Draft new Recommendation ITU-T Y.3819 (ex.Y.QKDN-amc)

Quantum key distribution networks – Requirements and architectural model for autonomic management and control enablement

Summary

As the number and diversity of devices and other resources that make up the individual QKDNs continue to grow, automating QKDN control and management tasks becomes ever-more important to avoid the untimely actions and improve the quality of service (QoS). To cope with the challenges of QKDN control and management, while minimizing human intervention towards full automation of QKDN services, this draft Recommendation specifies requirements and a possible architectural model for autonomic management and control (AMC)enabled QKDN (QKDNamc) including an overview, requirements, considerations for cognition process, an architectural model, and example procedures.

Keywords

Autonomic management and control (AMC), quantum key distribution (QKD), QKD network (QKDN)

Table of Contents

Draft new Recommendation ITU-T Y.3819 (ex.Y.QKDN-amc)

Quantum key distribution networks – Requirements and architectural model for autonomic management and control enablement

1 Scope

This Recommendation specifies one possible set of functional requirements and a possible architectural model for autonomic management and control (AMC)-enabled QKDN (QKDNamc). In particular, the scope of this Recommendation includes:

- Overview of QKDNamc;
- Requirements for QKDNamc;
- Consideration for cognition process of QKDNamc;
- Architectural model for QKDNamc;
- Example operational procedures of QKDNamc.

2 References

[ITU-T Y.3324] Recommendation ITU-T Y.3324(2018), *Requirements and architectural framework for autonomic management and control of IMT-2020 networks*.

[ITU-T Y.3800] Recommendation ITU-T Y.3800 (2019), *Framework for Networks to support Quantum Key Distribution*.

[ITU-T Y.3801] Recommendation ITU-T Y.3801 (2020), *Functional requirements for quantum key distribution networks*.

[ITU-T Y.3814] Recommendation ITU-T Y.3814 (2023), *Functional requirements and architecture for machine learning enablement.*

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1 **autonomic management and control (AMC)** [ITU-T Y.3324]: A behaviour or action which is determined in a reactive or proactive manner based on the external stimuli (environment aspects) as well as the goals they are required to fulfil, principles of operation, capabilities, experience and knowledge.

NOTE – In the case of software defined networks, this definition means that AMC has the ability to dynamically select the network's configuration, control and manage the network, through self-management functionality that reaches optimal decisions, taking into account the context of operation (environment requirements and characteristics), goals and policies (corresponding to principles of operation), profiles (corresponding to capabilities i.e. functional features supported), and machine learning (for managing and exploiting knowledge and experience.

3.1.2 **key manager (KM)** [ITU-T Y.3800]: A functional module located in a quantum key distribution (QKD) node to perform key management in the key management layer.

3.1.3 **quantum key distribution (QKD)** [b-ETSI GR QKD 007]: Procedure or method for generating and distributing symmetrical cryptographic keys with information theoretical security based on quantum information theory.

3.1.4 **quantum key distribution link (QKD link)** [ITU-T Y.3800]: A communication link between two quantum key distribution (QKD) modules to operate the QKD.

NOTE – A QKD link consists of a quantum channel for the transmission of quantum signals, and a classical channel used to exchange information for synchronization and key distillation.

3.1.5 **quantum key distribution module (QKD module)** [ITU-T Y.3800]: A set of hardware and software components that implements cryptographic functions and quantum optical processes, including quantum key distribution (QKD) protocols, synchronization, distillation for key generation, and is contained within a defined cryptographic boundary.

NOTE – A QKD module is connected to a QKD link, acting as an endpoint module in which a key is generated. These are two types of QKD modules, namely, the transmitters (QKD-Tx) and the receivers (QKD-Rx).

3.1.6 **quantum key distribution network (QKDN)** [ITU-T Y.3800]: A network comprised of two or more quantum key distribution (QKD) nodes connected through QKD links.

NOTE – A QKDN allows sharing keys between the QKD nodes by key relay when they are not directly connected by a QKD link.

3.1.7 **quantum key distribution network controller (QKDN controller)** [ITU-T Y.3800]: A functional module, which is located in a quantum key distribution (QKD) network control layer to control a QKD network.

3.1.8 **quantum key distribution network manager (QKDN manager)** [ITU-T Y.3800]: A functional module, which is located in a quantum key distribution (QKD) network management layer to monitor and manage a QKD network.

3.1.9 **quantum key distribution node (QKD node)** [ITU-T Y.3800]: A node that contains one or more quantum key distribution (QKD) modules protected against intrusion and attacks by unauthorized parties.

NOTE – A QKD node can contain a key manager (KM).

3.1.10 **quality of service (QoS)** [b-ITU-T P.10]: The totality of characteristics of a telecommunications service that bear on its ability to satisfy stated and implied needs of the user of the service (see [b-ITU-T E.800]).

3.1.11 **Machine learning-enabled quantum key distribution network (ML-enabled QKDN)[ITU-T Y.3814]:** A quantum key distribution network (QKDN) that extends or enhances its functionalities enabled by machine learning (ML) capabilities to achieve different objectives.

NOTE1 – ML is an optional functionality for QKDN.

NOTE2 – Examples of different objectives are specified in [b-ITU-T Y-Sup-70].

3.2 Terms defined in this Recommendation

This chapter defines all the terms used in this recommendation.

3.2.1 **Autonomic management and control enabled quantum key distribution network (AMC-enabled QKDN):** A quantum key distribution network (QKDN) that extends or enhances its functionalities enabled by autonomic management and control (AMC) capabilities to achieve different objectives.

NOTE – AMC is an optional functionality for QKDN.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

AMC	Autonomic Management and Control
CL	Control Layer
CLMO	Cross Layer Management and Orchestration
DE	Decision-making Element
KM	Key Manager
KMA	Key Management Agent
KML	Key Management Layer
KML DE	Key Management Layer autonomic Decision-making Element
KSA	Key Supply Agent
ME	Management Entity
ML	Machine Learning
NE	Network Element
NFV	Network Function Virtualization
NP	Network Performance
OSNR	Optical Signal-to-Noise Ratio
QBER	Quantum Bit-Error Ratio
QL	Quantum Layer
QKD	Quantum Key Distribution
QKDN	QKD Network
QL DE	Quantum Layer autonomic Decision-making Element
SDN	Software-Defined Networking
SPD	Single Photon Detector

5 Conventions

In this Recommendation:

The keywords “is required to” indicate a requirement which must be strictly followed and from which no deviation is permitted if conformance to this document is to be claimed.

The keywords “is recommended” indicate a requirement which is recommended but which is not absolutely required. Thus, this requirement need not be present to claim conformance.

6 Overview of QKDNamc

As the number and diversity of devices that make up the individual QKDNs continue to grow, automating QKDN control and management tasks becomes ever-more important, so as to avoid the untimely actions and improve the QoS. AMC can enable the network to adjust to the varying network conditions and service demands in a timely and efficient manner without requiring human intervention. AMC in [ITU-T Y.3324] is targeted to IMT-2020 networks, but its idea is applicable. AMC is about DEs as autonomic functions (i.e. control-loops) with cognition. The component (software logic) that drives autonomics at a particular level of abstraction for self-* functionality is called a Decision-making Element (DE) and described in depth in [b-ETSI TS 103 195-2]. DEs are responsible for autonomic management and adaptive control of systems and network resources, parameters, and services. Cognition enhances DE logic and enables DEs to manage and handle even the unforeseen situations and events detected in the environment around the DE(s).

To cope with the challenges of QKDN control and management, while minimizing human intervention towards full automation of QKDN, this Recommendation specifies the requirements and architectural model for AMC in QKDNs including the overview, consideration for cognition process, requirements, and an architectural model.

7 Consideration for cognition process of QKDNamc

The cognition process of AMC in QKDN is based on a decision-making feedback loop of monitoring, learning, decision, and action sub-processes as shown in Figure 7.1. The consideration for each sub-process is as follows.

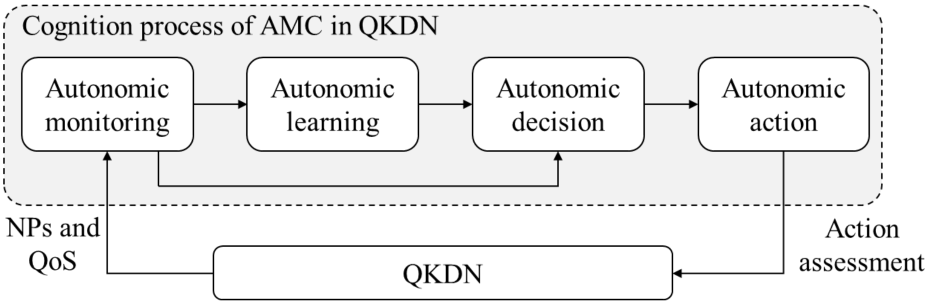

Figure 7.1 Cognition process of QKDNamc

- **Autonomic monitoring:** involves collecting and analyzing information about the network performances (NPs) and QoS of the QKDN, such as QKD modules status and key manager status, etc. The monitoring data can be used to detect changes in the QKDN that may require attention.
- **Autonomic learning:** involves acquiring and updating knowledge about the QKDN based on the monitored data and its environment, such as the behaviors of QKDN functional components. The learning data can be used to improve the understanding and prediction of the dynamic QKDN performances.
- **Autonomic decision:** involves selecting and planning the best course of actions to achieve the AMC goal in the QKDN. The decision data can be based on the monitoring and learning data, as well as on the predefined policies, rules, and objectives of the QKDN.
- **Autonomic action:** involves executing and evaluating the chosen action on the QKDN, so as to modify or optimize the QKDN. The action data can be used to assess the effectiveness and efficiency of the action, which will provide feedback for further monitoring and learning.

Cloud computing, SDN, NFV, and ML are core enablers of AMC. AMC requires the seamless intelligent decision-making feedback loop of the precise monitoring of status of managed resources, intelligent decision making and necessary policy generation based on the monitored information and open programmable enforcement of generated policies. Cloud computing provides an abundant resource pool for which complex autonomic decision-making processes are required. SDN provides open control capability of enforcing autonomic decision policies. NFV provides a virtual programmable execution environment that autonomic decision entities could run. Lastly, ML provides an intelligence for optimal decision making of the complex networking environment. The built-in cognitive management integrates ML as part of the workflows and operations to support intelligent operations.

The cognition process described in the Fig. 7.1 is the general one. Any autonomic closed-loop should follow the four general processes: monitoring, learning, decision, and action. However, these general processes can be further divided into more specific processes depending on the timing, priority, and action types. They are the processes of collect, normalize, compare, plan, decide, and act. Monitoring consists of collect and normalize sub-processes, Decision consists of compare, plan, and decide sub-processes. For example, when the cognition process of autonomic control and management is initiated for the first time, there are no learned or history information for autonomic control and management actions. In this case full cycle of monitoring, learning, decision, and action processes have to be executed. However, when learned or history information, decision, and/or actions exist, the cognition process can be executed more efficiently by skipping some processes, for example, learning, and/or decision-making processes. The fastest loop (urgent), thus, can be monitoring, compare, and action processes only when the monitored event is learned or known in priori. Another faster loop (high priority) can be monitoring, decision (compare and decide), and action when the learned or history information exist but a further decision-making process is necessary. The figure 7.2 illustrates the enhanced three types of autonomic management and control cognition process.

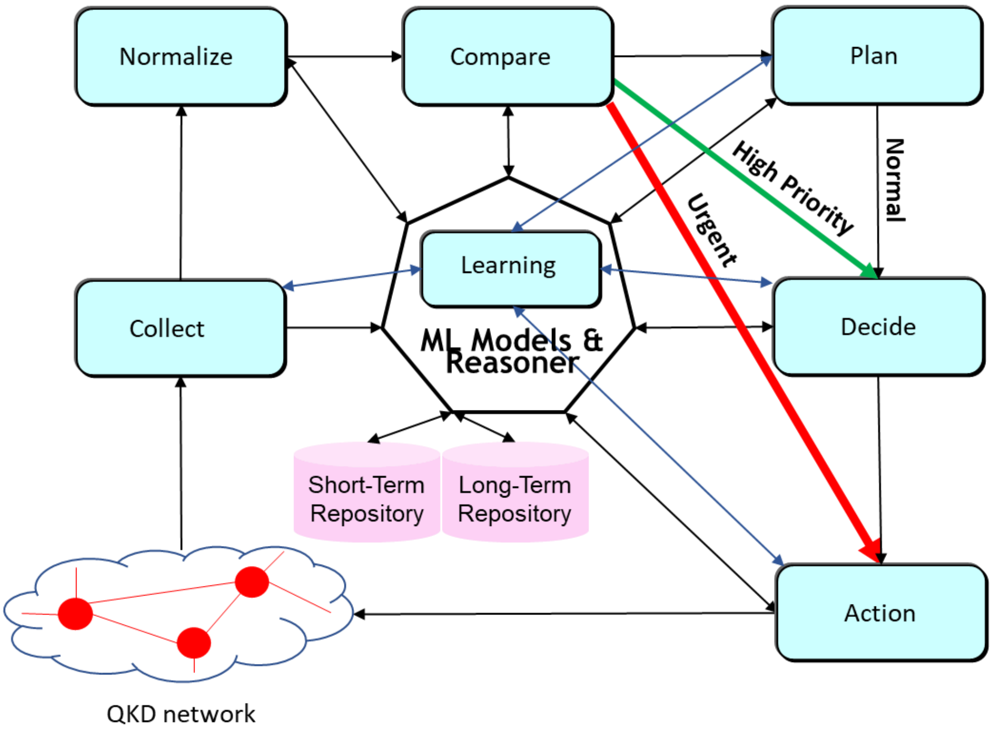

Figure 7.2 Enhanced cognition process of AMC in QKDN

8 Requirements for QKDNamc

8.1 High-level requirements for QKDNamc

The high-level requirements of QKDNamc are as follows.

- QKDNamc is recommended to support autonomic management capabilities including knowledge layer with cognitive management functionality.

  NOTE 1 – knowledge layer provides necessary functionality to support autonomic management of QKDN and services. The one of the main functions of the knowledge layer is a cognitive management process which is a control loop of monitoring, learning, decision and action sub-processes. Autonomic management decisions and associated actions are made through this process.

- QKDNamc is required to support scalability .

  NOTE 2 – AMC functionality should be scalable to be used in complex and large QKDN management and control environment.

- QKDNamc is required to support availability and reliability .
- QKDNamc is required to support real, near-real, and/or non-real time AMC decision making and operations.

  NOTE 3 – cognition process is required to support three modes of operations: urgent, high-priority, and normal to meet this requirement.

- QKDNamc is required to support interworking capability with the management functionality of QKDN.

  NOTE 4 – autonomic management should co-exist with other management functionality. It is a supporting functionality of the other management functionality.

- QKDNamc is required to support performance management functionality.
- QKDNamc is required to support coordination functionality among AMC functional capabilities.

8.2 Functional requirements for QKDNamc

Based on the QKDN control and management requirements specified in [ITU-T Y.3801] and the functional requirements of AMC specified in [ITU-T Y.3324], QKDNamc is required to meet the following functional requirements.

8.2.1 Functional requirements for QKDNamc autonomic monitoring

- QKDNamc is required to support autonomic collection/receipt of the status information from the quantum, key management, and control layers.
- QKDNamc is recommended to support autonomic semantic monitoring capability to reduce communication and computing overhead.
- QKDNamc is recommended to autonomically receive fault, performance, accounting and configuration information. .

8.2.2 Functional requirements for QKDNamc autonomic knowledge management

- QKDNamc is required to support capability of autonomic DE in the QKDN knowledge layer.

  NOTE1 – Autonomic DE in the QKDN knowledge layer is responsible for an entire QKDN context. Thus, a slow control loop operation is used in non-real time manner.

- QKDNamc is required to support capability of ML repository in the QKDN knowledge layer.

- QKDNamc is required to support capability of model-based information translation.
- QKDNamc required to support capability of cognitive management in the QKDN knowledge layer.
- QKDNamc is required to support capability of control layer autonomic DE in the QKDN management layer.

  NOTE2 – Control layer autonomic DE supports real-time or near real-time fast closed-loop operation.
- QKDNamc is required to support capability of key management layer autonomic DE in the QKDN management layer.

  NOTE3 – Key management layer autonomic DE supports real-time or near real-time fast closed-loop operation.
- QKDNamc is required to support capability of quantum layer autonomic DE in the QKDN management layer.

  NOTE4 – Quantum layer autonomic DE supports real-time or near real-time fast closed-loop operation.
- QKDNamc is required to support a reference point between the QKDN knowledge layer and QKDN ML layer.
- QKDNamc is required to support a reference point between the QKDN knowledge layer and QKDN management layer.

8.2.3 Functional requirements for QKDNamc autonomic configuration management

- QKDNamc is required to provide autonomic configuration control of QKD modules, QKD links, KMs and KM links.
- QKDNamc is required to provide autonomic configuration management of virtual and physical resource provisioning.
- QKDNamc is recommended to provide autonomic configuration management to support:
  - autonomic routing and rerouting of key relay if the QKDN supports key relay;
  - autonomic collecting and managing a network topology;
  - autonomic resource configuration for inventory management;
  - autonomic changing of managed resources based on the demand and availability;
  - autonomic discovering QKD managed resources in the managed QKDN.

8.2.4 Functional requirements for QKDNamc autonomic fault management

- QKDNamc is required to provide autonomic fault management to support autonomic analysis of the status information collected/received for fault indicators.
- QKDNamc is required to support autonomic diagnose of the known faults (e.g., traffic affected faults or non-traffic affected faults).
- QKDNamc is recommended to support autonomic healing, for example, based on autonomic location and autonomic correction of the root cause of known failures.

  NOTE – Since the target failure in this requirement is known in the past, autonomic location and autonomic correction of the same types of failure can be supported. Autonomic healing is the overall process to remedy such known failures based on these two functionalities (i.e., autonomic location and correction).
- QKDNamc is recommended to support autonomic protection from malicious attacks and unauthorized access.

8.2.5 Functional requirements for QKDNamc autonomic security management

- QKDNamc is recommended to autonomically provide security management to support auto-management of authentication and authorization.

8.2.6 Functional requirements for QKDNamc autonomic optimization

- QKDNamc is required to provide auto-routing control of key relay if the key relay function is supported based on the resource status and the service requirements;
- QKDNamc is recommended to provide autonomic QoS policy control;
- QKDNamc is recommended to provide autonomic QKDN resource optimization.;
- QKDNamc is recommended to provide optimal autonomic key supply.

8.2.7 Functional requirements for other QKDNamc

- QKDNamc is required to support interface with the management functionality of QKDN
- QKDNamc is required to support information model for the interface with the management functionality of QKDN.

9 Architectural model for QKDNamc

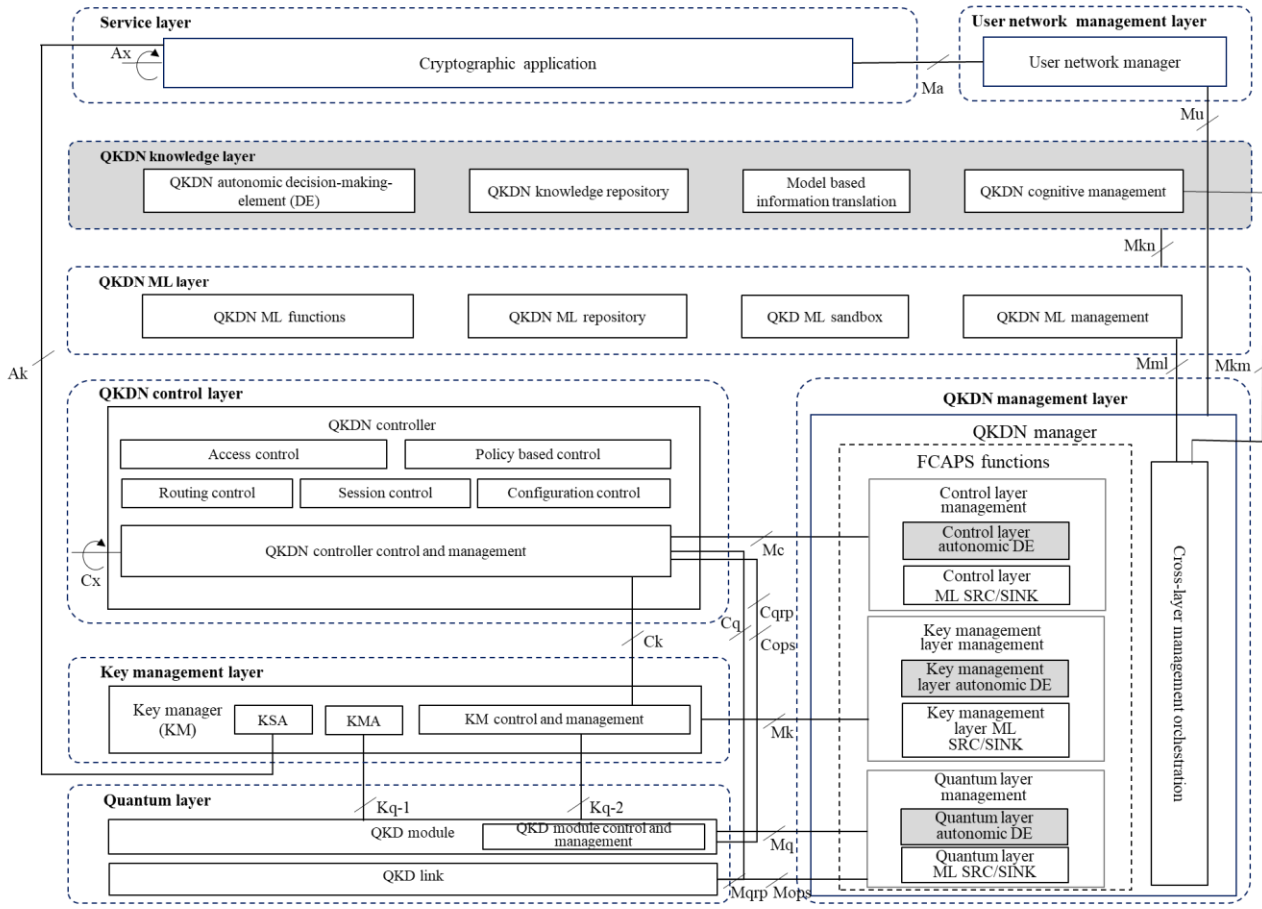

Figure 9.1. A possible architectural model for QKDNamc

The architectural model for QKDNamc is based on the architecture model of ML-enabled QKDN in [ITU-T Y.3814]. Autonomic DEs with closed control loops are the core elements for AMC implementation in QKDN.

9.1 General Autonomic DEs

Decision-making-Elements (DEs) are autonomic functions (i.e. control-loops) with cognition processes in the control and management plane. DEs realize self-* features (self-configuration, self-optimization, etc.) as a result of the decision-making behaviour of a DE that performs dynamic/adaptive control and management of its associated Managed Entities (MEs) and their configurable and controllable parameters. Such a DE can be embedded in a network node (Network Element (NE) in general) or higher at a specific layer of the outer overall network and services control and management architecture. An NE may be physical or virtualized (such as in the case of the NFV paradigm). From an architecture perspective, a control-loop can be based on a distributed model (for fast control loops). In this case the DE is embedded in the nodes (physical or virtualized). Whereas in a centralized model (for slow control-loops), the DE is embedded (implemented) outside of the network nodes. Both kinds of control-loops act towards a global goal to ensure a stable state of the network. A DE can negotiate with another DE to realize dynamic adaptation of network resources and parameters, or services, via reference points. This leads to the notion of global network autonomics, a result of interworking DEs as collaborative manager components that perform AMC of their associated MEs and their parameters [b-ETSI WP 16].

9.2 Functional elements for QKDNamc

There are two types of DEs including the QKDN DEs in network-level and the local DEs specific to different QKDN layers. The new QKDN knowledge layer is added between the QKDN ML layer and the QKDN management layer through the reference points Mkn and Mkm. It includes QKDN autonomic DEs, QKDN knowledge repository, model-based information translation and QKDN cognitive management. The local DEs are implemented in QKDN management layer specific to the quantum layer, key management layer and QKDN control layer, to make the faster closed-loop decisions based on the local AMC policies. In the architectural model, DEs in QKDN knowledge layer are logically centralized and act as slow control loops. DEs in the management layer specific to each control/key management/quantum layer, play fast control loops and negotiate one another to achieve global AMC goals. Figure 9.1

- **Quantum layer autonomic DE:** provides self-management capabilities of quantum modules and links. It supports real-time or near real-time closed-loop operation.
- **Key management layer autonomic DE:** provides self-management capabilities of optimal key storage and distribution. It supports real-time or near real-time closed-loop operation.
- **Control layer autonomic DE:** provides self-management capabilities of control plane resources and functional entities for the autonomic control orchestration and control entities management, etc. It supports real-time or near real-time closed-loop operation.
- **QKDN autonomic DEs provide global AMC capabilities for QKDN in network-level. It makes** autonomic policy decisions that encompass quantum, key management, and control layers as a slow closed-loop.
- **QKDN knowledge repository:** provides capabilities to store the QKDN-wide self-management policy information in a distributed manner to deal with the scalability of large volumes and performance of accessing distributed repositories.

- **Model-based information translation:** provides translation of self-management policies into layer specific provisioning rules. To support heterogenous types of information, a translation model is used to translate from/to heterogeneous information into a common one.
- **QKDN cognitive management:** provides a realization of the cognitive process by orchestration in network-level. It also supports interaction with QKDN management layer to deploy the DE decision policies into control layer autonomic DE, key management layer DE, and quantum layer DE and to monitor status of operation of DEs.

9.3 Reference Points for QKDNamc

The newly added reference points include:

- **Mkn:** a reference point connecting QKDN cognitive management and QKDN ML management. It is responsible for exchanging the orchestration information of requesting the ML capabilities and the learnt QKDN information between the QKDN ML management and the QKDN cognitive management.
- **Mkm:** a reference point connecting QKDN cognitive management and the cross-layer management orchestration. It is responsible for exchanging the orchestration information of realizing cognitive processes between the QKDN cognitive management and the QKDN manager.

10 Example operational procedures of QKDNamc

This clause describes some example operational procedures of AMC for QKDN based on the architectural model defined in the clause 9 to illustrate how autonomic management and control is performed in QKDN. Three AMC basic operational procedures are specified: AMC basic operational procedures for quantum channel performance prediction, key storage management, and key relay routing optimization. Note that three basic operation procedures referenced ML use cases specified in ITU-T Y-series Recommendations – Supplement 70 [b-ITU-T Y.Sup70].

10.1 Example AMC operational procedure supported by ML-based quantum channel performance prediction

Stable and predictable quantum channel performance and transmission quality in the quantum layer is crucial for the implementation and commercialization of QKDNs. The main challenge is that the noise falls into the quantum channel, thereby reducing the quality of quantum channel and causing low key rate, especially when quantum-encoded photons coexist with high-intensity classical signals. This procedure specifies autonomic management and control supported by ML-based quantum channel performance prediction such as optical signal-to-noise ratio (OSNR) and quantum bit-error ratio (QBER) as shown in Figure 10.1.

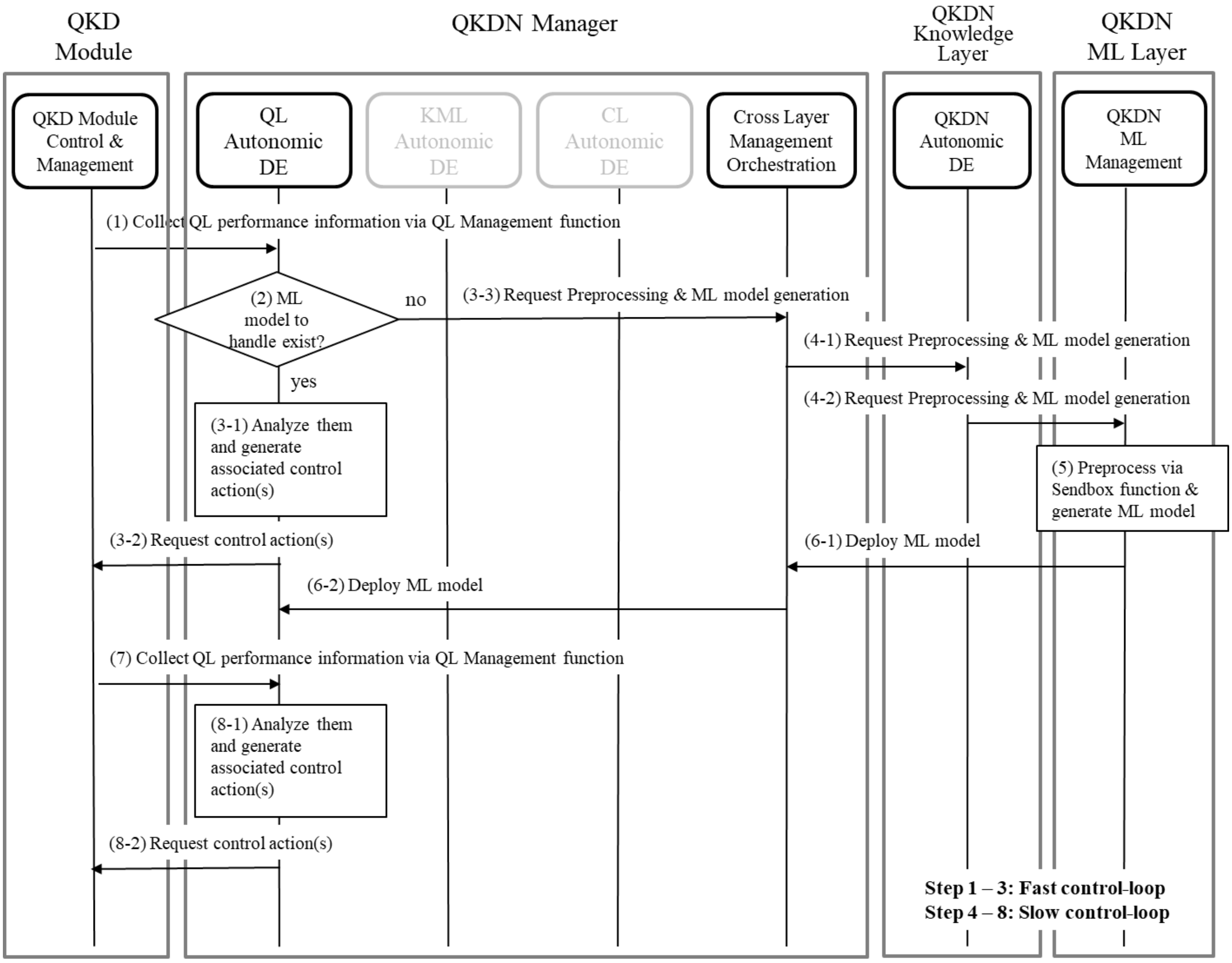

Figure 10.1 Example AMC operational procedure supported by ML-based quantum channel performance prediction

(1) The Quantum Layer autonomic Decision-making Element (QL DE) function in QKDN manager collects quantum channel performance-related parameters (e.g., QBER of quantum channel, the Sigle Photon Detector (SPD) output counter, code formation rates under different noise environments) from the Quantum Layer QKD module control and management function.

(2) The QL DE checks if ML model is available to analyze the collected data and generate associated autonomic control action(s).

(3-1) If an ML model exists, perform analytics of the collected performance data and generate associated control action(s)

(3-2) send it (them) to QL QKD module control and management function to apply. This control action is performed in real or near-real time which is the urgent process (fast control-loop) of the AMC cognition process.

(3-3) If an ML model doesn't exist, it requests to create an ML model to QKDN knowledge layer via Cross Layer Management and Orchestration (CLMO) function in QKDN manager.

(4-1) CLMO then conveys the request message to knowledge layer QKDN automatic DE for further processing.

(4-2) QKDN automatic DE then send the request to the QKDN ML layer to pre-process, train the data and generate an ML model.

(5) This process is handled by ML sandbox function in ML layer.

(6-1) The generated ML model deployment request is sent back to CLMO.

(6-1) Then it further is sent to CLMO.

(7) The QL autonomic DE function in QKDN manager collects quantum channel performance-related parameters which can handle them now.

(8-1) QL autonomic DE then performs analytics of the collected performance data and generates associated control action(s).

(8-2) send it (them) to QL QKD module control and management function to apply. This control action is performed in non-real time since the ML model generation process is involved which is the normal process (slow control-loop) of the AMC cognition process.

10.2 Example AMC operational procedure supported by ML-based key storage management

Since the QKDN services are dynamic and extensive, it is necessary to have efficient key storage management, so as to realize the reasonable scheduling and efficient utilization of key resources. The ML-based key storage management solution reasonably evaluates and predicts the health state of key storage. This procedure specifies autonomic management and control supported by ML-based key storage management as shown in Figure 10.2.

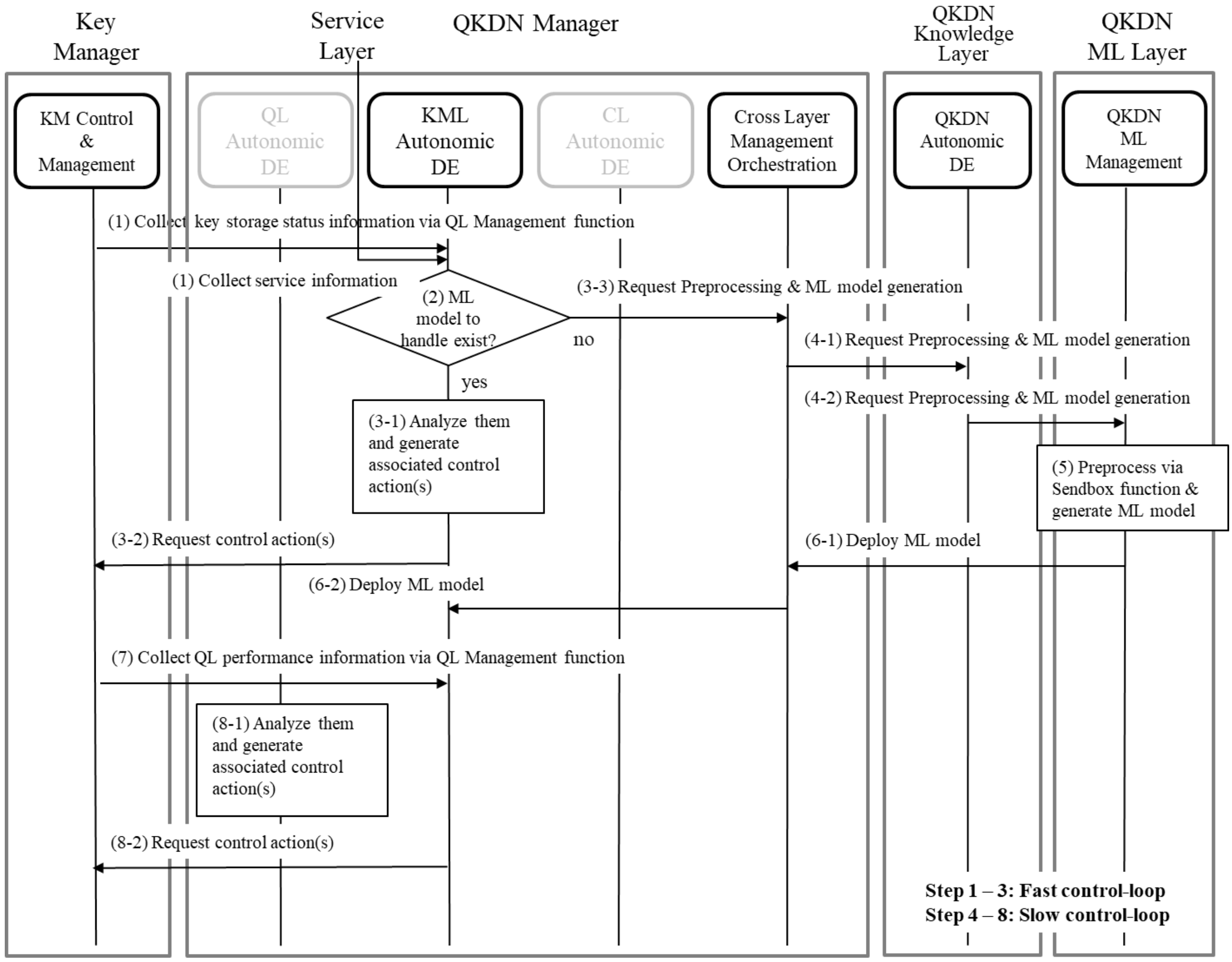

Figure 10.2 Example AMC operational procedure supported by ML-based key storage management

(1) The Key Management Layer autonomic Decision-making Element (KML DE) function in QKDN manager collects service data from the service layer (e.g., service type, security level, required key quantity) in real time and key storage status (e.g., key numbers, key life cycle) from the KML control and management function.

(2) The KML DE checks if ML model is available to analyze the collected data and generate associated autonomic control action(s).

(3-1) If an ML model exists, perform analytics of the collected service and key storage status data and generate associated control action(s)

(3-2) send it (them) to KML control and management function to apply. This control action is performed in real or near-real time which is the urgent process (fast control-loop) of the AMC cognition process.

(3-3) If an ML model doesn't exist, it requests to create an ML model to QKDN knowledge layer via CLMO function in QKDN manager.

(4-1) CLMO then conveys the request message to knowledge layer QKDN automatic DE for further processing.

(4-2) QKDN automatic DE then send the request to the QKDN ML layer to pre-process, train the data and generate an ML model.

(5) This process is handled by ML sandbox function in ML layer.

(6-1) The generated ML model deployment request is sent back to CLMO.

(6-1) Then it further is sent to CLMO.

(7) The KML autonomic DE function in QKDN manager collects service data and key storage status data which can handle them now.

(8-1) QL autonomic DE then performs analytics of the collected service and key storage data and generates associated control action(s).

(8-2) send it (them) to KML control and management function to apply. This control action is performed in non-real time since the ML model generation process is involved which is the normal process (slow control-loop) of the AMC cognition process.

10.3 Example AMC operational procedure supported by ML-based key replay routing optimization

When a service request arrives, an appropriate route needs to be selected according to the key requirements and resource states in QKDN. The QKDN control and management layers are responsible to find and provision the optimal key relay route. Due to the dynamic and explosive nature of services, the generation and consumption of key resources are often unbalanced. When the keys on the chosen route cannot meet the key requirements of services, the success rate of services is reduced. ML algorithms enable computation of an optimal routing in a reasonable amount of time. This procedure specifies autonomic management and control supported by ML-based key relay routing optimization as shown in Figure 10.3.

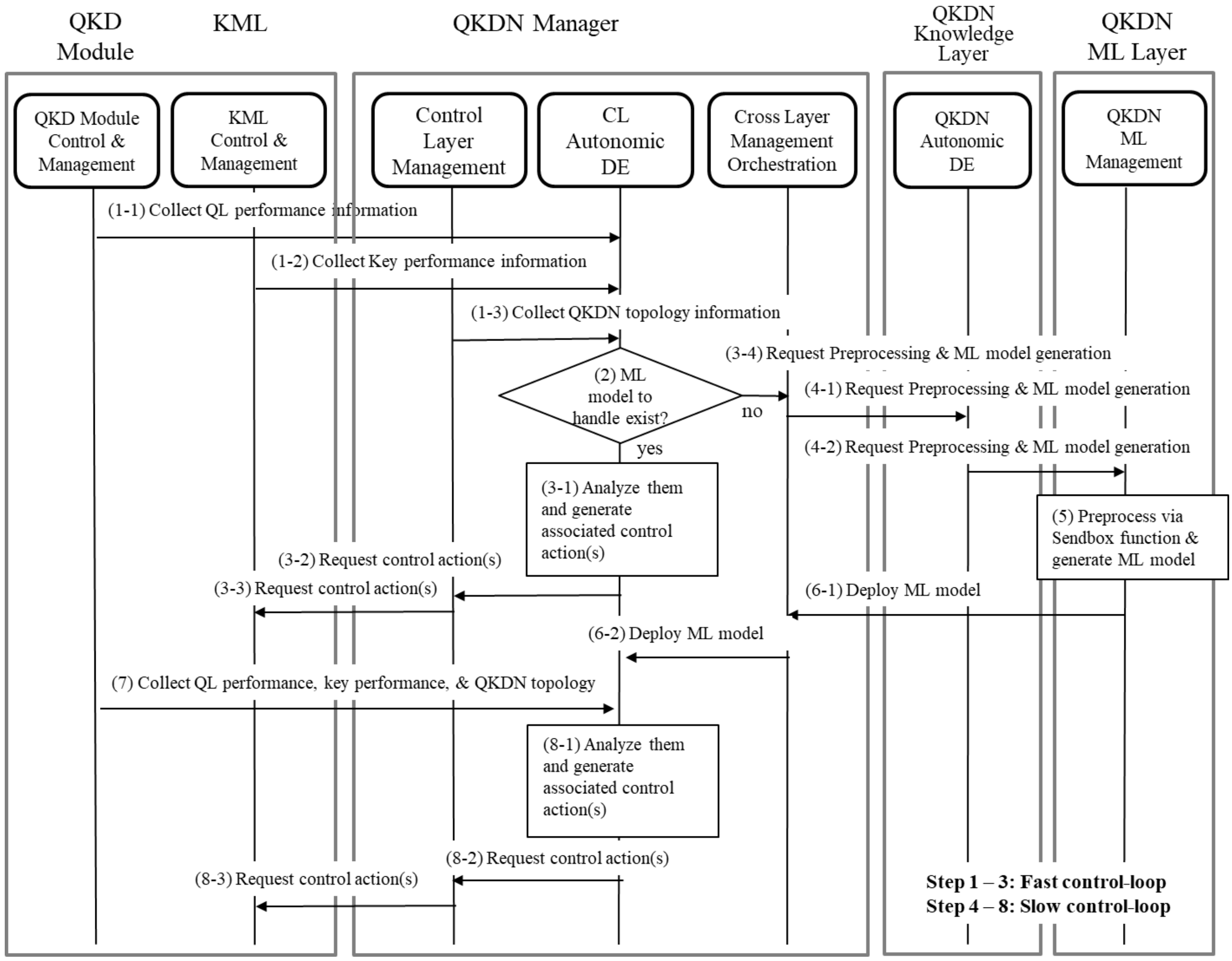

Figure 10.3 Example AMC operational procedure supported by ML-based key relay routing optimization

(1) The CL DE function in QKDN manager collects QKD link parameters, key consumption rate and service requirements, and QKDN topology from the QL QKD module, KML control and management function and CL management function in QKND manager.

(2) The CL DE checks if ML model is available to analyze the collected data and generate associated autonomic control action(s).

(3-1) If an ML model exists, perform analytics of the collected performance data and generate associated control action(s) which include an optimal key relay route. It may also include an re-routing action if re-routing is required to keep key relay routing optimal.

(3-2) send it (them) to CL management function.

(3-3) It further sends it (them) to KML control and management function to apply (that is, provisioning the optimal key relay route). This control action is performed in real or near-real time which is the urgent process (fast control-loop) of the AMC cognition process.

(3-4) If an ML model doesn't exist, it requests to create an ML model to QKDN knowledge layer via CLMO function in QKDN manager.

(4-1) CLMO then conveys the request message to knowledge layer QKDN automatic DE for further processing.

(4-2) QKDN automatic DE then send the request to the QKDN ML layer to pre-process, train the data and generate an ML model.

(5) This process is handled by ML sandbox function in ML layer.

(6-1) The generated ML model deployment request is sent back to CLMO.

(6-1) Then it further is sent to CLMO.

(7) The CL DE function in QKDN manager collects QKD link parameters, key consumption rate and service requirements, and QKDN topology which can handle them now.

(8-1) CL DE then performs analytics of the collected performance data and generate associated control action(s).

(8-2) send it (them) to KML control and management function to apply. This control action is performed in non-real time since the ML model generation process is involved which is the normal process (slow control-loop) of the AMC cognition process.

11 Security Consideration

This Recommendation describes requirements and architectural model for autonomic management and control for QKDNs; therefore, security requirements described in [ITU-T X.1710], [ITU-T Y.3801] and [ITU-T Y.3802] and general network security requirements and mechanisms in IP based networks described in [ITU-T Y.2701] and [ITU-T Y.3101] should be applied. Details are outside the scope of this Recommendation.

Appendix I
Use cases of AMC in QKDN

(This appendix does not form an integral part of this Recommendation.)

The following contents describe several potential use cases for AMC in QKDN.

I. AMC for optimization of key supply in QKDN

Need: To ensure high quality of key supply for the cryptographic applications with different characteristics such as time-sensitive or multi-granularity, it is essential to have dynamic optimization of key supply in QKDN while maintaining high operational efficiency and optimal resource utilization.

Solution: AMC for optimization of key supply in QKDN is to firstly monitor various parameters in QKDN, such as key generation rate, available key number and key demand number. Then, the collected data is analyzed to identify whether there is potential optimization space of key supply in QKDN with the help of the learnt knowledge based on the problem in QKDN such as key shortage, key surplus and key invalidation. The appropriate decisions of how to adjust the key supply are made for optimizing the key supply, so as to meet the requirements of cryptographic applications as much as possible. The prediction of the future key demand and supply situation can also be performed to help the decision-making. At last, the corresponding actions are executed by the QKDN controller and manager.

II. AMC for recovery of key supply in QKDN

Need: To ensure consistent key supply with high availability for the served cryptographic applications, it is essential to have rapid recovery of key supply in QKDN in the case of disruptions, system failures, or security breaches. The autonomic ability is needed to help the recovery of key supply to be timely and efficient.

Solution: The AMC for recovery of key supply in QKDN is firstly to monitor status of key supply for cryptographic applications, status of available keys in the key managers and parameters of the QKD/KM links. Then, the monitored data is analyzed to learn the key-supply-recovery related knowledge with the help of knowledge repository, to find whether there are abnormal status, and to predict the future status of key supply. If the key supply is found to be interrupted, the detailed recovery strategy of key supply will be decided such as the recovery path based on the key-supply-recovery related knowledge. At last, the QKDN controller will configure the recovery path for the cryptographic application to keep the key supply consistent.

Bibliography

[b-ETSI GR QKD 007] ETSI GR QKD 007 (2018), Quantum Key Distribution (QKD); Vocabulary.

[b-ETSI TS 103 195-2] ETSI TS 103 195-2 (2018), Autonomic network engineering for the self-managing Future Internet (AFI); Generic Autonomic Network Architecture; Part 2: An Architectural Reference Model for Autonomic Networking, Cognitive Networking and Self-Management. <https://portal.etsi.org/webapp/WorkProgram/Report_WorkItem.asp?WKI_ID=50970>

[b-ETSI WP 16] ETSI White Paper no. 16 (2016), The Generic Autonomic Networking Architecture Reference Model for Autonomic Networking, Cognitive Networking and Self-Management of Networks and Services. <http://www.etsi.org/images/files/ETSIWhitePapers/etsi_wp16_gana_Ed1_20161011.pdf>

[b-ITU-T E.800] Recommendation ITU-T E.800 (2008), Definitions of terms related to quality of service.

[b-ITU-T P.10] Recommendation ITU-T P.10/G.100 (2017), Vocabulary for performance, quality of service and quality of experience.

[b-ITU-T Y.Sup-70] Supplement ITU-T Y.Sup-70 (2021), *ITU-T Y.3800-series – Quantum key distribution networks – Applications of machine learning*