Attachments:

Annex I: A.1 justification for proposed draft new recommendation

Annex II: Proposed New Draft Recommendation ITU-T Y.QKDN-ml-fra "Quantum key distribution networks - functional requirements and architecture for machine learning"

Annex I:

A.1 Justification for proposed draft new recommendation

Question:	16/13	**Proposed new ITU-T Recommendation**	E-meeting, 5-16 July 2021	
Reference and title:	ITU-T Y.QKDN-ml-fra "Quantum key distribution networks - functional requirements and architecture for machine learning"			
Base text:	TD607/WP3		**Timing:**	2022-12
Editor(s):	Qingcheng Zhu, Yongli Zhao, Xiaosong Yu, Zhangchao Ma, Junsen Lai, and Taesang Choi		**Approval process:**	AAP

Scope (defines the intent or object of the Recommendation and the aspects covered, thereby indicating the limits of its applicability):

This Recommendation specifies the role of ML in QKDN, the functional requirements and architecture for ML in QKDN.

In particular, the Recommendation includes:

- Role of ML in QKDN;
- Functional requirements for ML in QKDN;
- Functional architecture model for ML in QKDN;

Summary (provides a brief overview of the purpose and contents of the Recommendation, thus permitting readers to judge its usefulness for their work):

QKDN is expected to be able to maintain the stable operation and meet various cryptographic application requirements in an efficient way. Due to the advantages of machine learning (ML) related to autonomous learning, ML can help to overcome the challenges of QKDN in terms of quantum layer performance, key management layer performance and QKDN control and management efficiency. Based on the functional requirements and architecture of QKDN in [ITU-T Y.3801] and [ITU-T Y.3802], this recommendation is to specify the role of ML in QKDN, the functional requirements and architecture for ML in QKDN.

Relations to ITU-T Recommendations or to other standards (approved or under development):

ITU-T Y.3800 "Overview on networks supporting quantum key distribution"
ITU-T Y.3801 "Recommendation ITU-T Y.3801 (2020), Functional requirements for quantum key distribution networks"
ITU-T Y.3802 "Recommendation ITU-T Y.3802 (2020), Functional architecture of the Quantum Key Distribution network"
ITU-T Y.3803 "Recommendation ITU-T Y.3803 (2020), Key management for quantum key distribution network"
ITU-T Y.3804 "Recommendation ITU-T Y.3804 (2020), Control and Management for Quantum Key Distribution Network"
ITU-T SG13 Y.QKDN_SDNC "Software Defined Networking Control for Quantum Key Distribution Networks"
ITU-T Y.3172 "Architectural framework for machine learning in future networks including IMT-2020"
ITU-T Y Suppl. 55 "ITU-T Y.3170-series – Machine learning in future networks including IMT-2020: use cases"
ITU-T Y.3174 "Framework for data handling to enable machine learning in future networks including IMT-2020"
ITU-T Y.supp.QKDN-mla "ITU-T Y.3800-series - Quantum key distribution networks - Applications of machine learning"

Liaisons with other study groups or with other standards bodies:

ITU-T SG2, SG16, SG17, FG-AN, FG-QIT4N

Supporting members that are committing to contributing actively to the work item:

Beijing University of Posts and Telecommunications, China; CAS Quantum Network Co. Ltd., China; QuantumCTek Co., Ltd., China; China Academy of Information and Communication Technology (CAICT), MIIT. P.R. China, ETRI.

Annex II:

Draft new Recommendation ITU-T Y.QKDN-ml-fra

Quantum key distribution networks - functional requirements and architecture for machine learning

Summary

QKDN is expected to be able to maintain the stable operation and meet various cryptographic application requirements in an efficient way. Due to the advantages of machine learning (ML) related to autonomous learning, ML can help to overcome the challenges of QKDN in terms of quantum layer performance, key management layer performance and QKDN control and management efficiency. Based on the functional requirements and architecture of QKDN in [ITU-T Y.3801] and [ITU-T Y.3802], this recommendation is to specify the role of ML in QKDN, the functional requirements and architecture for ML in QKDN.

Keywords

Functional requirements; functional architecture; machine learning (ML); quantum key distribution (QKD); QKD network (QKDN);

Table of Contents

Draft new Recommendation ITU-T Y.QKDN-ml-fra

Quantum key distribution networks - functional requirements and architecture for machine learning

1. Scope

This Recommendation specifies the role of ML in QKDN, the functional requirements and architecture for ML in QKDN.

In particular, the Recommendation includes:

- Role of ML in QKDN;
- Functional requirements for ML in QKDN;
- Functional architecture model for ML in QKDN;

This draft Recommendation specifies requirements for generic data collection. It does not specify the requirements for specific data related to PII.

2. References

[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.3802] Recommendation ITU-T Y.3802 (2020), *Functional architecture of the Quantum Key Distribution network.*

[ITU-T Y.3803] Recommendation ITU-T Y.3803 (2020), *Key management for quantum key distribution network.*

[ITU-T Y.3804] Recommendation ITU-T Y.3804 (2020), *Control and Management for Quantum Key Distribution Network.*

[ITU-T Y.QKDN_SDNC] Recommendation ITU-T Y.QKDN_SDNC (2019), *Software Defined Networking Control for Quantum Key Distribution Networks*

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

[ITU-T Y Suppl. 55] Recommendation Y Suppl. 55 (2019), *ITU-T Y.3170-series – Machine learning in future networks including IMT-2020: use cases*.

[ITU-T Y.3174] Recommendation ITU-T Y.3174 (2020), *Framework for data handling to enable machine learning in future networks including IMT-2020.*

[ITU-T Y.supp.QKDN-mla] Supplement ITU-T Y.supp.QKDN-mla (2021), *ITU-T Y.3800-series - Quantum key distribution networks - Applications of machine learning*.

3. Terms and definitions

3.1. Terms defined elsewhere

This recommendation uses the following terms defined elsewhere:

TBD

3.2. Terms defined in this Recommendation

This chapter defines all the terms used in this recommendation.

TBD

4. Abbreviations and acronyms

This chapters describes all the abbreviations and acronyms used in the recommendation.

KML	Key management layer
ML	Machine learning
QKD	Quantum key distribution
QKDN	Quantum key distribution network
QL	Quantum layer

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 prohibited from” 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.

The keywords “is not recommended” indicate a requirement which is not recommended but which is not specifically prohibited. Thus, conformance with this specification can still be claimed even if this requirement is present.

The keywords “can optionally” indicate an optional requirement which is permissible, without implying any sense of being recommended. This term is not intended to imply that the vendor’s implementation must provide the option and the feature can be optionally enabled by the network operator/service provider. Rather, it means the vendor may optionally provide the feature and still claim conformance with the specification.

6. Overview

Quantum key distribution network (QKDN) is a technology that extends the reachability and availability of quantum key distribution (QKD), which is stated in [ITU-T Y.3800]. It is comprised of two or more QKD nodes connected through QKD links. In a QKDN, two or more designated parties in a user network can share the keys for various cryptographic applications. When the QKDN becomes large-scale where there are multiple QKD nodes and links, the challenges of operating QKDN efficiently will increase. QKDN is expected to be able to maintain the stable operation and meet various cryptographic application requirements in an efficient way.

However, QKDN faces the following important challenges:

1) Without the awareness of sudden QKDN performance deterioration in advance, high cost (e.g. time cost, labour cost) and instability of QKDN will be increased.

2) For the large amount of heterogenous data in QKDN, there is difficulty to accurately perceive the needed and valuable information for use, which will affect the QKDN performance.

3) Since the requirements of cryptographic applications (e.g. different security requirements) are various and a large number of cryptographic applications arrive and leave dynamically, it is a difficult problem to schedule the QKDN resources for cryptographic applications under the finite resource limit.

To overcome the above challenges, QKDN needs to support the following capabilities:

1) The capability to predict the future QKDN status (e.g. quantum channel performance, remaining use life of components in a QKD system, potential faults in QKDN) and characteristics of cryptographic applications (e.g. the arriving time, the duration time, the required security levels) with high confidence.

2) The capability to perceive the useful information from a large amount of heterogenous data in QKDN accurately, so as to improve the QKDN performance efficiently.

3) The capability to detect the QKDN events (e.g. suspicious behavior detection, QKDN fault detection, QoS related events) accurately and timely, so as to trigger automatic and immediate actions.

4) The capability to optimize the schedule of QKDN resources for different cryptographic applications, which will maintain the stability of QKDN and improve the overall QoS in QKDN.

It is hard for traditional programmer, which develops traditional codes based on expert knowledge, to support the above capabilities, especially when there is no mathematically causal relationship among the data in QKDN and the expected information.

Applying ML technology into QKDN is a promising solution. ML can extract the implicit relationships between input and output data, and use this learnt mapping to analyse new data. ML techniques, especially neural networks, have been widely used in computer vision, language recognition, and automated control. ML can be applied to networking field which can intelligently learn the network environment and react to dynamic situations ([ITU-T Y.3170]). In recent years, ML technologies based on neural networks have seen much development in both hardware and software, and they have attracted a huge amount of attention from both the academia and the industry. There is also an increasing number of new low-power devices implementing on-board acceleration chips for neural networks.

Many applications of ML in QKDN have been discussed in [ITU-T Y.supp.QKDN-mla]. In the quantum layer of QKDN, ML can be applied to realize quantum channel performance prediction, QKD system parameter optimization and remaining use life (RUL) prediction of components in a QKD system; In the key management layer of QKDN, ML can be applied to realize intelligent key formatting, key storage management, and suspicious behavior detection; In the control and management layers of QKDN, ML can be applied in routing and QKDN fault prediction to improve control and management efficiency. Hence, ML can be beneficial for QKDN.

Due to the advantages of machine learning (ML) related to autonomous learning, ML can help to overcome the challenges of QKDN in terms of quantum layer performance, key management layer performance and QKDN control and management efficiency. Based on the functional requirements and architecture of QKDN in [ITU-T Y.3801] and [ITU-T Y.3802], this recommendation is to specify the role of ML in QKDN, the functional requirements and architecture for ML in QKDN.

7. Functional requirements for ML in QKDN

7.1 Functional requirements for QKDN data collection

The QKDN data can be collected from the quantum layer, key management layer, QKDN control layer, QKDN management layer and service layer either passively or actively.

The functional requirements for quantum layer data collection are as follows.

- QKDN is required to collect static QKDN data from quantum layer (e.g., parameters of QKD modules).
- QKDN is required to collect dynamic QKDN data from quantum layer (e.g., QKD performances, QBER, key generation rate).

The functional requirements for key management layer data collection are as follows.

- QKDN is required to collect static QKDN data from key management layer (e.g., history quantum layer data set).
- QKDN is required to collect dynamic QKDN data from key management layer (e.g., status of key storage, status of key authentication).

The functional requirements for QKDN control layer data collection are as follows.

- QKDN is required to collect static QKDN data from QKDN control layer (e.g., network topology).
- QKDN is required to collect dynamic QKDN data from QKDN control layer (e.g., routing and rerouting information, status of resource allocation).

The functional requirements for QKDN management layer data collection are as follows.

- QKDN is required to collect static QKDN data from QKDN management layer (e.g., history data of fault management, history data of configuration, history data of security management).
- QKDN is required to collect dynamic QKDN data from QKDN management layer (e.g., multi-layer resource usage data, multi-layer performance data).

The functional requirements for service layer data collection are as follows.

- QKDN is required to collect static QKDN data from service layer (e.g., history service information).
- QKDN is required to collect dynamic QKDN data from service layer (e.g., current service information).

7.2 Functional requirements for QKDN data pre-processing

- QKDN control layer is required to pre-process the collected QKDN data.
- QKDN is required to perform extract-transform-load (ETL) and transform the collected multi-source, heterogeneous QKDN raw data into understandable, unified and easy-to-use structures.
- QKDN is required to clean and filter noisy data from the collected multi-source, heterogeneous QKDN raw data.
- QKDN is recommended to normalize and unify the data format of the collected multi-source, heterogeneous QKDN raw data for further storage and analysis.

7.3 Functional requirements for history QKDN data repository

- QKDN is required to store a large amount of heterogeneous QKDN pre-processed data.
- QKDN control and management layers are recommended to support history QKDN data repository.

7.4 Functional requirements for modelling and training

- QKDN is required to construct ML models based on the pre-processed QKDN data.

- QKDN control layers are recommended to support ML modelling and training.
- QKDN is recommended to train the machine learning models be based on the available pre-processed QKDN data.

7.5 Functional requirements for ML model applying

- QKDN is required to select ML models.
- QKDN is recommended to support the ML model application in quantum layer, key management layer, QKDN control layer and QKDN management layer.

8. Functional architecture model for ML in QKDN

Based on the functional architecture model of QKDN in [ITU-T Y.3802] and the ML pipeline in [ITU-T Y.3172], the functional architecture model for ML in QKDN is designed, as shown in Fig. 8.1. The ML functions support a set of functional elements in a ML pipeline subsystem including collector (C), pre-processor (PP), model (M), policy (P) and distributor (D). The ML functions are able to collect input data from source of data (SRC) through data handling interfaces. The SRC can be in different layers of QKDNs. The target of the ML output (SINK) can be elements in quantum layer, key management layer and QKDN control and management layers. More details related to ML pipeline subsystems can be found in [ITU-T Y.3172].

Different layers of QKDN may support different ML functions. The ML pipeline functions in the QKDN control layer have the functions including the data collecting, data pre-processing, ML model construction and training, application of policy, distributing ML output. The ML pipeline functions in the QKDN management layer have the functions including the data collecting, application of policy, distributing ML output. The QKDN management layer plays the role as ML function orchestrator, which manages and orchestrates the ML pipeline subsystems based on QKDN-related ML intent and/or dynamic conditions.

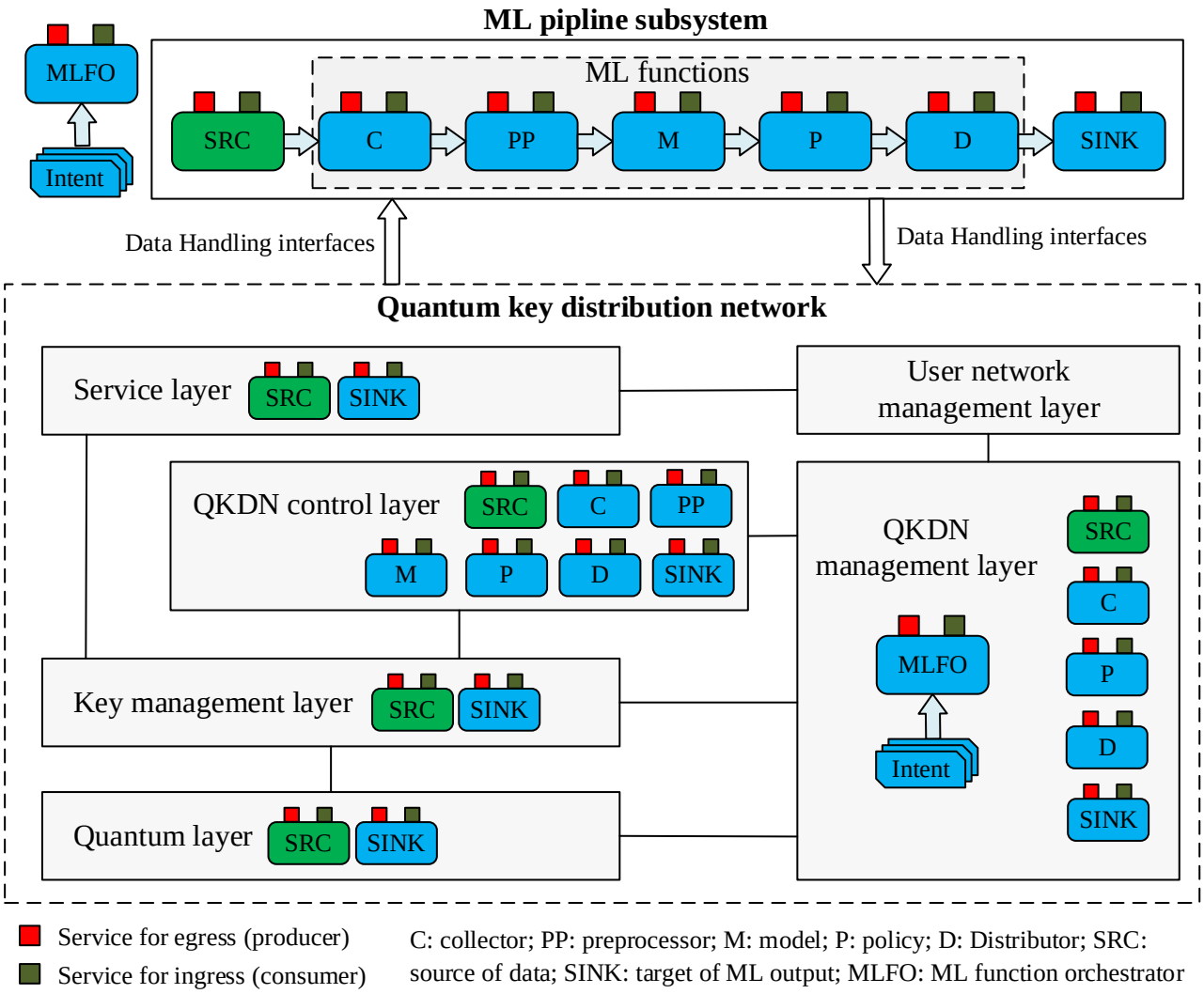

Fig. 8.1. Functional architecture model for ML in QKDNs

The functional architecture enhancement for ML in QKDN considers the following aspects. (TBD)

8.1 Enhancement for ML in the quantum layer of QKDN

8.2 Enhancement for ML in the key management layer of QKDN

8.3 Enhancement for ML in the QKDN control layer of QKDN

8.4 Enhancement for ML in the QKDN management layer of QKDN

8.5 Enhancement for ML in the service layer of QKDN

Bibliography

_____