International Telecommunication Union

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

Y.3177

(02/2021)

SERIES Y: GLOBAL INFORMATION INFRASTRUCTURE, INTERNET PROTOCOL ASPECTS, NEXT-GENERATION NETWORKS, INTERNET OF THINGS AND SMART CITIES

Future networks

Architectural framework for artificial intelligence-based network automation for resource and fault management in future networks including IMT-2020

Recommendation ITU-T Y.3177

ITU-T Y-SERIES RECOMMENDATIONS

GLOBAL INFORMATION INFRASTRUCTURE, INTERNET PROTOCOL ASPECTS, NEXT-GENERATION NETWORKS, INTERNET OF THINGS AND SMART CITIES

GLOBAL INFORMATION INFRASTRUCTURE	
General	Y.100–Y.199
Services, applications and middleware	Y.200–Y.299
Network aspects	Y.300–Y.399
Interfaces and protocols	Y.400–Y.499
Numbering, addressing and naming	Y.500–Y.599
Operation, administration and maintenance	Y.600–Y.699
Security	Y.700–Y.799
Performances	Y.800–Y.899
INTERNET PROTOCOL ASPECTS	
General	Y.1000–Y.1099
Services and applications	Y.1100–Y.1199
Architecture, access, network capabilities and resource management	Y.1200–Y.1299
Transport	Y.1300–Y.1399
Interworking	Y.1400–Y.1499
Quality of service and network performance	Y.1500–Y.1599
Signalling	Y.1600–Y.1699
Operation, administration and maintenance	Y.1700–Y.1799
Charging	Y.1800–Y.1899
IPTV over NGN	Y.1900–Y.1999
NEXT GENERATION NETWORKS	
Frameworks and functional architecture models	Y.2000–Y.2099
Quality of Service and performance	Y.2100–Y.2199
Service aspects: Service capabilities and service architecture	Y.2200–Y.2249
Service aspects: Interoperability of services and networks in NGN	Y.2250–Y.2299
Enhancements to NGN	Y.2300–Y.2399
Network management	Y.2400–Y.2499
Network control architectures and protocols	Y.2500–Y.2599
Packet-based Networks	Y.2600–Y.2699
Security	Y.2700–Y.2799
Generalized mobility	Y.2800–Y.2899
Carrier grade open environment	Y.2900–Y.2999
FUTURE NETWORKS	**Y.3000–Y.3499**
CLOUD COMPUTING	Y.3500–Y.3599
BIG DATA	Y.3600–Y.3799
QUANTUM KEY DISTRIBUTION NETWORKS	Y.3800–Y.3999
INTERNET OF THINGS AND SMART CITIES AND COMMUNITIES	
General	Y.4000–Y.4049
Definitions and terminologies	Y.4050–Y.4099
Requirements and use cases	Y.4100–Y.4249
Infrastructure, connectivity and networks	Y.4250–Y.4399
Frameworks, architectures and protocols	Y.4400–Y.4549
Services, applications, computation and data processing	Y.4550–Y.4699
Management, control and performance	Y.4700–Y.4799
Identification and security	Y.4800–Y.4899
Evaluation and assessment	Y.4900–Y.4999

For further details, please refer to the list of ITU-T Recommendations.

Recommendation ITU-T Y.3177

Architectural framework for artificial intelligence-based network automation for resource and fault management in future networks including IMT-2020

Summary

Recommendation ITU-T Y.3177 specifies an architectural framework for network automation based on artificial intelligence (AI) for resource and fault management in future networks, including international mobile telecommunications-2020. The purpose of the framework is to improve network efficiency and performance by continuously monitoring the network and promptly determining appropriate actions for resource adaptation and fault recovery with the help of AI, including machine learning.

History

Edition	Recommendation	Approval	Study Group	Unique ID*
1.0	ITU-T Y.3177	2021-02-13	13	11.1002/1000/14598

Keywords

Artificial intelligence, fault management, machine learning, network automation, resource management.

* To access the Recommendation, type the URL http://handle.itu.int/ in the address field of your web browser, followed by the Recommendation's unique ID. For example, http://handle.itu.int/11.1002/1000/11830-en.

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications, information and communication technologies (ICTs). The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

Compliance with this Recommendation is voluntary. However, the Recommendation may contain certain mandatory provisions (to ensure, e.g., interoperability or applicability) and compliance with the Recommendation is achieved when all of these mandatory provisions are met. The words "shall" or some other obligatory language such as "must" and the negative equivalents are used to express requirements. The use of such words does not suggest that compliance with the Recommendation is required of any party.

INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU had not received notice of intellectual property, protected by patents/software copyrights, which may be required to implement this Recommendation. However, implementers are cautioned that this may not represent the latest information and are therefore strongly urged to consult the appropriate ITU-T databases available via the ITU-T website at http://www.itu.int/ITU-T/ipr/.

Table of Contents

Recommendation ITU-T Y.3177

Architectural framework for artificial intelligence-based network automation for resource and fault management in future networks including IMT-2020

1 Scope

This Recommendation specifies an architectural framework for network automation based on artificial intelligence (AI) for resource and fault management in future networks, including international mobile telecommunications-2020 (IMT-2020). The purpose of the framework is to improve network efficiency and maintain quality of service (QoS) by continuously monitoring the network and promptly determining appropriate actions by using AI, including machine learning (ML).

The scope of this Recommendation includes:

- high-level architecture of network automation for resource and fault management with AI including ML;
- resource management functions;
- fault management functions.

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[ITU-T Y.3102] Recommendation ITU-T Y.3102 (2018), *Framework of the IMT-2020 network*.

[ITU-T Y.3111] Recommendation ITU-T Y.3111 (2017), *IMT-2020 network management and orchestration framework*.

[ITU-T Y.3112] Recommendation ITU-T Y.3112 (2018), *Framework for the support of network slicing in the IMT-2020 network*.

[ITU-T Y.3172] Recommendation ITU-T Y.3172 (2019), *Architectural framework for machine learning in future networks including IMT-2020*.

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1 fault management [b-ITU-T X.790]: Fault management consists of a set of functions that enable the detection, isolation, and correction of abnormal operation of the telecommunications network and its environment.

3.1.2 network function [b-ITU-T Y.3100]: In the context of IMT-2020, a processing function in a network.

NOTE 1 – Network functions include but are not limited to network node functionalities, e.g., session management, mobility management and transport functions, whose functional behaviour and interfaces are defined.

NOTE 2 – Network functions can be implemented on a dedicated hardware or as virtualized software functions.

NOTE 3 – Network functions are not regarded as resources, but rather any network functions can be instantiated using the resources.

3.1.3 network slice [b-ITU-T Y.3100]: A logical network that provides specific network capabilities and network characteristics.

NOTE 1 – Network slices enable the creation of customized networks to provide flexible solutions for different market scenarios which have diverse requirements, with respect to functionalities, performance and resource allocation.

NOTE 2 – A network slice may have the ability to expose its capabilities.

NOTE 3 – The behaviour of a network slice is realized via network slice instance(s).

3.1.4 network slice instance [b-ITU-T Y.3100]: An instance of network slice, which is created based on a network slice blueprint.

NOTE 1 – A network slice instance is composed of a set of managed run-time network functions, and physical/logical/virtual resources to run these network functions, forming a complete instantiated logical network to meet certain network characteristics required by the service instance(s).

NOTE 2 – A network slice instance may also be shared across multiple service instances provided by the network operator. A network slice instance may be composed of none, one or more sub-network slice instances which may be shared with another network slice instance.

3.1.5 machine learning (ML) [ITU-T Y.3172]: Processes that enable computational systems to understand data and gain knowledge from it without necessarily being explicitly programmed.

NOTE 1 – This definition is adapted from [b-ETSI GR ENI 004].

NOTE 2 – Supervised machine learning and unsupervised machine learning are two examples of machine learning types.

3.1.6 machine learning model [ITU-T Y.3172]: Model created by applying machine learning techniques to data to learn from.

NOTE 1 – A machine learning model is used to generate predictions (e.g., regression, classification, clustering) on new (untrained) data.

NOTE 2 – A machine learning model may be encapsulated in a deployable fashion in the form of a software (e.g., virtual machine, container) or hardware component (e.g., IoT device).

NOTE 3 – Machine learning techniques include learning algorithms (e.g., learning the function that maps input data attributes to output data).

3.1.7 machine learning overlay [ITU-T Y.3172]: A loosely coupled deployment model of machine learning functionalities whose integration and management with network functions are standardized.

NOTE – A machine learning overlay aims to minimize interdependencies between machine learning functionalities and network functions using standard interfaces, allowing for the parallel evolution of functionalities of the two.

3.1.8 machine learning pipeline [ITU-T Y.3172]: A set of logical nodes, each with specific functionalities, that can be combined to form a machine learning application in a telecommunication network.

NOTE – The nodes of a machine learning pipeline are entities that are managed in a standard manner and can be hosted in a variety of network functions [b-ITU-T Y.3100].

3.1.9 machine learning underlay network [ITU-T Y.3172]: A telecommunication network and its related network functions which interfaces with corresponding machine learning overlays.

NOTE – An IMT-2020 network is an example of a machine learning underlay network.

3.1.10 resource management [b-ITU-T Y.3520]: The most efficient and effective way to access, control, manage, deploy, schedule and bind resources when they are provided by service providers and requested by customers.

3.2 Terms defined in this Recommendation

None.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

AI Artificial Intelligence

C Collector

D Distributor

IMT International Mobile Telecommunications

IoT Internet of Things

M Model

ML Machine Learning

MLFO Machine Learning Function Orchestrator

NF Network Function

P Prediction

PP Preprocessor

QoS Quality of Service

RCA Root Cause Analysis

SDN Software-Defined Networking

SFC Service Function Chaining

VNF Virtualized Network Function

5 Conventions

None.

6 Introduction

The control of networks involves the following two groups of functions: design and deployment (i.e., design-time procedures); and operation and management (i.e., run-time procedures). The design-time procedures require processing of a lot of data related to service requirements obtained from customers to design and configure a network that can meet all service requirements. Similarly, run-time procedures also require processing of a lot of data related to performance, resource utilization and workload obtained from system operation measurement to infer the resource usage pattern or detect any fault and decide to carry out appropriate resources adjustment or fault recovery actions. The volume of data to be processed in the design and run-time procedures is so huge that it would be difficult for humans to process it quickly and make a timely decision. This Recommendation specifies the architectural framework for run-time procedures in future networks including IMT-2020 [ITU-T Y.3102], while the architectural framework for design-time procedures lies outside of scope of this Recommendation. The Recommendation provides a supportive framework to realize use cases and their potential requirements specified in [b-ITU-T Y-Suppl.55].

The architectural framework of network automation for resource and fault management in future networks including IMT-2020 leverages AI including ML (AI/ML) techniques to process the huge volume of related data, infer or predict system behaviour, and make prompt and highly accurate decisions for executing appropriate control operations. The major technical objective of resource and fault management is to utilize the available resources as optimally as possible while maintaining the QoS of the services provided to customers despite changing network conditions or occurrences of faults. Software-defined networking (SDN) [b-ITU-T Y.3300], network virtualization [b-ITU-T Y.3011], service function chaining (SFC) [b-ITU-T Y-Suppl.41] and orchestration [b-ITU-T Y.3100] techniques play supporting roles in the execution of resource and fault management actions and enable agile control of the network slices [b-ITU-T Y.3100] according to the management and orchestration framework specified in [ITU-T Y.3111] and [ITU-T Y.3112]. The automation of network slice resources adjusted for dynamic scaling is possible by extending the SDN controller and orchestrator with AI/ML.

In clause 7, a high-level architecture for the automation of resource and fault management by AI/ML is described. The resource and fault management functions described in clauses 8 and 9 use AI/ML to: collect and analyse network measurement data related to performance, resource utilization and workload; infer their patterns or trends; identify faults by root cause analysis (RCA); prescribe suitable fault recovery mechanisms; predict the future demands of resources or possibility of faults in various network nodes; and based on the inference, take appropriate actions for fault recovery and resource adaptation.

7 High-level architecture

This clause describes the high-level architecture of resource and fault management with AI/ML based on the high-level architecture for ML in future networks including IMT-2020 specified in [ITU-T Y.3172].

Figure 1 shows the high-level architecture for resource and fault management with AI/ML. It is composed of four subsystems: the management subsystem; AI/ML sandbox; AI/ML pipeline; and AI/ML underlay network. The management subsystem includes three management functions: resource management; fault management; and other management functions. The machine learning function orchestrator (MLFO) takes ML intents as input.

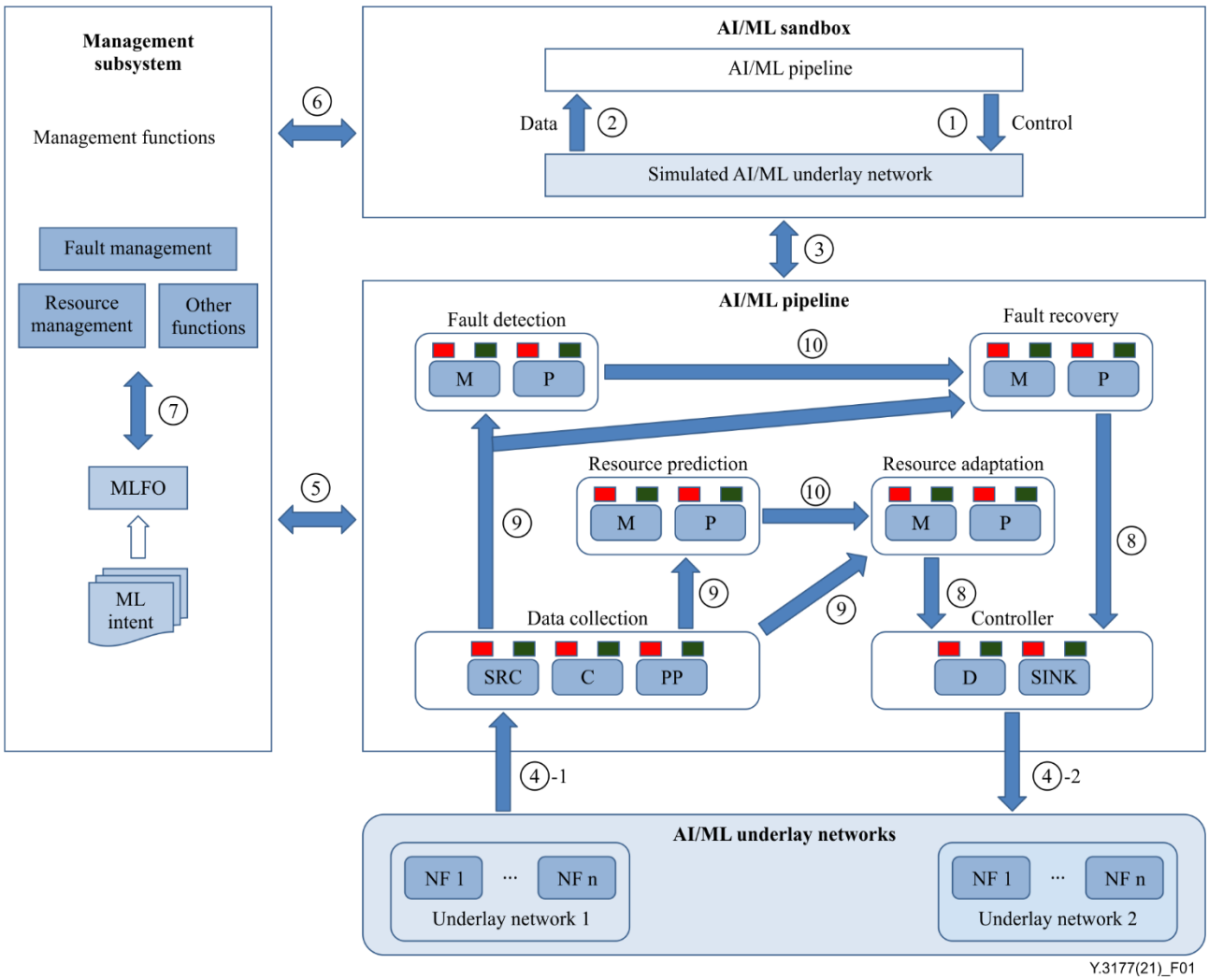

Figure 1 – High-level architecture of resource and fault management

The reference points shown in Figure 1 are similar to those specified in [ITU-T Y.3172]. These are briefly described as follows.

- (1) and (2): These are data-handling reference points between simulated underlay networks and an AI/ML pipeline in the AI/ML sandbox subsystem. (1) Is used to transfer control commands or requests from the AI/ML sandbox to the simulated AI/ML underlay network to send requests for obtaining monitoring data or inject simulated network faults, and (2) is used to transfer monitoring data in the reverse direction to collect data related to AI/ML simulated underlay network performance, resource utilization and faults.
- (3): This is the reference point between an AI/ML sandbox subsystem and an AI/ML pipeline subsystem.
- (4)-1 and (4)-2: These reference points are similar to (1) and (2), except that they interact with the production-level live underlay network, e.g., IMT-2020 networks. (4)-1 Carries monitoring data related to topology, alarms, performances and resource utilization, while (4)-2 carries control commands or requests.

  NOTE – These reference points represent the data-handling reference points as indicated in [ITU-T Y.3172].
- (5) and (6): These are reference points between the management subsystem, and the AI/ML pipeline subsystem and AI/ML sandbox subsystem, respectively. They are the same as those specified in [ITU-T Y.3172].

- (7): This is the reference point between the MLFO and other management and orchestration functions of the management subsystem.
- (8), (9) and (10): These are reference points between AI/ML pipeline nodes located in different levels and functions. (8) Is used to transfer control decisions from resource adaptation functions and fault recovery functions to the controller. (9) Is used to transfer control data from the data collection functions to ML models of resource prediction, resource adaptation, fault detection and fault recovery functions, and (10) is used to transfer one ML model output of the AI/ML pipeline to another ML model from the resource prediction functions to the resource adaptation functions and from the fault detection functions to fault recovery functions.

This Recommendation is based on several specific functions and capabilities of the AI/ML sandbox, AI/ML pipeline, and AI/ML underlay networks that can host and run virtualized network functions (VNFs) [b-ITU-T Y.3515]. The functions of resource and fault management functions are described in subsequent clauses.

As clarified in [ITU T Y.3172], the AI/ML pipeline is a logical pipeline of AI/ML nodes that are superimposed on various network functions (NFs). The services of the MLFO are used for instantiation and setup of these AI/ML pipeline nodes.

The AI/ML pipeline shown in Figure 1 consists of six functional groups: data collection; fault detection; fault recovery; resource prediction; resource adaptation; and controller. These functional groups are described as follows, together with MLFO.

- Data collection:

  Data collection functions, supported by the SRC, collector (C) and preprocessor (PP) nodes, collect system data including performance, resource utilization, alarms, fault reports and workload through (4)-1 reference points from the underlay network. The SRC and C nodes can exploit data-handling reference points specified in other related standards such as [ITU-T Y.3111]. The PP node can exploit various AI/ML techniques for extracting important features or removing noise in the data through regression, classification or clustering.

- Fault detection:

  Fault detection functions, supported by the model (M) and prediction (P) nodes, detect network faults using monitoring data obtained from the data collection function through the reference point (9). The M node analyses data to obtain fault information. The P node confirms the cause of fault and notifies the fault to the fault recovery functions.

- Fault recovery:

  The fault recovery functions, supported by the M and P nodes, recover from a fault through the selection of the most appropriate recovery actions, when the fault notification is received from the fault detection functions through reference point (10).

- Resource prediction:

  The resource prediction functions, supported by the M and P nodes, analyse network measurement data, infer their patterns or trends and predict the future demands on resources by the different network nodes and functions.

- Resource adaptation:

  The resource adaptation functions, supported by the M and P nodes, decide an appropriate function of resource adaptation among the resource arbitration, NF migration and network slice reconfiguration. These functions are described in detail in subsequent clauses.

- Controller:

  The controller functions, supported by the distributor (D) and sink nodes, are implemented through reference point (4)-2. As specified in [ITU-T Y.3172], a D node is responsible for

identifying the relevant sink(s) and distributing the output of M and P nodes to the corresponding sink nodes. Although the sink node is shown completely separated from the underlay network, it may exist inside the underlay network to implement the AI/ML action decisions on the AI/ML underlay network.

- MLFO:

  MLFO manages and orchestrates (e.g., instantiates) the AI/ML nodes needed in the AI/ML sandbox subsystem and in the AI/ML pipeline subsystem based on the ML intent (e.g., features and model type). It consists of three phases: training; evaluation; and deployment. In the training phase, a model is instantiated in the sandbox using the dataset based on the intent. In the evaluation phase, the trained model is verified to check whether performance meets the intent requirements. Finally, in the deployment phase, the model is deployed on the AI/ML pipeline at the time specified in the intent.

8 Resource management with AI/ML

To leverage AI/ML for the automation of network slice run-time operation and management, this clause describes resource management with AI/ML to achieve agile control of network resources allocated to network slices consisting of network service function chains [b-ITU-T Y-Suppl.41]. Figure 2 shows the resource management-related AI/ML pipeline functions, which are data collection, resource prediction, resource adaptation and controller.

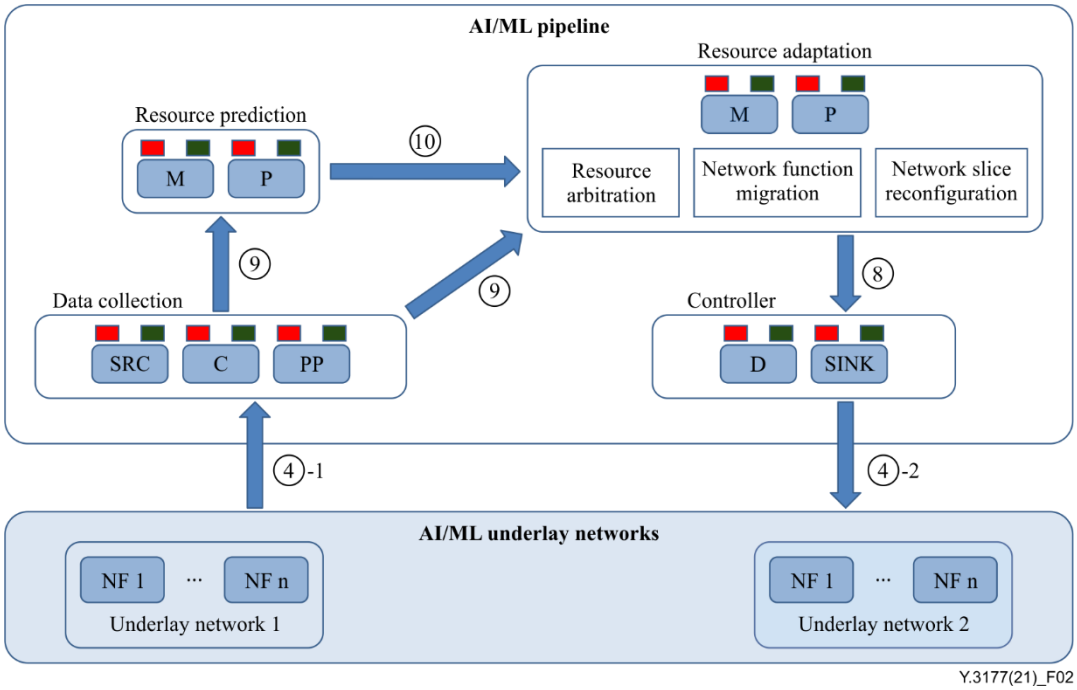

Figure 2 – Resource management with AI/ML

The data collection functions receive performance measurements and monitoring data from the AI/ML underlay network through reference point (4)-1. They pre-process the data and provide the processed data in specified formats to the resource prediction functions through reference point (9).

The resource prediction functions analyse the data by using AI/ML models to determine the current status of the underlay network and predict the future status of resource usage and performance levels. The results of resource prediction functions are fed through reference point (10) to the resource adaptation functions, which again process the data by AI/ML models and apply policy to the results to determine appropriate action for resource adaptation.

The resource adaptation functions can also receive the data required from data collection functions through reference point (9). The resource adaptation decisions can be one among resource arbitration, NF migration and network slice reconfiguration. Resource adaptation decisions are provided to the controller functions through reference point (8). The controller functions execute the resource adaptation decision on the AI/ML underlay network.

8.1 Resource prediction

Resource prediction functions take performance metrics, workload and resource usage monitoring data related to each of the underlay network systems (each consisting of a series of NFs) as input data from the data collection functions. They obtain data from the data collection functions through reference point (9), process the data by using AI/ML models, and infer the operational conditions of the AI/ML underlay network. The inference results can generate patterns or trends of network workload, resource utilization and performance. Depending on the type of AI/ML model used, resource prediction functions can provide results in the form of data clustering, classification, regression and prediction. To enable AI/ML models to make very accurate predictions, the AI/ML models are trained in the AI/ML sandbox by using data acquired from the AI/ML underlay network online or offline. If the AI/ML models are to be trained online by data generated in real time, the volume of data or the AI/ML models are required to be chosen in such a way that the data can be processed very quickly so that real time and agile control of the system becomes possible. Resource predictions are sent to resource adaptation functions through reference point (10).

8.2 Resource adaptation

Resource adaptation functions represent multi-stage resource control decision functions. They come to an appropriate decision among the three possible actions of resource arbitration, NF migration and network slice reconfiguration. AI/ML techniques are used to make intelligent decisions about the most appropriate action. Resource adaptation functions process the analysis results coming from resource prediction functions through reference point (10). They also access data from reference point (9), which includes data related to the service and performance requirements, and current status of resource allocation and utilization, to be used for more intelligent decisions of resource adaptation.

Resource adaptation decisions can be executed in two ways: reactively and proactively. In a reactive approach, resource adaptation functions only need to determine the required amount of resources to be allocated in the next timeslot based on an analysis of control data of the current timeslot. On the other hand, in a proactive approach, resource adaptation functions predict resource usage of future timeslots and adjust the resource before those timeslots begin. The policy selects a reactive or proactive approach so as to make the resource control task agile and efficient by preventing instability in the system due to an unnecessarily high frequency of resource adjustments.

The output of the resource adaptation functions is given to the controller functions through reference point (8). The three actions of resource arbitration, NF migration and network slice reconstruction are described in clauses 8.2.1 to 8.2.3. To reduce the overheads of the controller functions and avoid possible service interruptions, only one of these actions is recommended for execution at a time. The resource arbitration control function involves only one physical node; the NF migration involves two physical nodes; and the network slice reconfiguration may involve many physical nodes. Resource adaptation decisions are carried out in order from the highest to lowest priority order of resource arbitration, NF migration and network slice reconfiguration.

8.2.1 Resource arbitration

The resource arbitration function allocates limited available resources among many network slices (i.e., underlay network system consisting of a series of NFs) so that the service requirements of each network slice can be optimally satisfied despite fluctuation in the workload. The resource arbitration function determines the priority of a network slice over others for resource allocation according to policies, e.g., the importance and requirements of the services. When there are many slices providing

different levels of QoS to different application services by using the virtualized resources from the same node, the resource arbitration process becomes quite complex because many parameters (e.g., QoS requirements, allocated amount of resources, used amount of resources, as well as current and future workloads) of each slice need to be considered. In this case, AI/ML helps in making intelligent decisions to select or group appropriate candidate slices and nodes from where the resources can be taken and added to a high priority slice that urgently needs more resources.

Figure 3 depicts the resource arbitration procedure, which is executed in the virtualized AI/ML underlay network. There are two slices (network slice 1 and network slice 2) shown by dashed boxes, being deployed over the same resources (node A and node B). Network slice 1 has NF 1-a and NF 1-b, which are embedded in underlay node A and B, respectively. Similarly, network slice 2 has NF 2-a and NF 2-b, which are also embedded in underlay node A and B, respectively. If network slice 1 is for delivery of a high priority service, e.g., a disaster rescue support service created in an emergency situation, and network slice 2 is for a best effort service, e.g., a regular entertainment video service, the former would get a higher resource allocation priority than the latter. So, in Figure 3, when network slice 1 demands more resources (e.g., due to a sudden increase in user population or network traffic), the resources allocated to it from node A are increased by α, while decreasing the same amount of resources allocated to network slice 2 from the same node. Similarly, resources allocated to NF 1-b of network slice 1from node B are also increased by β, while resources allocated from the same node to NF 2-b of network slice 2are decreased by β.

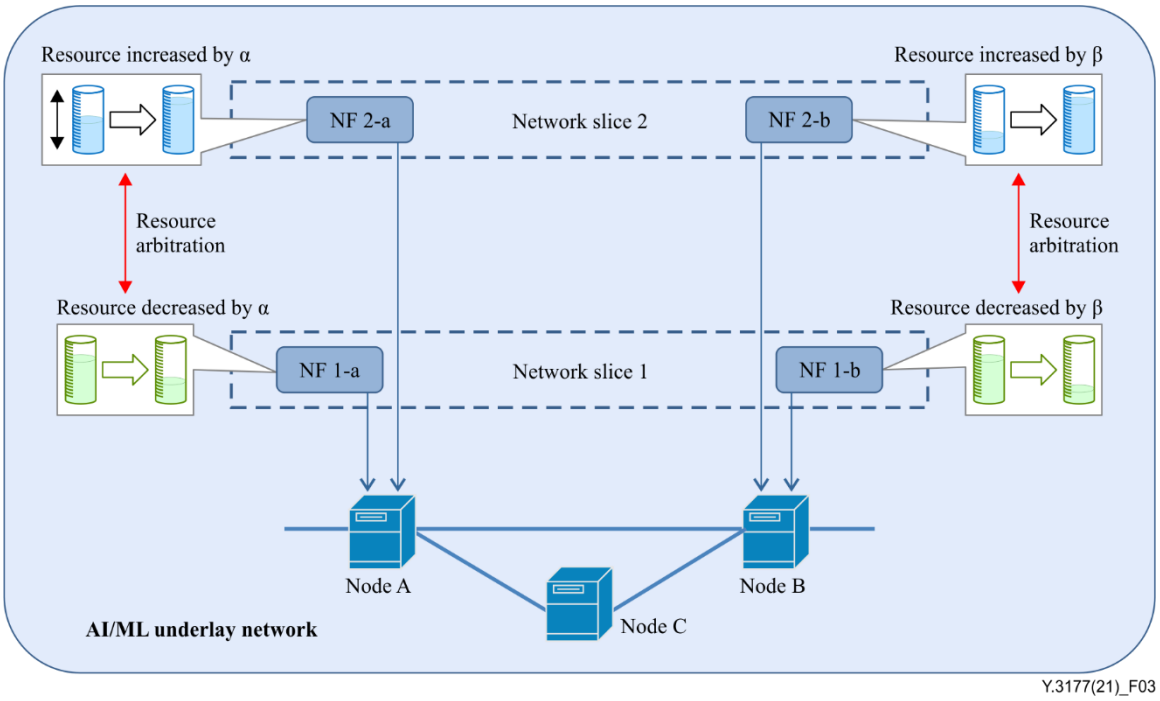

Figure 3 – Resource arbitration

8.2.2 Network function migration

If the resource arbitration function cannot satisfy the resource requirements of a network slice due to limited available resources in the physical node currently hosting the NF, the NF is migrated to the other node that already lies in the same network slice and SFC path. When the migration process is required to be made agile and efficient as well as stable for a longer time (to reduce the management and operation cost), there are many issues to be considered simultaneously. There can be many candidate target nodes, where the NF migration is possible at a given point of time, but the available resource pattern may keep changing as the traffic flows in other slices also fluctuate with time. Solving this problem by conventional methods, such as integer linear programming, in a short time

is too hard due to high complexity. Thus, AI/ML-based approaches can be helpful for making an intelligent decision about an NF migration process.

Figure 4 depicts an NF migration scenario. There is only network slice 1, which has two functions NF 1-a and NF 2-b, deployed over node A and node B, respectively. NF 1-a uses an amount of resources a_1 while NF 1-b uses an amount of resources b_1. Due to changes in user population or traffic patterns, assume that NF 1-a requires additional resources (say of amount α) that are not available in node A. In this case, the network function migration is activated to move NF 1-a from node A to another node (e.g., node B) that has enough available resources and is already included in the same slice and SFC path. Migrating the NF along the same SFC path or network slice does not invoke any configuration changes in the physical resource level (e.g., lower layer routing), thus the NF migration process can be faster, agile and not disruptive to the service. After NF migration, resource usage in node A by network slice 1 would be zero while the resource usage in node B by both NF 1-a and NF 1-b would be $a_1 + \alpha + b_1$.

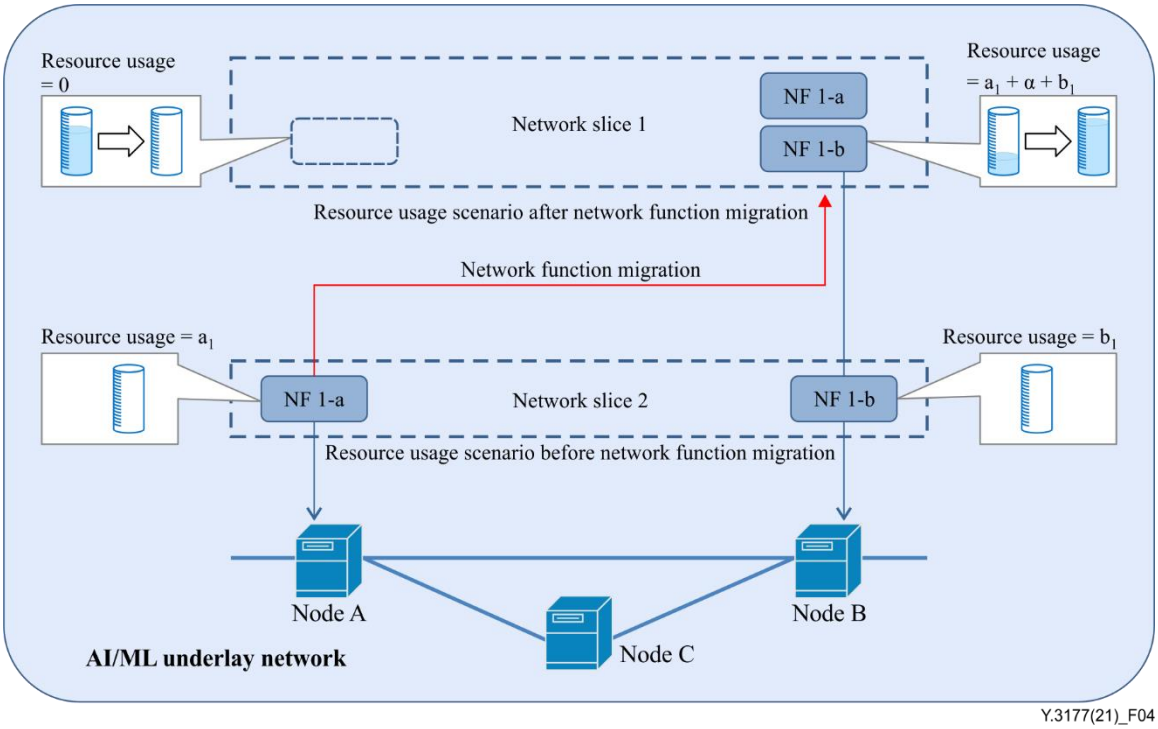

Figure 4 – Network function migration scenario

8.2.3 Network slice reconfiguration

If the migration function described in clause 8.2.2 cannot meet the resource requirements of the network slice, the slice is reconfigured by including new nodes that can contribute enough resources to the reconfigured slice. The network slice reconfiguration function takes a longer time to execute than the previous two functions because it requires interactions with the underlying resource controllers, as well as reconfiguration of the SFC path and underlying routing topology. The use of AI/ML models can optimize the process of new node selection during a network slice reconfiguration, so that frequent reconfiguration can be avoided. This can reduce operation and management cost, as well as cases of service disruption due to the time taken by the reconfiguration process.

Figure 5 depicts a scenario of network slice reconfiguration. NF 1-a of network slice 1 requires additional resources that are not available in any of the nodes lying along the SFC path (e.g., node A and node B) of the network slice. In this case, network slice 1 is reconfigured as network slice 1′ whose NF 1-a is deployed in a new node (e.g., node C), which has sufficient resources.

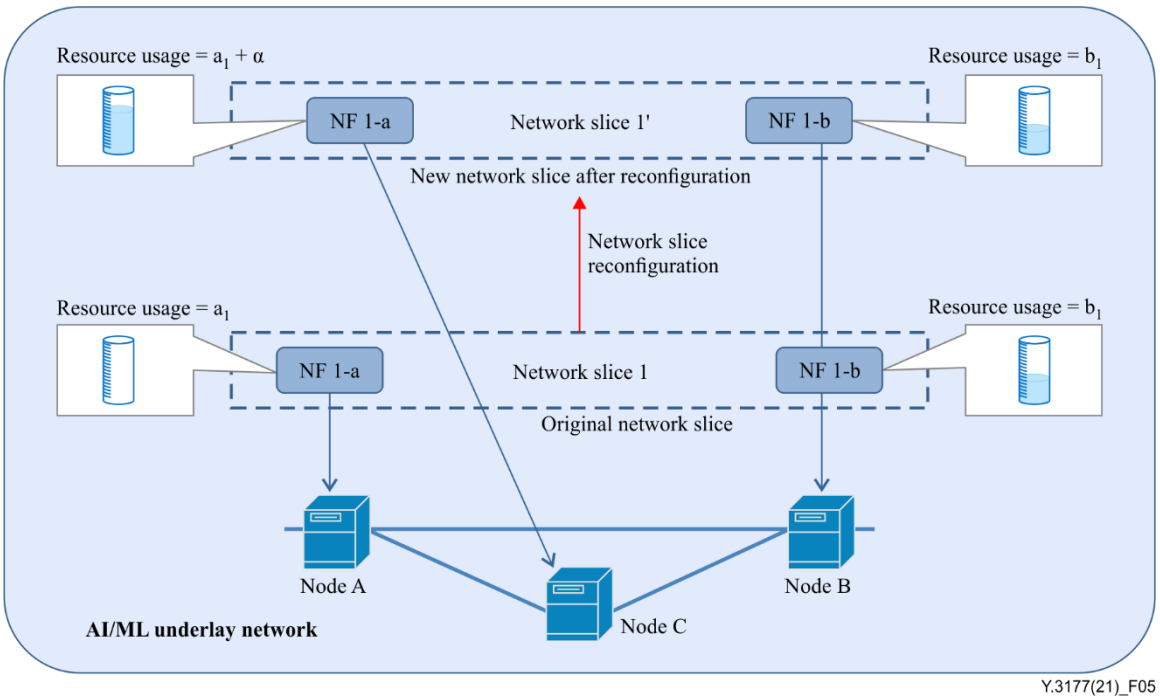

Figure 5 – Network slice reconfiguration scenario

9 Fault management with AI/ML

This clause describes fault management with AI/ML to automate network operations. Figure 6 shows fault management with AI/ML consisting of detection- and recovery-related functions.

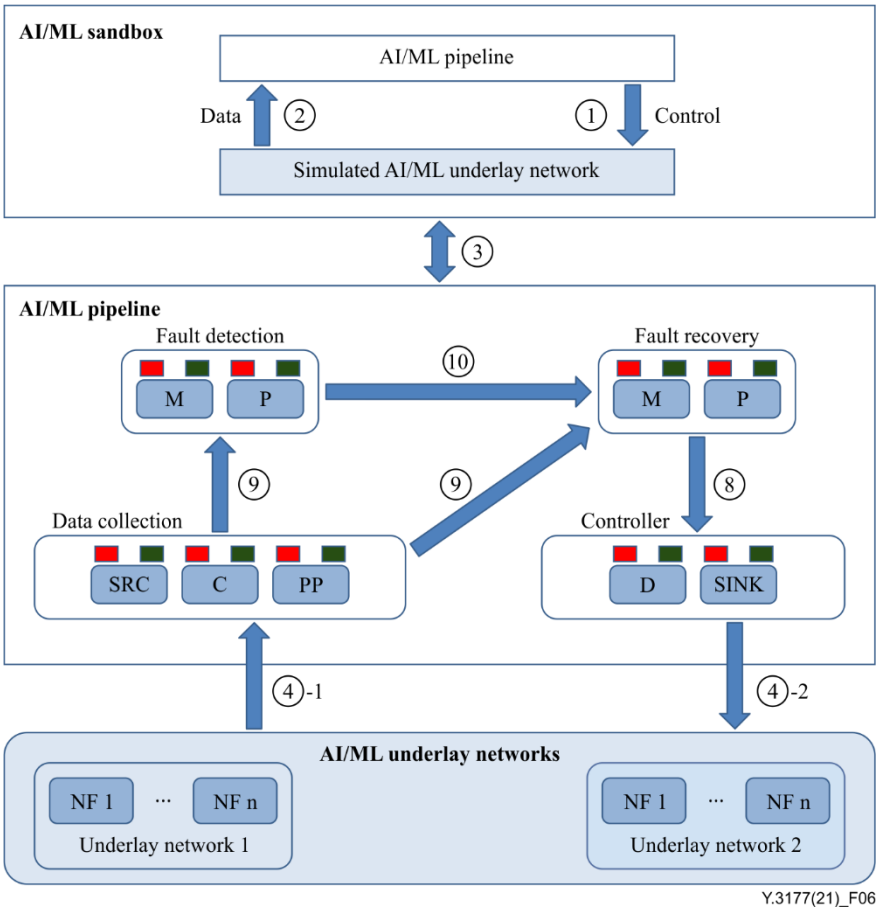

Figure 6 – Fault management with AI/ML

Fault management functions are executed into phases: fault detection and fault recovery. In the fault detection phase, the data collection functions receive and store operational data from the underlay network through reference point (4)-1. The fault detection functions using AI/ML models receive the stored data through reference point (9), and analyse it to find abnormal behaviours and identify the root causes of fault. In the fault recovery phase, fault recovery functions using AI/ML models derive operational data stored in the data collection functions through reference point (9), as well as the output coming from the fault detection function through reference point (10) and determines the fault recovery workflows. After that, the outputs of the workflows are sent to the action controller through reference point (8) as a control request to execute the fault recovery workflows through reference point (4)-2. The reference points shown in Figure 6 are described in clause 7.

9.1 Fault detection with AI/ML

This clause describes the functions and workflow of fault detection with AI/ML as shown in Figure 7. The fault detection has two phases: training and operating. It is not possible to collect a substantial representative training dataset from live underlay networks in a short time period because the faults do not occur frequently in live underlay networks. In order to address this issue, a simulated underlay network is utilized having the same characteristics as the live underlay network.

The simulated network is capable of intentionally injecting fault patterns according to the instructions given by the fault recipe executor (step 1) and promptly generating enough training data in the training phase. The abnormal behaviour data generated from the simulated network elements is collected through telemetry (step 2), and the dataset is stored with a fault type label provided by the fault recipe executor (step 3). In order to detect and analyse several types of fault, the procedure of these three steps is automatically repeated multiple times until a sufficient training dataset is obtained. Subsequently, the trained models for fault detection and RCA are created by using the training dataset and training functions (step 4). After training is completed, the trained model is transferred to the AI/ML pipeline in the live network in the operation phase.

In the operating phase, the fault detection function using trained models analyses data coming from the live underlay network through the data collection functions, and detects an abnormal behaviour and notifies the RCA function. Once the notification is received, the RCA function using trained models performs its analysis with the data provided by the data collection function, as well as the fault detection function, and derives the prospective root cause of the fault. The analysis result is sent to other functions or a human operator for further recovery actions. It may be possible to retrain and update the models with the data obtained from the live underlay network to maintain and enhance the quality of the models.

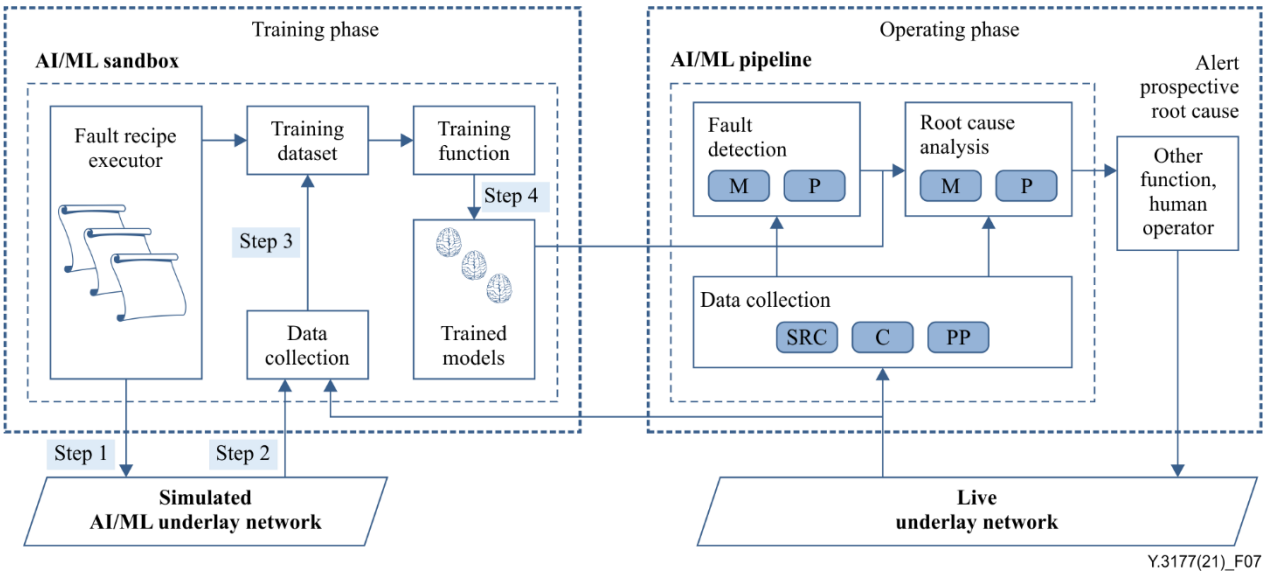

Figure 7 – Functions and workflow of fault detection with AI/ML

9.2 Fault recovery with AI/ML

This clause describes the functions and workflow of fault recovery with AI/ML as shown in Figure 8. Like the fault detection mechanism, the fault recovery mechanism also consists of two-phases: training and operating. In the case of fault recovery, it requires a lot of time and effort to gather enough training data and create suitable models for fault recovery actions for each type of fault, therefore, the fault recovery mechanism adopts dynamic and incremental learning models, such as reinforcement learning algorithms. Since it is not acceptable to impose fault models in the live networks, it also uses a simulated underlay network, which has the same characteristics as the live one. In the training phase, firstly, one of the fault recipes is executed to cause a fault (step 1). Next, the data collection function collects the data from the simulated underlay network (e.g., network performance, state and topology) through telemetry (step 2). After that, the training function receives the dataset, determines recovery actions as a workflow and sends it to the fault recovery recipe executor (step 3). Then, it performs the fault recovery workflow in the simulated underlay network (step 4). The trained models for the fault recovery workflow engine are created by repeating steps 1 to 4 until training is completed. The trained model is transferred to the AI/ML pipeline in the live network in the operating phase.

In the operating phase, when a fault occurs in the live underlay network, the RCA function sends the analysis result to the fault recovery workflow engine. Then, the recovery workflow engine function, using trained models, analyses the data obtained from the data collection function and RCA (e.g., network performance, state, topology and root cause), and notifies a human operator with recommended recovery actions for the fault. It may be possible to retrain and update the models with the data obtained from the live underlay network to maintain and enhance the quality of the models.

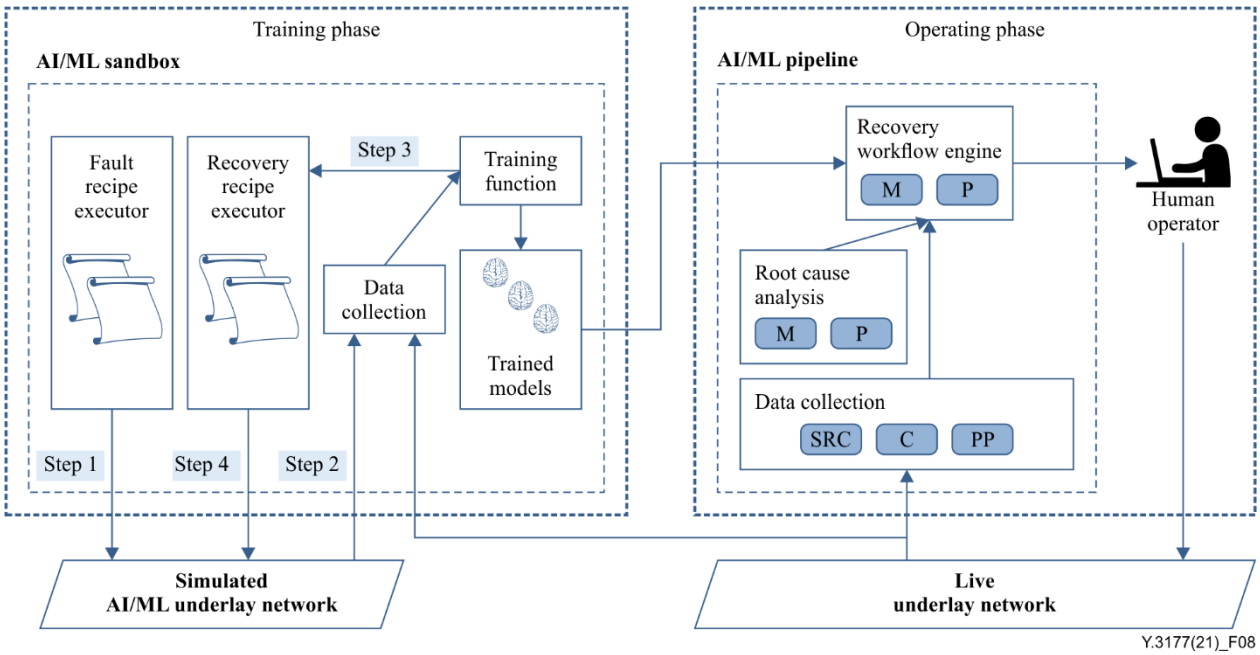

Figure 8 – Functions and workflow of fault recovery with AI/ML

10 Security considerations

This Recommendation describes a network automation architectural framework for resource and fault management with AI/ML on the basis of the high-level architecture of ML in future networks including IMT-2020 [ITU-T Y.3172]. Therefore, the security considerations of [ITU-T Y.3172] are also valid for this Recommendation.

It is required to have appropriate security measures in data collection, resource prediction and resource adaptation functions. All interfaces to collect monitoring data and transmit resource adaptation control decisions are required to have proper security measures.

The AI/ML models and pipeline functions are also required to possess appropriate protection measures so that their operations cannot be manipulated maliciously [ITU-T Y.3172]. It is required to add monitoring points to each step or process in the AI/ML pipeline so that any abnormal behaviour of functions can be easily detected and corrected.

Bibliography

[b-ITU-T X.790] Recommendation ITU-T X.790 (1995), *Trouble management function for ITU-T applications.*

[b-ITU-T Y.3011] Recommendation ITU-T Y.3011 (2012), *Framework of network virtualization for future networks*.

[b-ITU-T Y.3100] Recommendation ITU-T Y.3100 (2017), *Terms and definitions for IMT-2020 network*.

[b-ITU-T Y.3300] Recommendation ITU-T Y.3300 (2014), *Framework of software-defined networking*.

[b-ITU-T Y.3515] Recommendation ITU-T Y.3515 (2017), *Cloud computing – Functional architecture of network as a service*.

[b-ITU-T Y.3520] Recommendation ITU-T Y.3520 (2015), *Cloud computing framework for end to end resource management*.

[b-ITU-T Y-Suppl.41] ITU-T Y-series Recommendations – Supplement 41 (2016), *ITU-T Y.2200-series – Deployment models of service function chaining*.

[b-ITU-T Y-Suppl.55] ITU-T Y-series Recommendations – Supplement 55 (2019), *ITU-T Y.3170-series – Machine learning in future networks including IMT-2020: Use cases*.[b-ETSI GR ENI 004] Group Report ETSI GR ENI 004 V1.1.1 (2018), *Experiential networked intelligence (ENI); Terminology for main concepts in ENI.*

SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series D	Tariff and accounting principles and international telecommunication/ICT economic and policy issues
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	Transmission systems and media, digital systems and networks
Series H	Audiovisual and multimedia systems
Series I	Integrated services digital network
Series J	Cable networks and transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Environment and ICTs, climate change, e-waste, energy efficiency; construction, installation and protection of cables and other elements of outside plant
Series M	Telecommunication management, including TMN and network maintenance
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Telephone transmission quality, telephone installations, local line networks
Series Q	Switching and signalling, and associated measurements and tests
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	Data networks, open system communications and security
Series Y	**Global information infrastructure, Internet protocol aspects, next-generation networks, Internet of Things and smart cities**
Series Z	Languages and general software aspects for telecommunication systems

Printed in Switzerland
Geneva, 2021