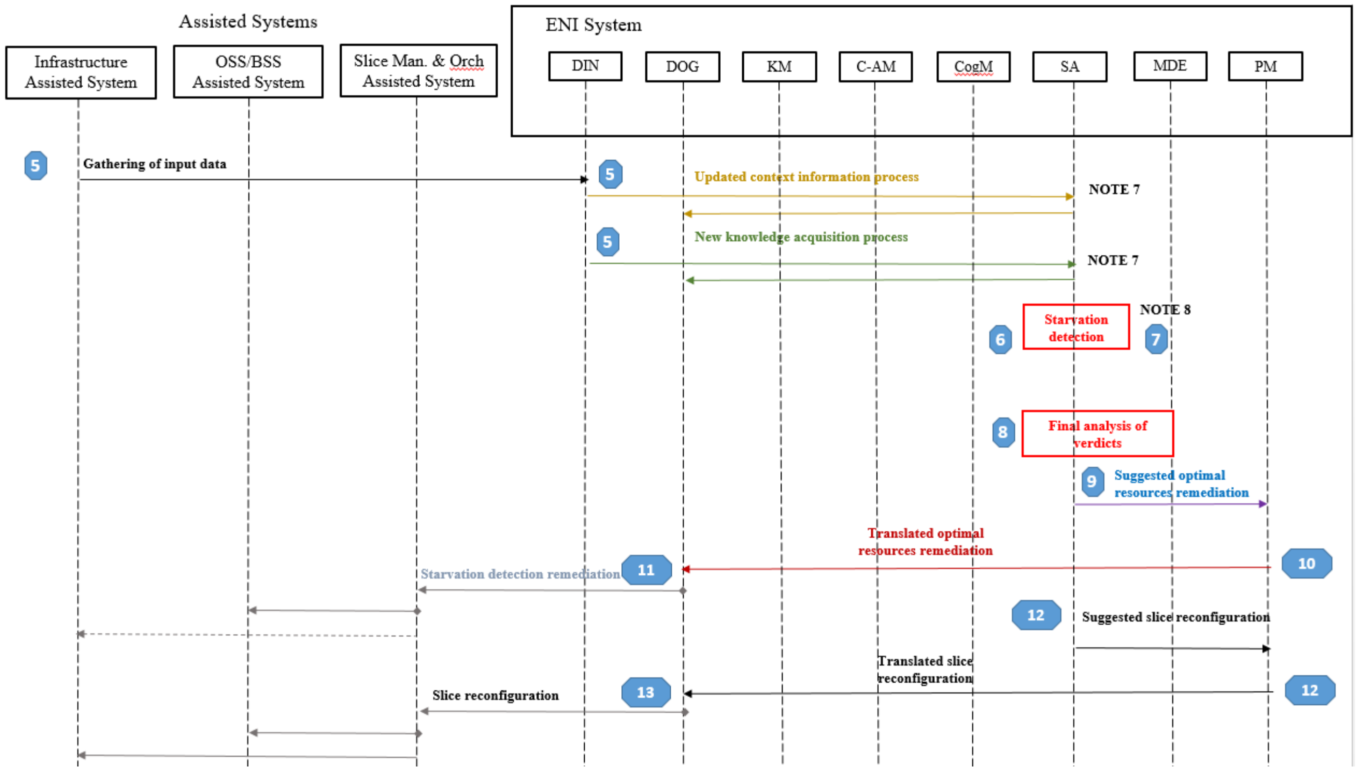

Assisted Systems
Infrastructure Assisted System
OSS/BSS Assisted System
Slice Man. & Orch Assisted System
ENI System
DIN
DOG
KM
C-AM
CogM
SA
MDE
PM
5
Gathering of input data
5
Updated context information process
NOTE 7
5
New knowledge acquisition process
NOTE 7
Starvation detection
NOTE 8
6
7
Final analysis of verdicts
8
9
Suggested optimal resources remediation
Translated optimal resources remediation
10
Starvation detection remediation
11
12
Suggested slice reconfiguration
Translated slice reconfiguration
Slice reconfiguration
13
12

Figure 5-28b: Part 2: Starvation detection and optimal resources reallocation

Again, the Situation Awareness FB, to mitigate the impact, may work together with other relevant FBs by making use of dynamic resource allocation techniques and reach a set of verdicts that are oriented to perform the migration of some resources to a temporary non-optimal location where, however, it is able to better accommodate the network slice. As for the spike occurrence, final analysis of verdict conflicts, and evaluation of consequences, is coordinated by the Situational Awareness FB with the assistance of the Model-Driven Engineering FB.

After the spike affecting resource consumption disappears, the Situational Awareness FB triggers the optimal allocation of resources to the network slice blueprint that was standing before the resource migration.

In the end, all services are according to the agreed SLA.

5.5.3 Use Case #4-3: Network fault root-cause analysis and intelligent recovery

5.5.3.1 Use case context

Traditional fault maintenance requires manual processing. The cost is high, and the fault locating efficiency is low and the period is long. It is hoped that applying machine learning algorithms in network fault root-cause analysis and intelligent recovery to form a more efficient solution, shorten the time of fault recovery and improve the efficiency of network maintenance.

When faults occur, AI algorithm (e.g. Decision Tree Algorithms) is used to the calculates the fault self-recovery policy with alarms data, network topology data, network service data collected from the monitoring system (MS). Then the fault self-recovery operation is delivered to network through the multi-vendor CMD platform (MCP). Self-recoverable faults can be quickly recovered and users are unaware of the faults. If a fault cannot be rectified, accurate diagnosis can be performed to locate the root cause (e.g. Big Data Mining Algorithms, Deep Learning Algorithms) and help engineers quickly rectify the fault.

5.5.3.2 Description of the use case

5.5.3.2.1 Overview

There are many difficulties and challenges in network operation and maintenance work, including:

- With the increasing scale and complexity of telecommunication network, operators not only have to monitor a large amount of real-time information generated by various highly integrated devices, but also need to deal with massive alarm data. In order not to reduce user experience, the faults should be quickly recovered.

- For the fault alarms, the current method is to complete fault diagnosis and recovery by manual execution of instructions, or need maintenance personnel to check on site, and then deal with the fault. This method has long recovery time, low operation and maintenance efficiency.

- Fault recovery work is complex. The experience cannot be shared, because the current fault recovery methods is different in different regional and most maintenance personnel change even every day.

It is imperative to achieve root cause analysis and self-recovery in maintenance work, and reduce the workload of maintenance personnel by using AI technology and ENI system.

5.5.3.2.2 Motivation

For solving the above problems and challenges, based on AI techniques the use case analyses and processes network alarms intelligently. The analysis results are feed to the external operation system (e.g. multi-vendor CMD platform and monitoring system) to improve the level of network maintenance intellectualization.

Such a scenario is illustrated in Figure 5-29. The following specific functions and objectives is proposed in the ENI system in this use case:

- Scenario analysis: This module is aligned with the knowledge & models function block in ENI architecture [i.9]. Building the base of self-recovery fault scenario based on the historical fault recovery data. When faults occurs, the fault scenario analysis function of ENI system is used to judge the self-recovery faults. This part of the fault does not dispatch tickets to deal with, thus reducing the number of tickets, reducing the cost of maintenance.

- Decision-making: This module is aligned with the knowledge & models function block in ENI architecture [i.9]. Calculates the fault self-recovery policy by using the decision-making model in ENI system, thus the time of artificial judgment is saved, and the fast recovery of fault is realized.

- Big Data Mining: This module is aligned with the knowledge & models function block in ENI architecture [i.9]. The data mining model in ENI system is used to achieve automatic generation of alarm correlation rules.

- RCA and SIA: This module is aligned with the knowledge & models function block in ENI architecture [i.9]. The function of Root Cause Analysis (RCA) model is to identify root alarm and derivative alarm based on the correlation rules. The Service Impact Analysis (SIA) model is used to find the correlation of root alarms and fault. Combining RCA and SIA functions to achieve the accurate location and delimitation of root cause fault, and then achieve accurate dispatch of tickets avoiding the errors and invalid tickets, improve manual fault recovery efficiency.

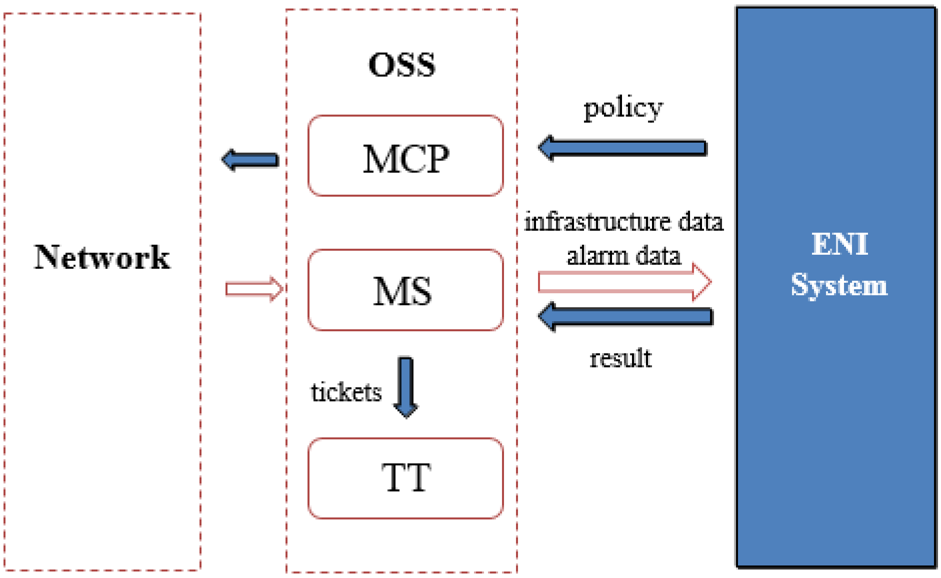


Figure 5-29: Scenario of AI and ENI system enabled network fault root-cause analysis and intelligent recovery

5.5.3.2.3 Actors and Roles

- MS: Monitoring System, monitor the real-time network alarms and dispatches the alarms data and network infrastructure data (e.g. network element data, network topology data, network service data) to ENI system. Generating tickets based on the result of ENI Feedback.
- TT: Trouble Ticket, issue the ticket and handing by maintenance personnel.
- MCP: Multi-vendor CMD Platform, responsible for command message passing from ENI system to the network.
- ENI system: system solution used to receive alarm data and network infrastructure data from MS; to analysis the fault scenario, calculates the fault self-recovery policy; to locate the root cause of network fault.

5.5.3.2.4 Initial context configuration

- Defining the analysis results formats of ENI system.
- Training intelligent decision model with the historical fault recovery data.
- Collecting alarm data and network infrastructure data in real-time.
- Building the base of alarm correlation rules and defining the parameters of data mining model, including time window, support and confidence.
- Related external systems connects to ENI system for acquiring decision-making and root-cause analysis results.

5.5.3.2.5 Triggering conditions

In this use case, as long as the ENI system receives the fault alarm dispatched by monitoring system, the function of fault scenario analysis and intelligent decision-making will be triggered. When the fault alarm cannot be self-recovery, RCA and SIA functions will be triggered, and the analysis results will be feedback to the monitoring system.

5.5.3.2.6 Operational flow of actions

ENI system with the AI capability, including scenario analysis, big data mining, decision- making, RCA&SIA, is intended to achieve root cause analysis and self-recovery in maintenance work, and reduce the workload of maintenance personnel. This kind of mechanism is realized with the following flow of activities:

- ENI system receives the fault alarm uploaded from the monitoring system, then it starts the fault scenario analysis function, and determines whether the fault is self-recovery.
- For the self-recovery fault, ENI system uses the intelligent decision-making model to calculate the fault self-recovery policy, and output the policy to MCP. The monitoring system monitors whether the fault has been recovered and feeds back to the ENI system, which can iteratively optimize the intelligent decision-making model through feedback information.
- Otherwise, ENI system triggers the functions of RCA and SIA with the correlation rules and network infrastructure data to locate the root cause of the fault.
- ENI system output the analysis results to the monitoring system to generate trouble ticket. Then trouble ticket system will issue the ticket and handing by maintenance personnel.
- If the fault is recovered, the result is fed back to ENI system to form a closed loop.

5.5.3.2.7 Post-conditions

- The base of fault scenarios is constantly enriched, and the application scope of intelligent decision-making model is improved, through the continuous learning of historical data.
- External monitoring system realizes intelligent monitoring by utilizing the alarm data analysis ability of ENI system.
- The base of alarm correlation rules is improving, through the mining of historical alarm data by ENI system. Then the accuracy of root cause analysis is increasing.
- The parameters of data mining model can be adapted according to the results of root cause analysis.

5.6 Network Security

5.6.1 Use Case #5-1: Policy-based network slicing for IoT security

5.6.1.1 Use Case context

In the near future, it is expected that smart cities will be built by using a myriad of IoT devices, where a significant number of them will be connected through 5G. These devices will play a vital role in the deployment of various services (e.g. civil protection or other services provided by the municipality, where each service will have its own target use and different device requirements).

To support this massive deployment of devices, the use of network slices will enable their aggregation either by functionality (e.g. security or city operations management support) or by other types of lower level requirements, such as low latency and high bandwidth.

In this context, the handling of Distributed Denial Of Service (DDOS) attacks plays a crucial role as those devices are usually meant to be part of the support to applications/services related to social interest.

NOTE: As an example of the severity of damage that these type of devices can achieve in such environments, consider the October 2016 IoT incident, where several different devices where infected with a Botnet malware designed to perform a DDOS attack. The initial reports point to an attack that roughly doubles previous massive attacks, all thanks to the nature of IoT, where a huge number of devices deployed in a distributed way can be used to target specific network infrastructures.

Description of the Use Case One of the key benefits of the network slicing concept, from the IoT perspective, is that it adds value by offering network and cloud resources that can be used in an isolated, disjunctive or shared manner. In this context, network slicing can be used to support very diverse requirements imposed by IoT services as well as flexibility and scalability to support massive connections of different natures.

Different slices may be used and re-tasked to accommodate changes in context. This requires coordination and management of each slice. It is recommended that one or more AI algorithms are used to pre- and/or post-process the information gathered prior to executing a set of policy rules to manage a set of slices. In addition, the use of different AI algorithms to monitor the execution of the policy rules is recommended to ensure that the new behaviour of the set of slices is correct. The use of AI at these different places in the control loop is necessary to support the integration of millions of devices in complex topologies and distinct communication patterns, while still guaranteeing infrastructure security and optimal resource usage.

The use of AI in concrete scenarios addressing specific situations that involve DDOS attacks enables the ENI System to provide automatic and dynamic responses in different contexts. For example, when a set of IoT devices become infected, they may cause a service degradation or disruption; hence, they need to be isolated in order to be repaired or replaced. This also prevents the spreading of malware.

5.6.1.2 Description of the Use Case

5.6.1.2.1 Motivation

One use of machine learning in the ENI System is to detect specific traffic patterns indicating DDOS or other type of attacks. This is because the increasing sophistication of such attacks makes it harder to use simpler algorithms (e.g. pattern recognition) that focus on a set of predefined information. The symptoms of a DDoS attack include unusually slow network performance and/or the inability to access a particular set of web sites. When this happens, the ENI System will be able to detect and learn from the occurrence by using AI methods. If the new traffic pattern is identified as an attack based on past history, the ENI System will be able to trigger appropriate responses from the related management components. In addition, AI enables different types of attacks to be correlated. For example, different attacks could use different protocols, but all be directed at the same target. This type of conclusion is extremely hard to make without using inferencing.

By using those techniques, the ENI System will be able to identify these and other types of attacks with a shorter timeframe and better precision when compared to today's systems.

Figure 5-30 provides a pictorial representation of the Use Case just described, where the first one shows the isolation of a network device once suspicious traffic behaviour is detected by the ENI System.

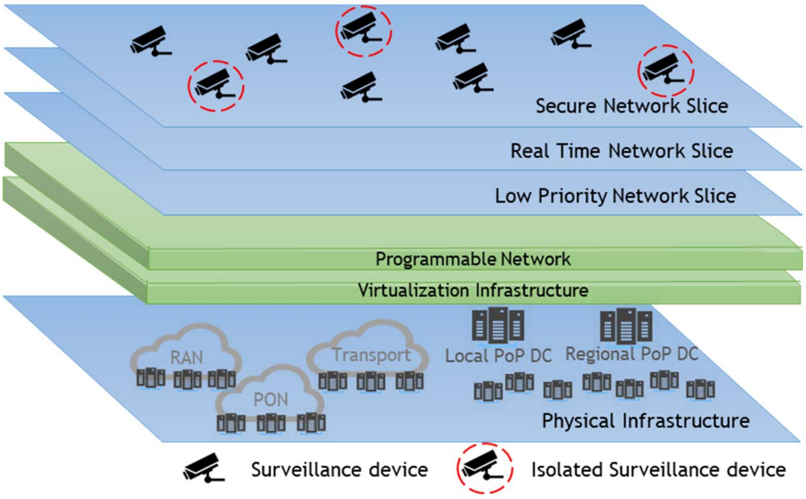


Figure 5-30: Device isolation within a Network Slice

5.6.1.2.2 Actors and Roles

The presence of the following actors/entities as well as their associated roles are envisaged in the current Use Case:

- Customers/clients: the operators themselves.
- Network Infrastructure: infrastructure that includes resources and devices that are meant to provide applications/services related to social interest.
- IoT devices: normal devices and infected devices, e.g. those that are victims of a Botnet malware attack.
- Network Administrator: entity/person responsible for the policy design that encompasses the isolation of devices that were victims of DDOS attacks.
- OSS/BSS: components that provide monitoring data slicing management functionalities for ENI to detect and mitigate attacks. In addition, they also provide interfaces to network administrators and customers.
- ENI System: System solution that makes use of AI methods to identify and trigger responses to attacks.

5.6.1.2.3 Initial context configuration

The network is operating correctly.

5.6.1.2.4 Triggering conditions

A first trigger is when the ENI System detects changes in services provided (e.g. a web site).

A second trigger is when the ENI System identifies abnormal traffic patterns from a set of devices.

5.6.1.2.5 Operational flow of actions

The following sequence of actions may be identified:

1) The ENI system monitors services (e.g. a web site) and devices that support the service itself (e.g. the network and a supporting server farm) as well as provide access to the service, looking for anomalous behaviour.

2) The ENI system detects that an abnormal event has occurred (e.g. web site is no longer accessible, or the traffic patterns of a device do not correspond to its expected behaviour).

3) The ENI system analyses the changes indicated by the abnormal event, and determines whether this is an attack or not. If it is an attack, then it notifies the Network Administrator, requesting the necessary operations to mitigate the attack.

4) These OSS/BSS entities enforce related policies and isolate the infected IoT devices from the rest of the network.

5) These OSS/BSS entities also notify the Network Administrator and related customers, if applicable, about the occurrence of the attack and restores normal service to the customer.

5.6.1.2.6 Post-conditions

The infected devices have been identified and isolated from the network in a swift and efficient manner, and all other devices were able to maintain their normal operations.

If the customer's service was interrupted by the attack, then the customer's service is restored.

5.6.2 Use Case #5-2: Limiting profit in cyber-attacks

5.6.2.1 Use Case context

There are a great activity over the globe from hackers searching how to obtain direct profits with the digital resources from victims that become their target. Among the several methods that a hacker make profits with an attack, this use case puts its focus on ransomware and cryptocurrency mining techniques, that are mostly being used by cyber-criminals and give them direct benefits from the victims.

Both attacks search for the same objective but with different effects. With ransomware the hacker use extortion of the victim under the threat of permanent damage in the victim's systems by encrypting their data. This type of attack due to the extortion is discover by the victim in a short time (regardless of the damage). On the contrary, with a cryptomining or cryptojacking technique, the attacker try to remain unnoticed and it is difficult to detect, only by illegal resource consumption (CPU, power & memory). Profit is directly related to the time use before being discovered and the number of infections achieved. Attack using cryptomining does not damage or destroy victim data in a first instance but use victim resources and at the end, there is a malware in the system that can be use in the future to cause a greater damage.

This use case provide a solution based in ENI system to limit the damage and therefore decrease the profit of cybercriminals attacks through a network operator.

5.6.2.2 Description of the Use Case

5.6.2.2.1 Motivation

Adjusting to the definition of the ENI system, this use case suggests to leverage network virtualization technologies (NFV) in process of adoption by service provider to offer detection and prevention functionalities as services rather than as products. Moving towards a Security-as-a-Service paradigm allows providing different types of security functionality as detection and mitigation of ransomware and cryptomining attacks as is proposal in this use case. This approach allows putting ISPs, Telecommunications operators or Data Centres in a position to offer security services to customers at lowered cost and with reduced CAPEX.

A data centre infrastructure is extended by a certain number of distributed computing clusters to accommodate VNFs at various locations in the network (PoP, clients premises, etc.). The VNFs perform a wide variety of operations, including security-related ones. Depending on the type of cyber-threats they are meant to detect or mitigate, VNFs may implement e.g. virtual firewalls and intrusion detection systems (IDS).

This network architecture is completed with an ENI entity with specific machine learning algorithms trained to detect ransomware and cryptojacking attacks and an intent based policy language that automatically proposes new security policies to the OSS. The latter is in charge to enforce the policies through the NFVO into the VNFs, following policy-driven control loop.

Figure 5-31 shows the proposal architecture for this use case.

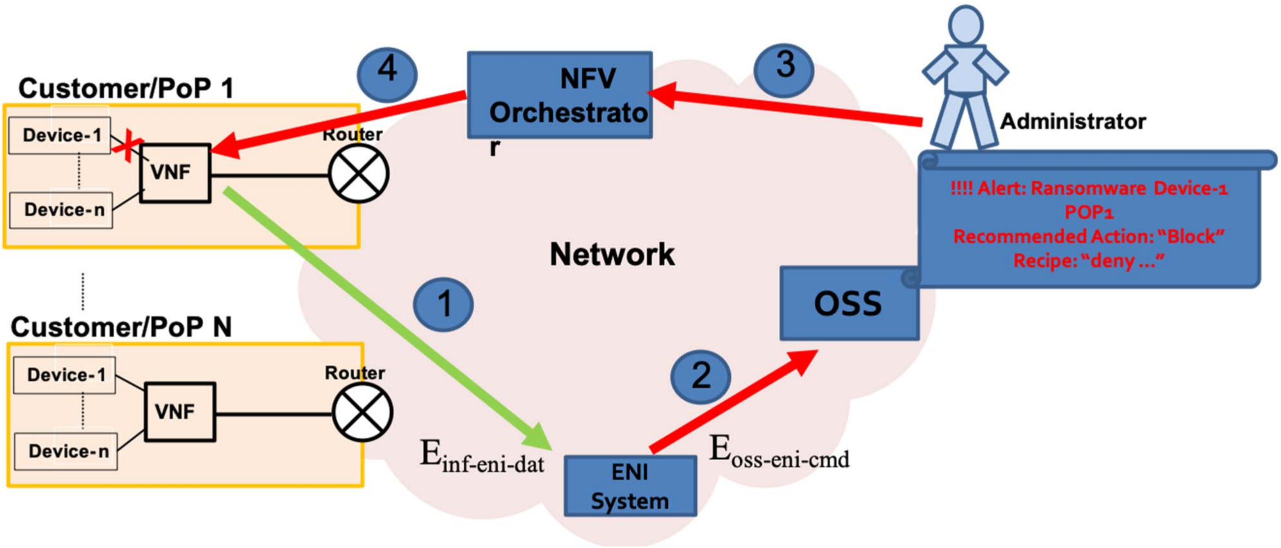


Figure 5-31: Use case to limit profit in cyber-attacks using NFV and ENI system

5.6.2.2.2 Actors and Roles

- Customer devices.
- ENI System: System solution that makes use of AI methods to identify and trigger responses to attacks.
- Network Infrastructure with NFV capacities (NFVI).
- NFV MANO including an Orchestrator (NFVO), that manage interconnectivity for the services.
- Network Administrator taking decisions to apply the proposed policy by ENI System.
- OSS providing interfaces to network administrators and customers.

5.6.2.2.3 Initial context configuration

The network is operating correctly.

5.6.2.2.4 Triggering conditions

The triggering is activated when the machine learning algorithm detects an anomaly based on the collected data from the network traffic (e.g. network flows) and is classified as a ransomware or a cryptojacking attack.

5.6.2.2.5 Operational flow of actions

- Network traffic is monitored by the ENI system monitoring service (e.g. data collected from a virtual DPI or netflow probe) (Step 1 in Figure 5-31).
- The machine learning algorithms, in the ENI system, analyses traffic data collected, by means of aggregation and normalization, to produce relevant insights about the attacks.
- The ENI system identifies an attack in the early stage of propagation or mining, and notifies the Network Administrator and propose a recipe (a security policy) to block the infected systems to avoid its propagation or its operation in the case of cryptojacking (Step 1 in Figure 5-31).
- The OSS accepts and applies (Steps 3 and 4 in Figure 5-31) the policy based on ENI recommendation:
    - Ransomware: Isolating affected devices and related protocols from the rest of the network.

- Cryptojacking:
  a) blocking the cryptomining functionality (block DNS queries, connectivity to mining pools or proxies); or
  b) isolating it from the rest of the network if we considered it a high risk for the network.

5.6.2.2.6 Post-conditions

The infected devices have been identified and isolated stopping the spread of the attack.

The customer can restore the situation in the infected devices (antimalware cleaning, reinstalling, etc.).

5.6.2.3 Mapping to ENI reference architecture

5.6.2.3.1 Functional blocks

The global system (network infrastructure including NFVI, NFV Orchestrator) should be considered a Class 1 assisted system (An Assisted System that has no AI-based capabilities) from the point of view of ENI architecture. This is the current situation of the ETSI MANO functionality, NFVO has not embedded AI engine, but an orchestration capacity based on External Designated Entity of the assisted system (i.e. the operator, OSS, BSS).

In relation with Operation mode, the behavioural is "recommendation mode" [MOP.1]. This mode is recommended in order to allow the operator to acknowledge the mitigation actions, and avoid automatic responses especially if we consider legal aspect of alter client traffic [MOP.5]. Security problem detected by ENI System, is reported to the OSS Dashboard including a recommendation action to solve the problem [MOP.7].

Table 5-4 shows the mapping between ENI System module and ENI architecture functional block.

Table 5-4: Mapping of ENI RPs to Use Case functionalities description

ENI Functional Blocks	Use case description
Knowledge Representation and Management	ENI will have a knowledge representation of the network being monitored (e.g.: NFV network service descriptor, ISP network topology, address pools allocation clients, network gateway, etc.).
ENI Ingestion and Normalization	Different VNFs distributed over the network collect and aggregate traffic information and sent it to the ENI using standard streaming protocols to this functional block, who support several data types (network flows, event logs or alerts from the network) and normalize them in a common format, interpreted easily by ML algorithms (such an array of dataset)
Context-Aware Management	Directions of the network flows (client to internet or vice versa) and normal traffic profiles (bandwidth, common protocols) are store in this block.
Situational Awareness	N/A
Policy Management	Previous blocks jobs allow to select the best recipe expressed in an Intend based language: such as "block traffic from client IP X" or "deny access" to the domain "miningpool.example.com". It will be reported to the OSS (recommendation mode).
Denormalization and Output Generation	Policies are translated to a format understandable by the technology provided by the VNFs or security device (e.g.: firewall rule command to block a IP source address to any destination, or configuration change in DNS server database to point to answer NXDOMAIN for "miningpool.example.com").
Cognition Framework	Uses specific machine learning algorithms tailored to detect specific ransomware and cryptomining.
Lifecycle Management	N/A
Ancillary	N/A

5.6.2.3.2 Interfaces

Table 5-5 describes the Reference point used in this use case where there is only interactions with external system shown in Figure 5-31.

Table 5-5: Mapping of ENI RPs to Use Case interface description

ENI External Reference Points	Description
$E_{oss\text{-}eni\text{-}cmd}$	OSS receive security policies recommendations to address security threats from ENI System. Recommendation mode use a feasible language compatible with the OSS, such as XML, YAML, etc. Flow (2) in Figure 5-31.
$E_{inf\text{-}eni\text{-}dat}$	ENI System collect data for security monitoring, including networks flows activity, or VNFs system logs. Flow (1) in Figure 5-31.

5.6.2.3.2 Flow of information

The flow of information is given in Figure 5-32.

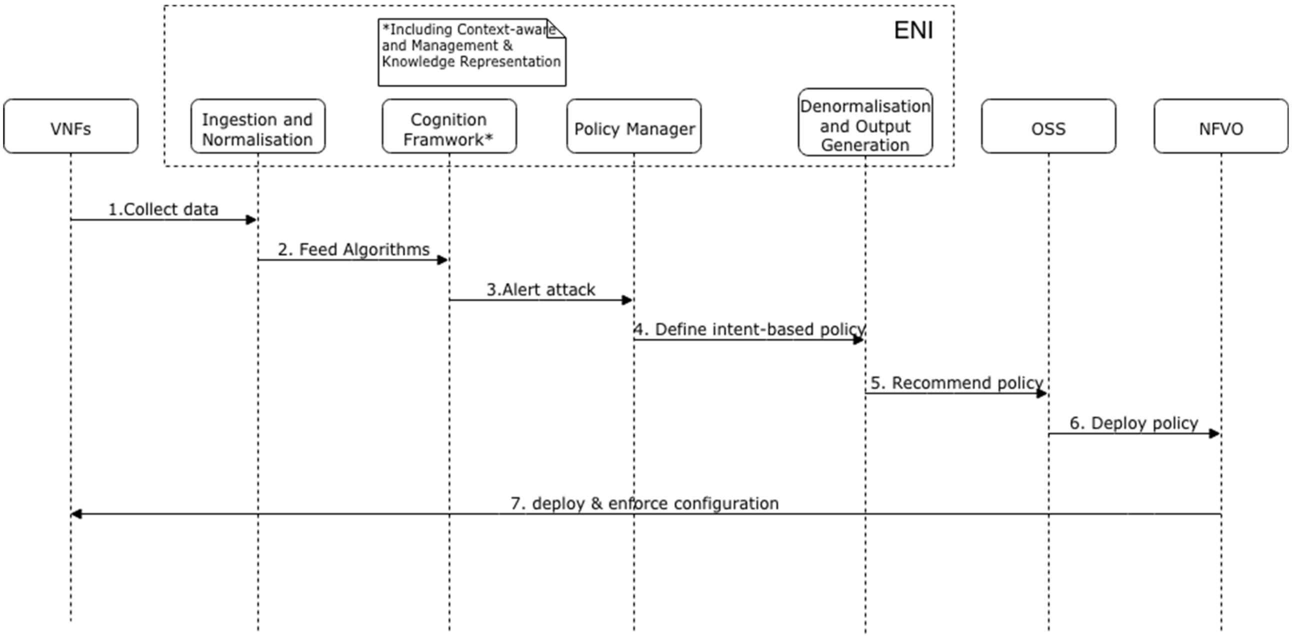

*Including Context-aware and Management & Knowledge Representation
ENI
VNFs
Ingestion and Normalisation
Cognition Framwork*
Policy Manager
Denormalisation and Output Generation
OSS
NFVO
1.Collect data
2. Feed Algorithms
3.Alert attack
4. Define intent-based policy
5. Recommend policy
6. Deploy policy
7. deploy & enforce configuration

Figure 5-32: Flow of information among the ENI System's functional blocks and the Assisted System

Step 1: The monitoring VNFs sends traffic information to the ENI system through the interface $E_{inf\text{-}eni\text{-}dat}$

Step 2: The Ingestion and Normalization Functional Block translate data from multiple input formats into a normalized form and feed to the machine learning models

Step 3: An alert is produced by the models in the Cognition Framework functional block with the aim of additional functional blocks

Step 4: The policy manager defines a recipe (intent based security policy) to mitigate the ransomware or criptojacking attacks

Step 5: The mitigation policy recommendation is sent in valid format that the OSS can understand through the interface $E_{oss\text{-}eni\text{-}cmd}$

Step 6: OSS through and operator decide to apply the recommended policy

Step 7: NFV Orchestrator enforce the mitigation policy using ETSI NFV reference points. This enforcement can include deploy new VNFs or configuration changes over the existing ones

6 Recommendations to ENI

The requirements extracted from the Use Cases captured in the present document are specified in the Requirement document [i.8]. These requirements are divided into service requirements, functional requirements, and non-functional requirements. The service requirements are further sub-divided into: general requirements; service orchestration and management; network planning and deployment; network optimization; resilience and reliability; security and privacy. The functional requirements are further sub-divided into: data collection and analysis, policy management and data learning. The non- functional requirements are further sub-divided into performance requirements, operational requirements and regulatory requirements.

The ENI architecture [i.9] will support all the requirements generated from the Use Cases.

Annex A (informative): Authors & contributors

The following people have contributed to the present document:

Rapporteur:
Yue Wang, Samsung R&D Institute UK

Other contributors:
Antonio Gamelas, Portugal Telecom

Bruno Parreira, Portugal Telecom

Georgios Karagiannis, Huawei

Haining Wang, China Telecom

Jizhuang Zhao, China Telecom

John Strassner, Huawei

Mehrdad Shariat, Samsung R&D Institute UK

Shucheng (Will) Liu, Huawei

Weiping Xu, Huawei

Xiaojian Ding, Huawei

Yu Zeng, China Telecom

Yali Wang, Huawei

LiLei Wang, AsiaInfo

Weiyuan Li, China Mobile

Bingming Huang, China Unicom

Annex B (informative): Bibliography

ETSI GR ENI 003 (V1.1.1): "Experiential Networked Intelligence (ENI); Context-Aware Policy Management Gap Analysis".

History

Document history		
V1.1.1	April 2018	Publication as ETSI GR ENI 001
V2.1.1	September 2019	Publication