Recommendation

ITU-T Y.3658 (04/2024)

SERIES Y: Global information infrastructure, Internet protocol aspects, next-generation networks, Internet of Things and smart cities

Big Data

Big data driven networking – Functional requirements and functional architecture of network programmability

ITU-T Y-SERIES RECOMMENDATIONS

Global information infrastructure, Internet protocol aspects, next-generation networks, Internet of Things and smart cities

GLOBAL INFORMATION INFRASTRUCTURE	Y.100-Y.999
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	Y.1000-Y.1999
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	Y.2000-Y.2999
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
Computing power networks	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	Y.4000-Y.4999
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.3658

Big data driven networking – Functional requirements and functional architecture of network programmability

Summary

Big data driven networking (bDDN) can provide a significant enhancement to the network programmability in a network control function, network infrastructure and network application. By using network programmability and intelligence of big data plane, bDDN can automatically adjust the network behaviour according to the user's requirements and network status. Recommendation ITU-T Y.3658 specifies an overview of the network programmability for bDDN, functional requirements of network programmability and functional architecture of network programmability for bDDN.

History *

Edition	Recommendation	Approval	Study Group	Unique ID
1.0	ITU-T Y.3658	2024-04-29	13	11.1002/1000/15874

Keywords

Big data driven networking, functional architecture, network programmability, requirement.

* To access the Recommendation, type the URL https://handle.itu.int/ in the address field of your web browser, followed by the Recommendation's unique ID.

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.3658

Big data driven networking – Functional requirements and functional architecture of network programmability

1 Scope

This Recommendation specifies the functional requirements and functional architecture of network programmability for big data driven networking (bDDN). The scope of this Recommendation includes:

a) Overview of network programmability for big data driven networking;

b) Functional requirements of network programmability for big data driven networking;

c) Functional architecture of network programmability for big data driven networking;

d) Security consideration.

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.2704] Recommendation ITU-T Y.2704 (2010), *Security mechanisms and procedures for NGN*.

[ITU-T Y.3650] Recommendation ITU-T Y.3650 (2018), *Framework of big-data-driven networking*.

[ITU-T Y.3653] Recommendation ITU-T Y.3653 (2021), *Big data driven networking – functional architecture*.

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following term defined elsewhere:

3.1.1 big data driven networking (bDDN) [ITU-T Y.3650]: bDDN is a type of future network framework, that provides big data intelligence to facilitate network management, operation, control, optimization and security, etc. based on the big data generated by the network itself.

It is notable that bDDN has the same meaning as DDN.

3.2 Terms defined in this Recommendation

None.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

API Application Interface

AR	Augmented Reality
BGP	Border Gateway Protocol
bDDN	Big Data Driven Networking
DDoS	Distributed Denial of Service
DHCP	Dynamic Host Configuration Protocol
FPGA	Field Programmable Gate Array
HTTP	Hypertext Transfer Protocol
IP	Internet Protocol
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6
IPU	Intelligent Processing Unit
MPLS	Multi-Protocol Label Switching
MR	Mixed Reality
NFV	Network Functions Virtualization
NFVI	Network Functions Virtualization Instance
NIC	Network Interface Card
SDN	Software Defined Networking
TLV	Type Length Value
VR	Virtual Reality
XR	Extended Reality

5 Conventions

This Recommendation uses the following conventions:

The keywords "is required to" indicate a requirement which must be strictly followed and from which no deviation is permitted, if conformance to this Recommendation is to be claimed.

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

In the body of this Recommendation and its annexes, the words 'should' sometimes appear, in which case they are to be interpreted as is recommended, and can optionally. The appearance of such phrases or keywords in an appendix or in material explicitly marked as informative are to be interpreted as having no normative intent.

6 Overview of network programmability for big data driven networking

Software defined networking (SDN) solution achieves the separation of network control and forwarding and enhances the centralized programmable ability of the control plane. It has been popular for the functions of the control plane on traditional switches and routers to be realized by the software. However, the programmable functions provided by the control plane are designed by hardware manufacturers and are closely bound with hardware. These functions are difficult to provide to a third party for secondary development. Although some network devices can provide more advanced SDN control capabilities, the functions that users can control are also very limited.

The purpose of SDN is to unify the programming ability of the control plane, so as to make the network equipment more flexible and not bound by the manufacturer.

Different manufacturers of forwarding devices have their own different implementation methods for data or packet forwarding. It is difficult to achieve consistency in the support of the new protocol, resulting in the problem of device compatibility. Once the new network protocol is published, each manufacturer will implement it separately. Because the protocol is bound to the hardware, manufacturers need to redesign the network equipment from the chip, which greatly limits the rapid development of the network technology.

The programmability of the application layer is always ignored, and with the continuous development of new applications such as augmented reality (AR) / virtual reality (VR) / mixed reality (MR) / extended reality (XR) and metaverse applications, it is often necessary to customize the network resources and capabilities according to the application requirements. Therefore, programmable capabilities are also needed at the application layer.

The current network is actually a huge, distributed computing system. In order to meet the requirements of applications, network devices should be able to identify the intention of the applications and provide services according to the intention of the applications. Network programmability in bDDN, such as the computer programmability, is to translate the intention of the service carried by the network into a series of forwarding instructions of the network infrastructure. The network infrastructure completes the corresponding forwarding action according to the forwarding instructions to meet the requirements of services for the network, so as to meet the customized requirements of the service and realize the network programmability. The bDDN network programmability model is shown in Figure 6-1.

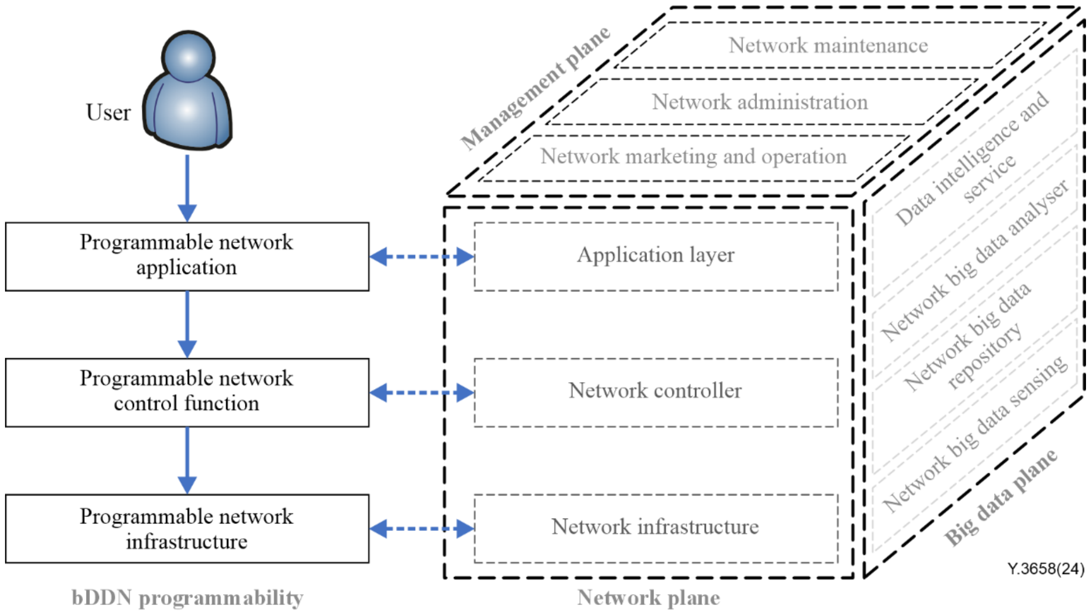

Figure 6-1 – bDDN network programmability model

Network programmability is the ability to deploy, manage, and troubleshoot network elements through software. Network programmability in bDDN is the ability that bDDN manages and adjusts the network behaviour according to the user's requirements and intelligence of the big data plane. The network programmability of bDDN enables bDDN to adapt to the requirements of new types of applications as well as the complex needs of various users without changing the network architecture.

The network programmability in bDDN is realized mainly through three parts: programmable network application, programmable network control function, and programmable network infrastructure.

Programmable network application: Network application is an end-to-end service running on the network. The programmable network applications need to carry the requirements of users and translate these requirements into specific performance requirements that the network can recognize. Different applications have different network QoS and resources requirements, including bandwidth, delay, jitter and other performance capabilities. The application also needs to program these requirements into the application's frame structure and then transmit them to the underlying software stack.

Programmable network control function: Programmable network control function is similar to the operating system in the computer. It will send the requirements of network applications into executable instructions of network infrastructure according to the context and state of the network.

Programmable network infrastructure: The programmable network infrastructure is responsible for forwarding packets or bits and executing some instructions of the network. The instructions may be firstly processed by the control plane of the network, such as the Internet protocol (IP) routing protocol, multi-protocol label switching (MPLS) protocol and other protocols. It is then processed by the network infrastructure, such as the programmable forwarding chip of the switch, router and the switch module of the network functions virtualization (NFV). Programmable network infrastructure can be traditional hardware devices, or software modules. Generally, the programmable forwarding infrastructure part needs to have programmable matching tables and packet header parsers parts.

bDDN can realize not only the programmability of network infrastructure, but also the programmability of network applications and network control functions. The programmable ability of network infrastructure is mainly used for packet forwarding, load balancing, network measurement, fault diagnosis, security prevention and other scenarios. The programmability of the programmable network infrastructure can forward and process the packets according to the user's intention. And the programmable ability of network applications is mainly used to identify the intention of applications and realize the customization of the user requirements. This way is to carry the user's intention through the network protocol and packet. Without changing the forwarding infrastructure, the user's intention is translated into a series of labels or bits, which are processed by the network infrastructure, so as to achieve flexible orchestration and on-demand customization of the network services. The programmability of the network control function can easily achieve the expansion of network protocols and network functions. At the same time, users and operators can easily control network behaviour by a programmable network control function.

7 Functional requirements of network programmability for big data driven networking

With the development of different subdivision technologies in the network, independent programmable technologies in different layers of networks have emerged, including programmable applications, programmable network control functions and programmable network infrastructure. bDDN network framework has a deep programmability in various aspects. The network programmability is very important for bDDN due to the following points:

1. Network is becoming a distributed computing system

With the advent of SDN including programmable forwarding and advanced programming language for hardware, the network has become a deeply programmable platform that can be controlled by network owners to meet their requirements. It is now possible to program the control plane and the forwarding pipeline end-to-end on the servers, network interface cards (NICs), switches and optical switch devices. With this, we can now reimagine the network as a dynamic distributed computing system.

2. Operator can take control of the network on demand

With the network now programmable end-to-end and top-to-bottom, software now defines network behaviour, control moves from equipment vendors to network owners, and network functions can be placed where they are best suited which may be in the hardware, on device software, or on the central controller. Thereby, the operator can take control of their own destiny when they start to leverage the network as a distributed computing platform, and the network becomes an asset for innovation and competitive advantage. The network programmability can enable the operator to build and dynamically deploy customized network functionality in a secure and reliable manner.

3. The future network must be a programmable network

The application scenarios of bDDN will be applied in future networks. Through rich programmability and open interfaces, operators can control their own network more freely. In bDDN, to automatically adjust the network policy and routing according to the user's needs, the network framework needs to support programmable features.

4. bDDN requires the network programmability

The bDDN is required to manage and adjust the network according to the big data analysis of the network state, so that the network can better serve the operators and users. Therefore, the network plane of big data driven networking must have programmable ability. This ability is to provide operators and users with the ability to change the network behaviour. In the bDDN, the network plane is no longer a black box.

7.1 Use case of network programmability for bDDN

1. Specify network forwarding behaviour

With the programmability of network application, network control function and infrastructure, the network operator can control the forwarding path and behaviour of the network equipment in the network according to the service's requirement or user's intention. bDDN can insert the forwarding operation or other requirements into the programmable application at the source of the routing path to guide the forwarding behaviour of the network infrastructure in the network. With this network programmability of network infrastructure, bDDN can meet different network QoS requirements, such as network slicing, deterministic delay and so on. Combined with the global network management and control capability of bDDN, it can realize flexible network functions and facilitate faster the deployment of new services.

According to the above analysis, the programmability of the application, network control function and infrastructure is required in this use case.

2. Real-time network intelligent awareness

The network management and maintenance depend on timely and effective network measurement and monitoring. Through programmable infrastructure, intelligent awareness technology can measure a wider range of network data, and collect flow-based measurement data. By programmable applications, users can not only send collection instructions to the underlying infrastructure, but also query some information about the device, such as queue size, link utilization, queue delay, etc.

According to the above analysis, programmable infrastructure and programmable applications are required in this use case.

3. Load balance

In large data centres, most of the traffic is virtual IP address traffic, which requires load balancing to maintain the consistency of each connection. Load-balancing often needs complex appliances that can be a burden on the network infrastructure. With programmable network infrastructure, load balancing can be done inside the switch, bridging the gap between multi terabit switches and gigabit servers and appliances. The consistency of each connection can be maintained through the bloom filter implemented by programmable infrastructure, so as to maintain the stately connection of millions of connections on the switch at the same time. This method not only reduces the cost of load balancing, but also reduces the delay and jitter. In addition, the existing transport layer protocol cannot give good consideration to low delay and high throughput. With the help of a programmable network control function, users can customize a new transport layer protocol in the data centre to solve this problem. Moreover, the programmable switch adopts packet by packet multi-path load balancing technology to realize low delay and high throughput transmission.

According to the above analysis, programmable infrastructure and programmable network control functions are required in this use case.

4. Network attack detection

Distributed denial of service (DDoS) attack is the largest and the most frequent network attack means in the current network environment. The traditional method uses middleware to alleviate it. The DDoS traffic cleaning method on the programmable network infrastructure is adopted to avoid interaction with the controller. The server does not need to maintain a large amount of traffic and data thus, it can deal with large attack traffic. Programmability of network infrastructure can enable networks that detect DDoS attacks and other network attacks faster and mitigate them more effectively due to their exceptional high performance, and because these tasks are performed in the network and do not rely on other resources.

According to the above analysis, programmable infrastructure is required in this use case.

5. Network resource optimization

The packet flow table resource is an important network resource affecting network scalability. On the one hand, programmable network infrastructure can realize a more flexible flow table matching method and provide an intelligent flow table expiration mechanism to make full use of the flow table resources. On the other hand, programmable network control functions can provide a flexible protocol extension which can make network devices support new protocols easily. Therefore, programmable network infrastructure and programmable control functions bring opportunities for network resource optimization and a more scalable data plane.

According to the above analysis, programmable infrastructure and programmable network control functions are required in this use case.

7.2 General functional requirements of network programmability for bDDN

The general requirements include programmable requirements, compatible requirements and protocol-independent requirements, which are for all use cases mentioned above.

1. Programmable requirements

– The application layer in bDDN is required to support programmable ability.

  The programmability of network application in bDDN means the user can program the application's requirements into the packet that carries the application. These requirements are translated into an ordered list of instructions, which are executed by the network devices along the path.

– The control function of the network plane in bDDN is required to support programmable ability.

  Due to the continuous emergence of new network protocols, the traditional switch cannot support the new protocol. Therefore, this requirement is to flexibly define the packet processing flow of the network control function by programming without replacing the forwarding hardware.

– The network infrastructure of the network plane in bDDN is required to support programmable ability.

This kind of network infrastructure can be designed based on the network devices such as switches, routers, network functions virtualization instance (NFVI) devices and other software forwarding components, as well as the smart NIC (network interface card) of the server in the cloud centre.

2. Compatible requirements

In bDDN, there are many kinds of network infrastructure, such as traditional switches, routers, NFV virtual devices, and field programmable gate array (FPGA) / intelligent processing unit (IPU) devices with programmable ability. The compatible requirements of the infrastructure are as follows:

– It is required for bDDN to be compatible with traditional infrastructure such as switch, routers, etc.
– It is required for bDDN to be compatible with traditional network protocols such as Internet protocol version 4 (IPv4), Internet protocol version 6 (IPv6), MPLS, etc.
– It is required for bDDN to be compatible with traditional application layer technology such as hypertext transfer protocol (HTTP), etc.

The programmable interfaces of these devices are also diverse, and the unified management and development are difficult. For users and applications, a unified interface can support the development of new business applications and the unified management of infrastructures. The requirements of the programmable interface are as follows:

– It is required for bDDN to have a unified programmable interface that should hide the differences of manufacturers and provide a consistent access interface for the upper application.
– It is required for bDDN to open the programmable interface for users.

3. Protocol-independent requirements

As a large number of protocols supported by traditional switches cannot be used in all scenarios, there are many redundant protocols. At the same time, in order to support new applications and network functions, bDDN is required to be protocol-independent. Therefore, this requirement is to make the network infrastructure not needed to pay attention to the format and semantics of the protocol. The packet processing behaviour supported by the network infrastructure is not limited by the protocol type. The requirements of protocol-independent are as follows:

– It is required for bDDN infrastructure to be protocol-independent.
– It is required for bDDN network control function to be protocol-independent.
– It is required for bDDN application to be protocol-independent.

7.3 Functional requirements of network programmability for bDDN

7.3.1 Functional requirements of programmable network application

The programmability of the application layer is also an important capability of bDDN, and the programmability requirements of the application layer mainly include the following:

– It is required to have packet fields for network protocol that can specify service level agreements.
– It is recommended to have packet fields for network protocol that can specify end to end delay parameters of the application or service.
– It is recommended to have packet fields for network protocol that can specify end to end jitter parameters of the application or service.

– It is recommended to have packet fields for network protocol that can specify end to end bandwidth parameters of the application or service.
– It is recommended to have packet fields for network protocol that can specify other parameters of the application or service.
– It is recommended to have packet fields for network protocol that can specify the forwarding path of the application or service.
– It is recommended to have packet fields for network protocol in the packets that can specify the actions of a forwarding node to handle the packet.
– It is recommended to have packet fields for network protocol to specify the required computing power.
– It is recommended to have packet fields for network protocol to specify the parameters to be measured.

7.3.2 Functional requirements of network control function

– It is required that the network control function can handle programmable parameters passed from the application layer.
– It is required that the control function handle service level parameters passed from the application layer.
– It is recommended that the control function can handle service performance measurement requirements.
– It is recommended that the control function can measure the performance parameter of the service.
– It is recommended that the control function can measure the end-to-end delay parameters of the service.
– It is recommended that the control function can measure the end-to-end jitter parameters of the service.
– It is required that the control function can abstract the different manufacturers hardware.
– It is required that the control function can handle the requirement from the application layer.
– It is required that the control function can abstract the network device to the logical functional block.
– It is recommended that the control function can abstract switch devices into standard models.
– It is recommended that the control function can abstract router devices into standard models.
– It is recommended that the control function can abstract optical transport network devices into standard models.
– It is recommended that the control function can abstract access network devices into standard models.
– It is recommended that the control function can abstract network links and optical lines into standard models.
– It is recommended for the control function to provide the configuration interface of the link layer.
– It is recommended for the control function to provide the configuration management function.
– It is recommended for the control function to provide the topology management function.

- It is recommended for the control function to provide the link discovery management function.
- It is recommended for the control function to provide the performance management function.
- It is recommended for the control function to provide the fault management function.
- It is recommended for the control function to provide the policy management function.

7.3.3 Functional requirements of programmable network infrastructure

- It is required that the packet processing and forwarding logic of the programmable infrastructure can be programmed or configured through software.
- It is required that programmable infrastructure provide programmable interfaces for the upper layer.
- If the programmable infrastructure is a type of packet forwarding device, it is recommended that the programmable infrastructure support the advanced programming languages.
- If the programmable infrastructure is a type of packet forwarding device, it is required that the programmable infrastructure can program packet processing and forwarding logic according to the requirements of the upper layer.
- If the programmable infrastructure is a type of packet forwarding device, it is recommended that the programmable infrastructure have a programmable ingress with a match action table.
- If the programmable infrastructure is a type of packet forwarding device, it is recommended that the programmable infrastructure have a programmable egress with a match action table.
- If the programmable infrastructure is a type of packet forwarding device, it is recommended that the match unit in ingress and egress can match any field and offset of a packet.
- If the programmable infrastructure is a type of packet forwarding device, it is recommended that the action unit in ingress and egress has programmable forwarding actions and packet processing functions.

8 Functional architecture of network programmability for big data driven networking

8.1 General functional architecture of network programmability for bDDN

The functional architecture of network programmability is based on the framework of bDDN defined in [ITU-T Y.3650] and the functional architecture defined in [ITU-T Y.3653]. The architecture defined in [ITU-T Y.3653] is the functional architecture of bDDN, while the architecture of this Recommendation develops and enhances the programmable functions and capabilities of bDDN. This architecture mainly includes three levels of programmable functionalities in the network plane of bDDN, including programmable network application, programmable network control function and programmable network infrastructure, as shown in Figure 8-1.

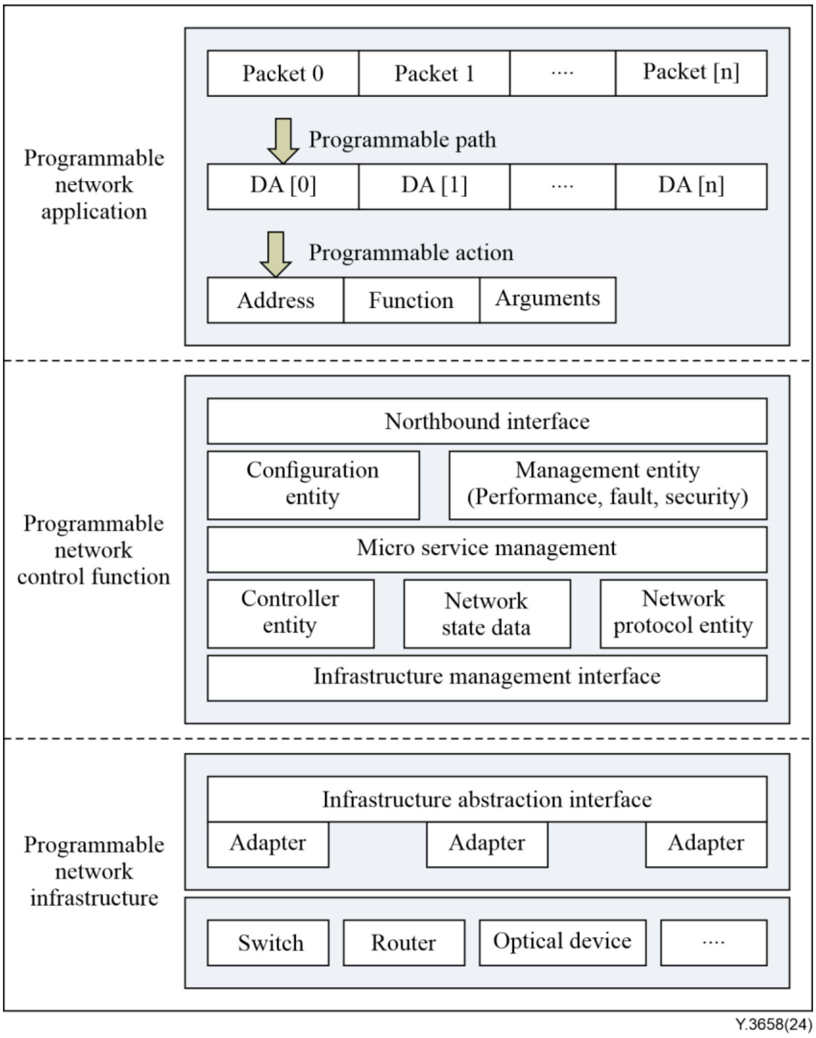

Figure 8-1 – Functional architecture of network programmability for bDDN

1 Programmable network application

The application layer in the network plane of bDDN also has the programmable capability. The programmable capability of the application includes two aspects:

– First, the packet forwarding path is programmable by the user or application. The packet forwarding path is not determined by the traditional routing protocol but is calculated by the bDDN big data plane according to the user's SLA requirements. The calculated path uses the source routing method to carry the path nodes in the data packet.

– Second, application programmability of bDDN can program the behaviour of the forwarding nodes in the path. The data packet can carry some information to the nodes in the forwarding path, and the forwarding node can complete the tasks according to the information. The application packets can carry some information for each node, which can be passed to the node in the form of type length value (TLV) or other parameters. This instruction information represents some customized requirements of the application for this node, such as QoS instructions, OAM instructions, measurement information and other instruction information. When the forwarding nodes process the packets, nodes can complete the specific task according to the information besides the forwarding packets. This can better meet the needs of the users and achieve flexible programming capabilities.

2 Programmable network control function

The network control layer is composed of a network controller and a network protocol stack, which is responsible for controlling the network resources and behaviour. The programmability of the network control stack is mainly to adapt to the extension of new network functions and new network protocols.

In the network control function, bDDN uses data media to achieve the decoupling of network function modules. Network control function modules no longer interact with each other through the direct data transmission mode but through the intermediate database. The network function changes from focusing on the process to focusing on the data, which realizes the decoupling among the network functional modules. The architecture using data media not only provides a convenient development of new functions but also provides the ability to solve function failures and function upgrades without affecting the forwarding plane.

The protocols of the network software (e.g., dynamic host configuration protocol (DHCP), border gateway protocol (BGP), etc.) are completely independent of the communication details of the specific hardware. The network control function reads the required data from the database or writes the data to the database without directly communicating with the hardware.

3 Programmable network infrastructure

The network infrastructure layer also needs to be programmable. The processing and forwarding logic of a network can be programmable by the network control function or applications. Programmable infrastructure allows network operators to define the complete processing flow of data packets from end to end. The forwarding logic of the network infrastructure layer is determined by the program loaded on the infrastructure.

Abstraction interface is part of the programmable infrastructure and provides a unified application interface (API) for the upper layer network control function. The network control function can run on different infrastructure platforms through the abstract interface. Because of the existence of an abstraction layer, the network control function can support any infrastructure from multiple manufacturers. On the basis of matching the abstraction interface, network infrastructure can provide features such as higher speed, lower power consumption, lower cost, and higher port density, without a specific adaptation of the network control function. This way enabling the infrastructure and network control functions to rapidly iterate and develop respectively.

The programmable infrastructure in bDDN includes optical switch devices, packet switches, routers and other devices with switching capability.

8.2 Functional architecture of programmable network application

The programmable network application of bDDN can program the application's requirements or network functions into the packet that carries the application. These requirements or functions are translated into an ordered list of instructions, which are executed by network devices along the way, achieving flexible programming and on-demand customization of the network applications.

Usually, a computer instruction includes two aspects: the operation instruction and the parameters of the instruction. Operation instructions determine the computer operation to be executed and parameters refer to the data needed by the instruction. Similarly, the programmability of the application of bDDN also requires the definition of network instructions, where a network instruction consists of three parts, as shown in Figure 8-2, address, function and arguments.

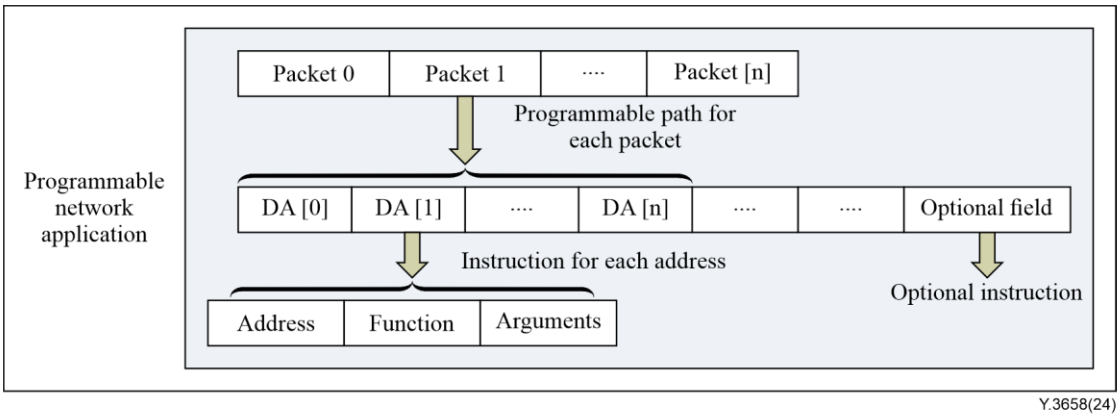

Figure 8-2 – Functional architecture of programmable application

Address is the identifier of a network node or device in a network topology, used for routing and forwarding packets to that node. The location information is identified by the address. The function field is used to express the forwarding action to be executed by the instruction, similar to the opcode of a computer instruction. In bDDN, different forwarding behaviours are represented by different functions, such as forwarding packets to a specified node or performing forwarding path lookup in a specified table. The arguments field is an optional field. It is the parameter corresponding to the instruction during execution, which may contain flow, service, or any other related information.

Instructions are the foundation for network application programmability. Based on these instructions, any network requirements of end-to-end service can be represented through a set of instructions. The application programmability of bDDN is achieved by encapsulating an ordered instruction list in the packet in the source node, instructing the network to execute corresponding instructions on the specified node to achieve network programmability. With the increasing number of application scenarios for programmable networks in the future, this instruction set continues to evolve and expand.

8.3 Functional architecture of programmable network control function

The programmable control function of bDDN is responsible for managing the underlying infrastructure and providing programming interfaces to upper level applications. The programmable network control function is different from the traditional network software protocol stacks. The functional architecture is shown in Figure 8-3.

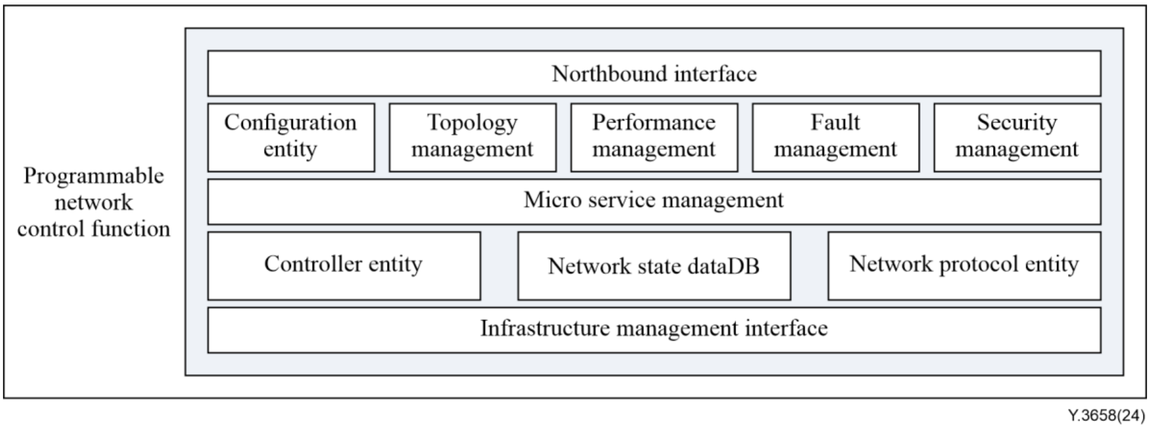

Figure 8-3 – Functional architecture of programmable control function

Programmable network control function provides open and scalable northbound interfaces for the application layer, rather than fixed interfaces based on manufacturers. These interfaces are mainly used to provide standard interfaces for applications, including service management interfaces, requirement management interfaces, configuration interfaces, topology management interfaces, device management interfaces, performance management interfaces, alarm management interfaces and other management interfaces. The application accesses devices based on these standard interfaces and passes the requirements to the underlying network. The application can be easily extended because the standard interface is used.

Programmable network control function manages the underlying infrastructure through standard infrastructure management interfaces. The infrastructure management interface logically abstracts various underlying infrastructures into standard network elements and functions. Programmable control function is designed and developed based on these standard network elements and functions. Through the infrastructure management interface, standard functional interfaces are provided to the network control stack and applications. Infrastructure management interfaces include device management interfaces and programming interfaces from manufacturers, which can be interfaced with advanced programming languages.

The main functions of the programmable control function include service management, slice management, resource scheduling, configuration management, topology management, device management, performance management, etc. These control functions are designed and developed based on the logical network elements and functions and are not directly related to specific manufacturers' hardware devices or specific application services. This approach makes the network control stack easier to expand.

The functional modules of the programmable network control function are loosely coupled based on microservices and data buses. This loose coupling relationship facilitates the expansion of new functions and protocols in the network control function. When it is necessary to extend new functions or add new protocols, only the new software modules need to provide standardized service interfaces and data interfaces. This approach greatly improves the network control function's scalability for new features and protocols.

Programmable network control function can manage various hardware and software resources, including computing, storage, networking, etc. These resources are accessed through infrastructure management interfaces. Network control function can orchestrate and schedule computing, storage, network and other resources based on the standard interfaces.

8.4 Functional architecture of programmable network infrastructure in bDDN

Programmable infrastructure can either be a packet switch device, such as programmable routers or switches, or a non-packet device such as optical transmission network devices. Since the underlying equipment of each manufacturer has different programmable capabilities, both packet infrastructure and non-packet infrastructure use standard programmable interfaces to shield the differences in the equipment. At the equipment layer, they may include various devices, such as routers, switches and optical devices. Programmable network infrastructure consists of two parts, one is the infrastructure standard interface, and the other is infrastructure entities, such as routers, switches and optical devices as shown in Figure 8-4.

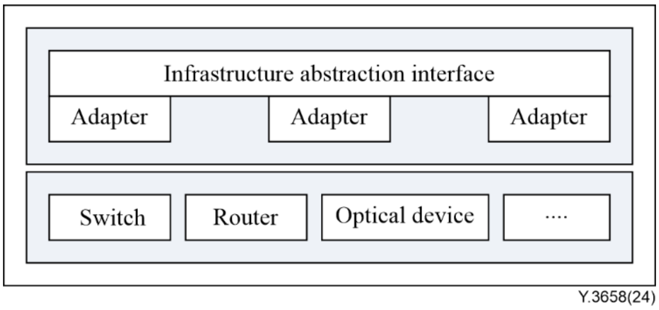

Figure 8-4 – Functional architecture of programmable infrastructure

8.4.1 Network infrastructure abstraction interface

The infrastructure abstraction interface can keep the base network control function simple, consistent and stable. It breaks the software-hardware coupling and enables us to choose the best fit of software and hardware based on the requirement of application or the requirement of networking. By providing simple, consistent interfaces for applications and protocol stacks, it helps reduce the impact of the underlying complex and heterogeneous infrastructure easily. The infrastructure abstraction interface is therefore a big step towards programmable networking infrastructure.

A standardized infrastructure API allows the network hardware vendors to develop innovative hardware architectures to achieve great speeds while keeping the programming interface consistent. As new hardware functions are exposed, hardware vendors can introduce extensions to the API. This would introduce change at the hardware programming level, but this change would be much less frequent than traditional hardware as it would only be required for functional changes, not simply implementation differences.

An important component of the infrastructure abstraction interface is the adapter. Adapter is a pluggable module, supplied by a vendor, that contains hardware driver code and implements the abstraction interfaces. Adapters are registered with the network control function and can then be loaded as needed. It is the responsibility of the adapter to discover and bind to the specified underlying hardware, including loading of or attaching to drivers if needed. Adapters are expected to be as simple as possible, ideally simple wrappers around the vendor's driver. During initialization, the adapter initiates the discovery process of the specified instance of a switching entity.

The infrastructure abstraction interfaces API are grouped into three categories:

– Mandatory functionality. This is a set of "core" interfaces that are required to build a basic switch appliance, which might be a packet switch or optical switch. All vendors must support these interfaces and the network control function stack will fail loading of the adapter, if any of these interfaces are missing.

– Optional functionality. This is a set of additional interfaces which are not required but enable various scenarios in some special cases. Definition of these interfaces is common for all the vendors. These interfaces are not required by the network control function stack to be exposed from the adapter. However, they become required if a given system configuration references any of these features.

– Custom defined functionality. This is a set of interfaces that are unique for a vendor and is not standardized. The default adapter host is not aware of these interfaces and a custom adapter host can be supplied by the vendor to expose these interfaces to the network control function stack.

8.4.2 Network infrastructure entities

In bDDN, there may be multiple underlying switching devices, such as optical switch devices, packet switch devices, or virtual network function based on software. Computing devices and storage devices will gradually integrate together to provide services. Therefore, bDDN supports multiple hardware and software devices to form its own underlying infrastructure. All these underlying infrastructures are programmable, including optical switch devices, packet switch devices and virtual network function. However, from the standardization perspective, the functional architecture of packet switch infrastructure is only illustrated here.

The programmable packet switch infrastructure consists of a few packet forwarding pipelines. A pipeline represents a complete set of data processing procedures. As shown in Figure 8-5, a pipeline is composed of the programmable parser, programmable match-action table and programmable deparser. The main modules of the pipeline are all programmable. A packet switch device can consist of multiple pipelines. When there are multiple pipelines, they may be parallel or serial. The former can carry more bandwidth and more network ports, while the latter can realize more complex functional logic and obtain more table capacity. Usually, a packet switch device has two stages, ingress and egress. Both ingress and egress stages can adopt this pipeline, or only one of them can be used.

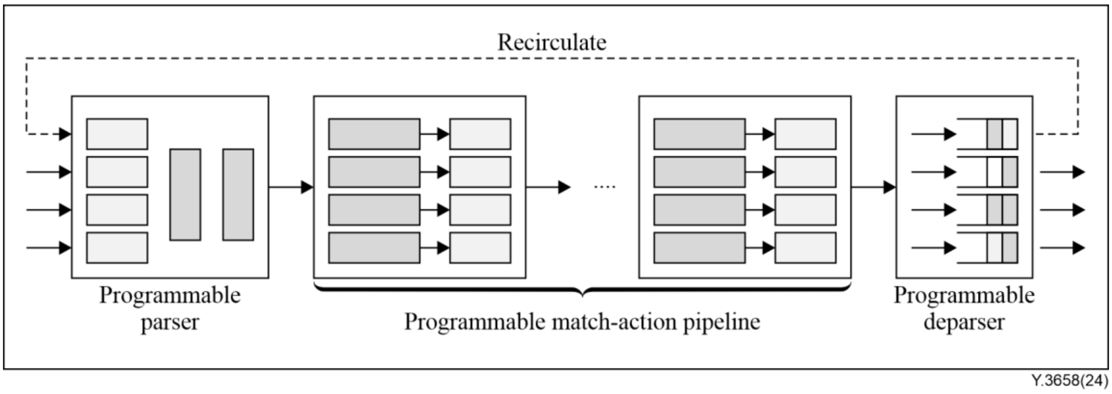

Figure 8-5 – Pipeline of programmable packet infrastructure

The programmable interface and network control function are basically programmed for these modules of the pipeline. These modules are not completely mandatory. In some cases, some modules may not appear in the pipeline (e.g., egress parser/deparser). When each packet passes through these processing modules, there is always a buffer or memory for storing metadata and caching intermediate processing data and results. The metadata information can be the output result of an action or the input of a match in the next pipeline.

9 Security considerations

When applying network programmability in bDDN, security best practices should be adopted such as authentication, authorization and access control as described in [ITU-T Y.2704].

In the meantime, the operation related to network resources should have multiple reliability guaranteeing measures in order to avoid incorrect operation to network resources and causing the degrading of the network performance.

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

Published in Switzerland
Geneva, 2024