International Telecommunication Union

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

Y.2775

(08/2019)

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

Next Generation Networks – Security

Functional architecture of deep packet inspection for future networks

Recommendation ITU-T Y.2775

International Telecommunication Union

ITU-T Y-SERIES RECOMMENDATIONS

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

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

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

Recommendation ITU-T Y.2775

Functional architecture of deep packet inspection for future networks

Summary

Recommendation ITU-T Y.2775 specifies the functional architecture of deep packet inspection (DPI) for future networks (e.g., software-defined networking, network function virtualization). This Recommendation specifies general DPI functional architecture aspects related to future networks, DPI functional architecture for software-defined networking, DPI functional architecture for network function virtualization, DPI functional architecture for service function chaining and DPI as a service, DPI functional architecture for network virtualization and DPI functional architecture for evolving mobile network.

History

Edition	Recommendation	Approval	Study Group	Unique ID*
1.0	ITU-T Y.2775	2019-08-13	13	11.1002/1000/13983

Keywords

Deep packet inspection, functional architecture, future networks.

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

FOREWORD

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

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

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

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

NOTE

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

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

INTELLECTUAL PROPERTY RIGHTS

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

As of the date of approval of this Recommendation, ITU had not received notice of intellectual property, protected by patents, 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 TSB patent database at http://www.itu.int/ITU-T/ipr/.

Table of Contents

Recommendation ITU-T Y.2775

Functional architecture of deep packet inspection for future networks

1 Scope

This Recommendation specifies the functional architecture of deep packet inspection (DPI) for future networks (e.g., software-defined networking, network function virtualization). The scope of this Recommendation includes:

- general DPI functional architecture aspects related to future networks;
- DPI functional architecture for software-defined networking;
- DPI functional architecture for network function virtualization;
- DPI functional architecture for service function chaining and DPI as a service;
- DPI functional architecture for network virtualization;
- DPI functional architecture for evolving mobile network.

Implementers and users of these described techniques shall comply with all applicable national and regional laws, regulations and policies. The mechanisms described in this Recommendation may not be applicable to the international correspondence in order to ensure the secrecy and sovereign national legal requirements placed upon telecommunications, but they shall comply with the ITU Constitution and Convention.

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 X.200] Recommendation ITU-T X.200 (1994) | ISO/IEC 7498-1 (1994), *Information technology – Open Systems Interconnection – Basic Reference Model: The basic model*.

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

[ITU-T Y.2770] Recommendation ITU-T Y.2770 (2012), *Requirements of deep packet inspection in next generation networks*.

[ITU-T Y.2771] Recommendation ITU-T Y.2771 (2014), *Framework for deep packet inspection*.

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

[ITU-T Y.3300] Recommendation ITU-T Y.3300 (2015), *Framework of software-defined networking*.

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1 deep packet inspection (DPI) [ITU-T Y.2770]: Analysis, according to the layered protocol architecture OSI-BRM [ITU-T X.200], of

- payload and/or packet properties (see list of potential properties in clause 3.2.11 of [ITU-T Y.2770]) deeper than protocol layer 2, 3 or 4 (L2/L3/L4) header information, and
- other packet properties

in order to identify the application unambiguously.

NOTE – The output of the DPI function, along with some extra information such as the flow information, is typically used in subsequent functions such as reporting or actions on the packet.

3.1.2 DPI analyser [ITU-T Y.2771]: A subsequent entity in the DPI processing path (within a DPI policy enforcement function) with focus on comparison functions between the particular packet headers and payloads of preselected packet flows. The primary scope of the DPI **analyser** is related to the evaluation of DPI policy *conditions* against *preselected* incoming packets.

NOTE – The DPI analyser may be located after a DPI scanner (see clause 3.2.5 of [ITU-T Y.2771]). The DPI analyser may provide the functionality of an intrusion detection system (IDS) analyser.

3.1.3 DPI engine [ITU-T Y.2770]: A subcomponent and central part of the DPI functional entity which performs all packet path processing functions (e.g., packet identification and other packet processing functions in Figure 6-1 of [ITU-T Y.2770]).

3.1.4 DPI node [ITU-T Y.2771]: A network element or device that realizes the DPI related functions. It is thus a generic term used to designate the realization of a DPI physical entity.

NOTE – Functional perspective: the DPI node function (DPI-NF) comprises the DPI policy enforcement function (DPI-PEF) and the (optional) local policy decision function (L-PDF), hence, the DPI-NF is functionally equal to the DPI functional entity.

3.1.5 future network (FN) [b-ITU-T Y.3001]: A network able to provide services, capabilities, and facilities difficult to provide using existing network technologies. A future network is either:

a) A new component network or an enhanced version of an existing one, or

b) A heterogeneous collection of new component networks or of new and existing component networks that is operated as a single network.

3.1.6 metadata [b-IETF RFC 7665]: Provides the ability to exchange context information between classifiers and SFs, and among SFs.

3.1.7 service function (SF) [b-ITU-T Y Suppl. 41]: A function, specifically representing network service function, that is responsible for specific treatment of received packets other than the normal, standard functions of an IP router (e.g., IP forwarding and routing functions) on the network path between a source host and destination host.

NOTE – The examples of service function are similar to, but not limited to that of a middlebox.

3.1.8 service function chain [b-ITU-T Y Suppl. 41]: A chain that defines an ordered set of abstract service functions and ordering constraints that must be applied to packets and/or frames and/or flows selected as a result of classification and/or policy.

3.1.9 service function chaining [b-ITU-T Y Suppl. 41]: A mechanism of building service function chains and forwarding packets/frames/flows through them.

3.2 Terms defined in this Recommendation

None.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

2G	Second Generation
3G	Third Generation
4G	Fourth Generation
5G	Fifth Generation
API	Application Programming Interface
BSC	Base Station Controller
BTS	Base Transceiver Station
DNNF	Determining Next Node Function
DPI-AnF	Deep Packet Inspection Analyser Function
DPI-AcEF	Deep Packet Inspection Action Execution Function
DPI	Deep Packet Inspection
DPIaaS	DPI as a Service
DPI-NF	DPI Node Function
DPI-PEF	DPI Policy Enforcement Function
DPI-PDF	DPI Policy Decision Function
DPI-PIB	DPI Policy Information Base
DPI-ScF	Deep Packet Inspection Scan Function
FIB	Forwarding Information Base
GGSN	GPRS Support Node
GPRS	General Packet Radio Service
HSS	Home Subscriber Server
IETF	Internet Engineer Task Force
IMT	International Mobile Telecommunication
IP	Internet Protocol
L2	Layer 2
L3	Layer 3
L4	Layer 4
L7	Layer 7
L-PDF	Local Policy Decision Function
LTE	Long Term Evolution
MANO	Management and Orchestration
MME	Mobile Management Entity
NFV	Network Function Virtualization
NFVI	Network Function Virtualization Infrastructure
NMS	Network Management System
NSH	Network Service Header
PCRF	Policy and Charging Rules Function

PDF Policy Decision Function

PFF Packet Forwarding Function

P-GW Packet data network Gateway

PIB Policy Information Base

QoS Quality of Service

R-PDF Remote Policy Decision Function

RAN Radio Access Network

RNC Radio Network Controller

SaaS Software as a Service

SBC Session Border Controller

SDN Software-Defined Networking

SFC Service Function Chaining

SGSN Serving GPRS Support Node

S-GW Serving-Gateway

TCO Total Cost of Ownership

TDF Traffic Detection Function

VNF Virtual Network Function

VGW Virtual Gate Way

VMME Virtual Machine Management Entity

vDPI virtual Deep Packet Inspection

5 Conventions

In the body of this Recommendation, the words shall and should may sometimes appear, in which case they are to be interpreted, respectively, as is required to and is recommended.

6 General DPI functional architecture for future networks

Figure 6-1 depicts a general DPI functional architecture for future networks, and this general DPI functional architecture is based on the general, high-level functional model defined in [ITU-T Y.2771] (see Figure 7-2 of [ITU-T Y.2771]).

There are two parts in the packet node, the DPI node function (DPI-NF) and the packet forwarding function (PFF). The DPI-NF comprises the DPI policy enforcement function (DPI-PEF) and the local-policy decision function (L-PDF). Hence, the DPI-NF is functionally equal to the DPI functional entity; The PFF comprises the determining next node function (DNNF) and the forwarding information base (FIB).

As shown in Figure 6-1, the packet flow passes through the DPI-PEF first and then enters the DNNF. In the DPI-PEF, the packet flow is processed by two stages. The first stage is the evaluation of policy conditions which is aimed for the packet flow identification. The second stage is the evaluation of policy actions, in which the packet flow is handled and transferred to the PFF.

A DPI functional architecture should be designed coincidently with other different types of network architectures. As a result, in the DPI-NF, there are some external interfaces to exchange information with other entities: DPI policy decision function entity [ITU-T Y.2770], or other policy decision

function entities (e.g., network controller). The external interfaces are allowed to make appropriate changes depending on the specific requirements of future networks.

The architecture described in Figure 6-1 is a basic DPI functional architecture for future networks and the architecture shall be modified if it is needed to match different types of network architecture.

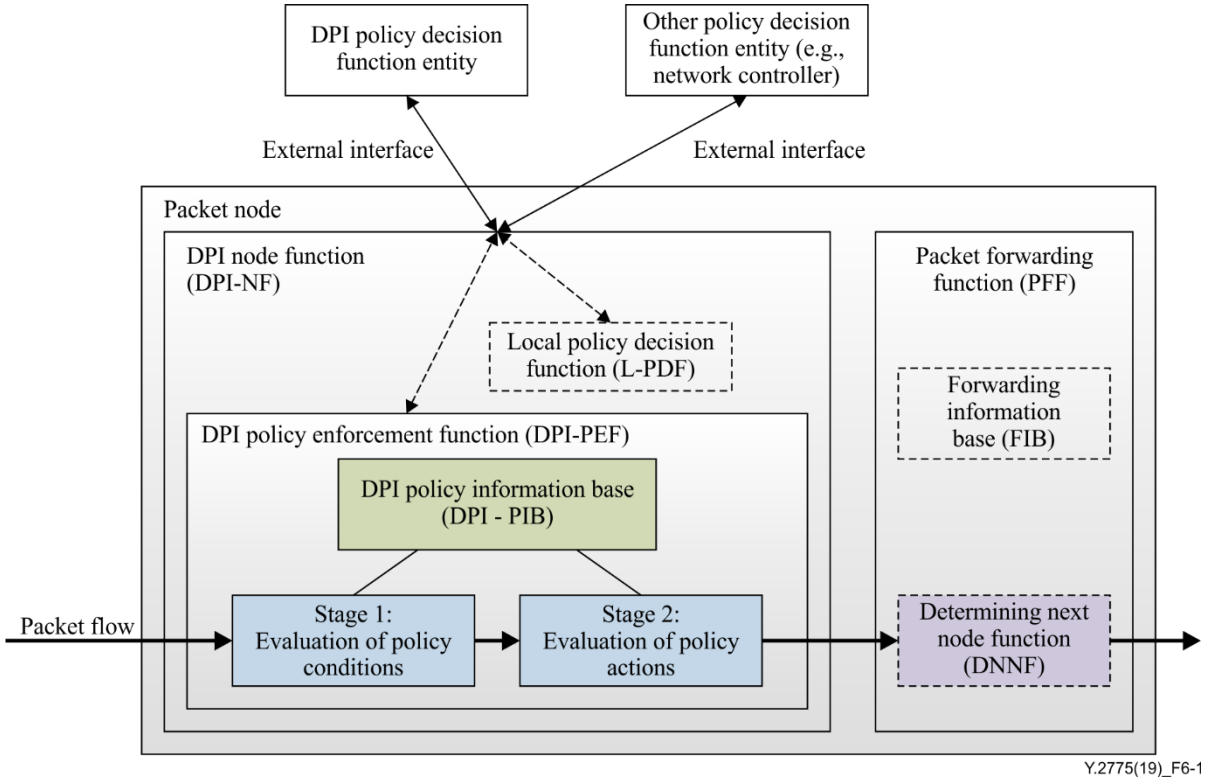


Figure 6-1 – General DPI functional architecture for future networks

7 DPI functional architecture for SDN

7.1 Overview of software-defined networking

As a typical future network technology or architecture, software-defined networking (SDN) has two basic principles:

- to create a layered network architecture in which all control functions are separated from network forwarding functions;
- to make networking functions available as programmable resources, via a logically centralized "controller," connected via standardized interfaces (typically application programming interfaces (APIs)) to "applications". Centralization of the controller is a core principle, providing an end-to-end view of the network to users and applications. The above basic principles are described visually in Figure 7-1 (see also Figure 11-1 of [ITU-T Y.3300]).

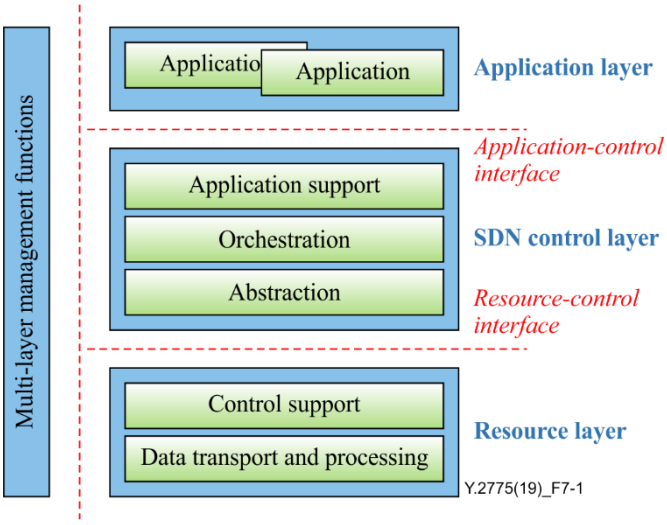


Figure 7-1 – High-level architecture of SDN architecture

7.2 General functional architecture of DPI for SDN

DPI functions deployed in SDN have different functional architecture from DPI functions deployed in the traditional network ecosystem.

Application identification is critical for providing visibility, quality of service (QoS), billing, and security. Application-awareness is even more important within SDN. Current SDN APIs are capable of Layer 2/3/4-based policy enforcement but they currently lack higher layer application awareness. Figure 6-1 depicts the general high-level functional architecture of DPI. Figure 7-1 depicts the high-level architecture of SDN. However, this DPI model architecture cannot directly be used in SDN architecture without modifications.

A DPI node is also a network entity, and it should be designed coincident with the above SDN architecture.

In SDN context, DPI functions could be deployed in the resource layer, or they could be deployed in both the resource layer and the control layer. The general functional architecture of DPI for SDN is given as Figure 7-2. Figure 7-2 shows that basic DPI functions are implemented in the resource layer, while the DPI policy decision function (DPI-PDF) is implemented in the control layer.

DPI resource layer implements the basic DPI functions as well as control support functions, and control support functions provide service for the control layer.

DPI control layer can be divided into three functions:

1) DPI application support function;

2) DPI-PDF;

3) DPI abstract function.

DPI application support function is used to connect with the upper application layer, while DPI abstract function can connect the lower resource layer. It is obvious that DPI-PDF is used for policy or rule control.

The DPI applications (network service, function or entity that is implemented based on DPI functions, e.g., firewalls) are mapped to the application layer of SDN. These DPI applications interact with the control layer via application-control interfaces, in order for the control layer to automatically customize the behaviour and properties of network resources. The programming of a DPI application makes use of the abstracted view of the network resources provided by the control layer via the application-control interface. In reality, the application-control interface is implemented through the DPI application support function.

The DPI application-support function is the DPI extension of SDN application support which provides application-control interface for DPI applications to access network information and customize DPI-specific network behaviours.

According to [ITU-T Y.2771], the DPI-PDF includes session dependent PDF (S_D-PDF) and session independent PDF (S_I-PDF) which are mapped to the SDN orchestration and multilayer management function, respectively. The DPI-PDF provides the automated control and management of DPI rules and coordination of requests from the DPI applications. DPI-PDF function is one of the core functions of orchestration sublayer of control layer.

The DPI abstraction function interacts with network resources, and provides an abstraction of the physical and virtual DPI resources in the network, including network DPI capabilities and characteristics (e.g., network topology, DPI-PIB) to support DPI management and orchestration of physical and virtual network resources.

The DPI L-PDF within the DPI resource layer is mapped to the SDN control support function. The L-PDF interacts with the DPI-PDF function through the DPI abstraction and supports programmability via resource-control interfaces.

The DPI policy enforcement function (DPI-PEF) is mapped to the SDN data transport and processing function. The DPI-PEF is part of the SDN data-forwarding functionalities. In the DPI-PEF, the packet flow is processed by three serialized functions:

1) DPI scan function (DPI-ScF);
2) DPI analyser function (DPI-AnF); and
3) DPI action execution function (DPI-AcEF).

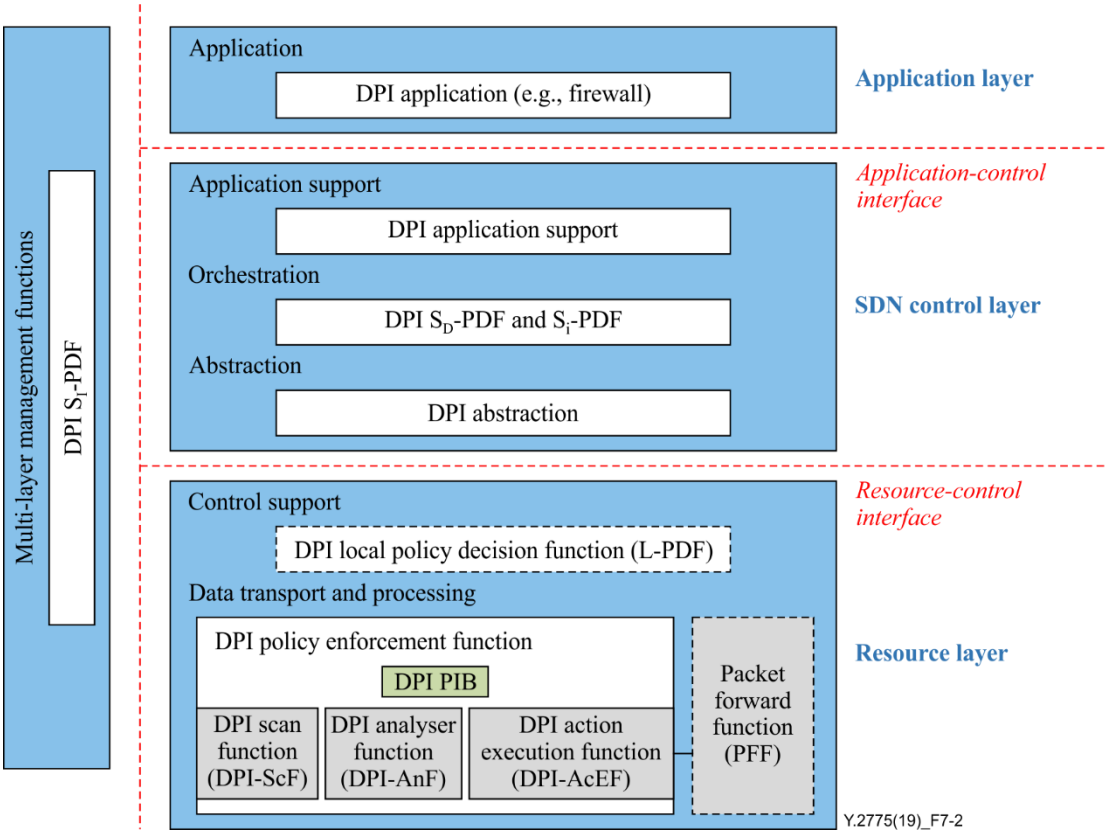


Figure 7-2 – General functional architecture of DPI for SDN

7.3 Functional model for DPI-PEF and PFF within a physical entity in the SDN context

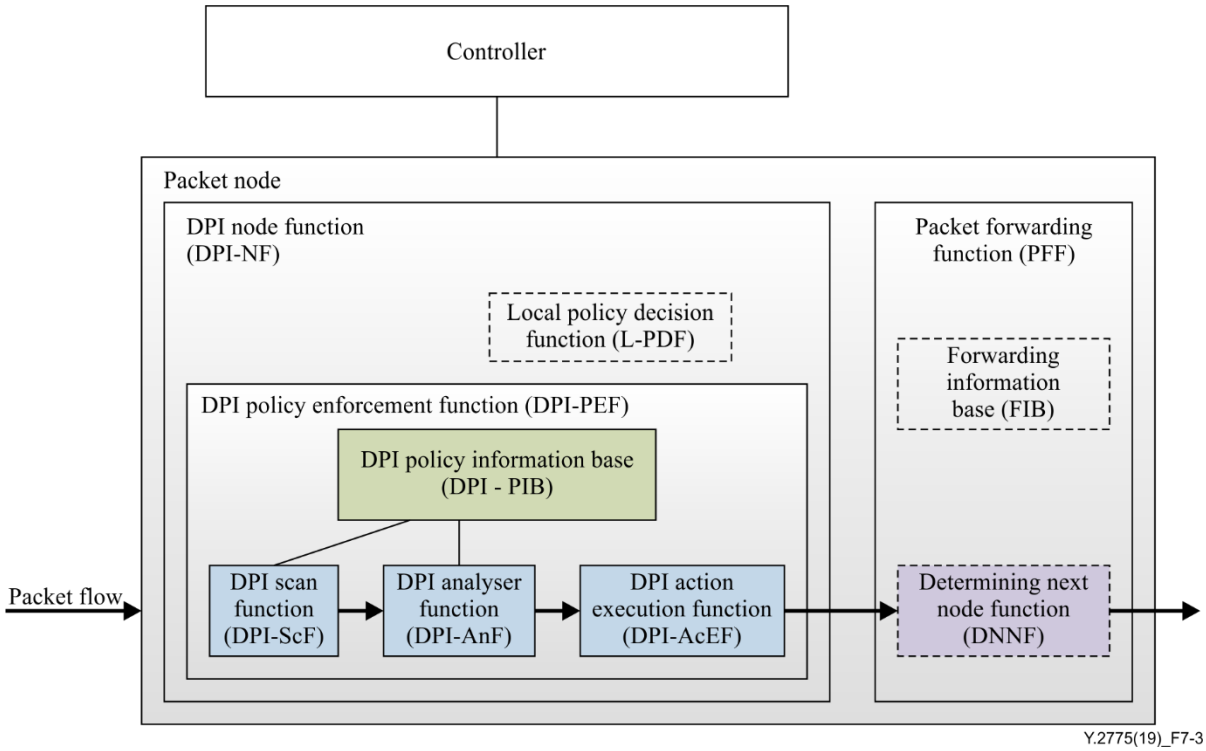


Figure 7-3 – Model for a physical entity including both PEF and PFF

If the DPI function is embedded in a physical network device, then the PEF and PFF are designed in a single physical network. Under such circumstance, a single controller is used to control both PEF and PFF, and Figure 7-3 depicts such a model.

7.4 Functional model for DPI-PEF and PFF within different physical entity in the SDN context

If the DPI function is deployed as an independent network device, the DPI entity is possible not to include a PFF, in other words, PEF and PFF belongs to different physical network devices. Because of separation of PEF and PFF, either a single controller or double controllers can be used to control the above network devices.

7.4.1 Single-controller model

Figure 7-4 describes the single-controller mode. Within the above controller, PEF-control function and PFF-control function can be design as independent and mutual-connected logical entities.

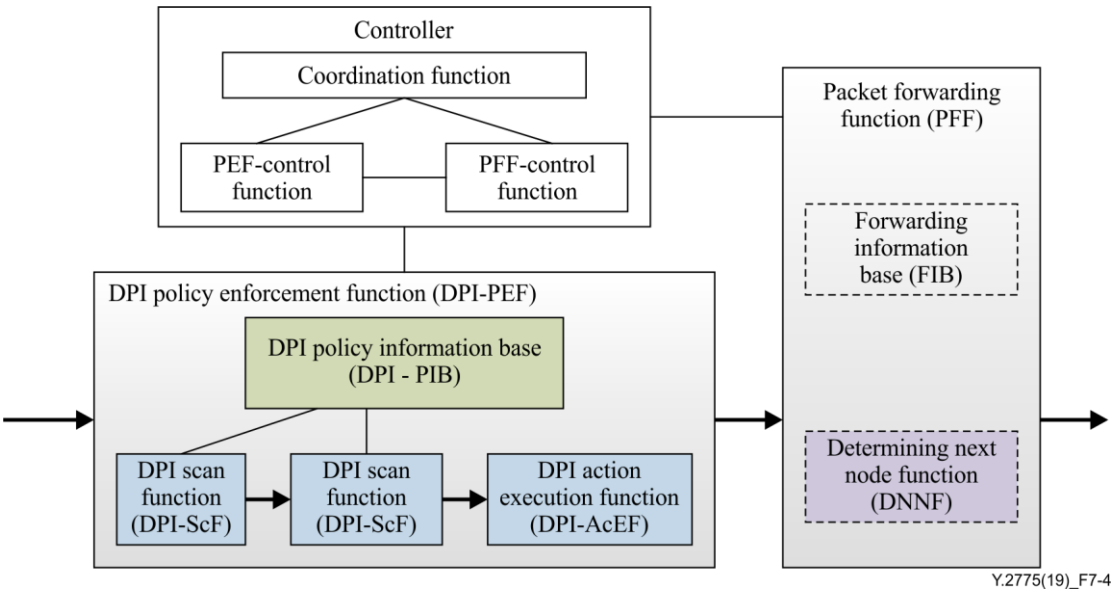


Figure 7-4 – PEF and PFF belong to different physical entities with single controller

7.4.2 Double-controller model

Figure 7-5 describes the double-controller mode. Controller-PEF and Controller-PFF should cooperate to realize the control functions of the related physical devices.

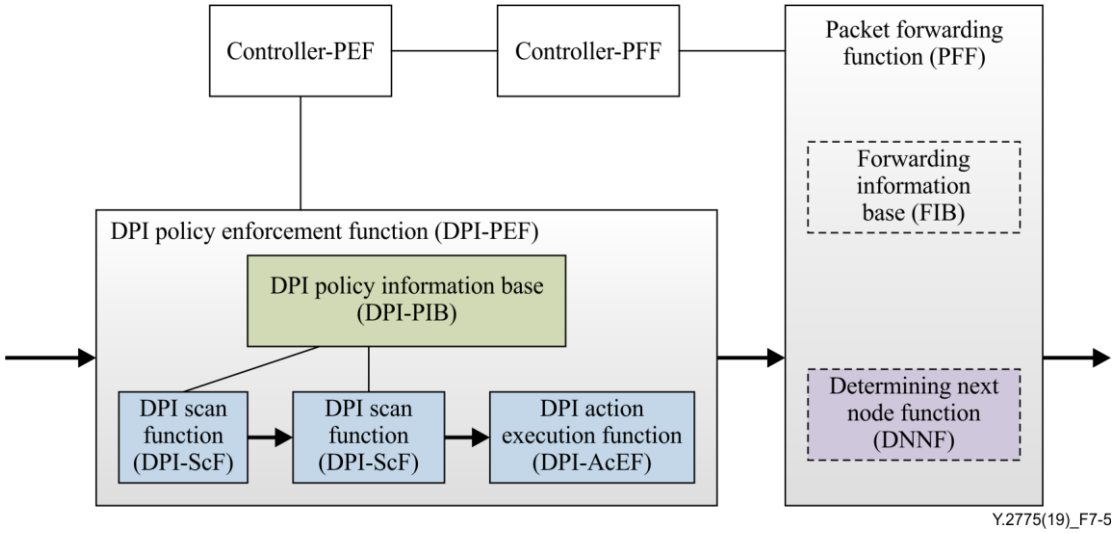


Figure 7-5 – PEF and PFF belong to different physical entities with different controller

7.5 Functional model for bidirectional DPI in the SDN context

7.5.1 General functional model for bidirectional DPI

A general bidirectional DPI functional model is depicted in Figure 7-6. Without losing generalization, the two independent DPI nodes cooperate to handle the bidirectional traffic. One of them handles a direction of packet traffic, while the other controls another direction of packet traffic. Meanwhile a network management system (NMS) server can manage the above DPI nodes if there are logical links between the NMS server and the DPI nodes.

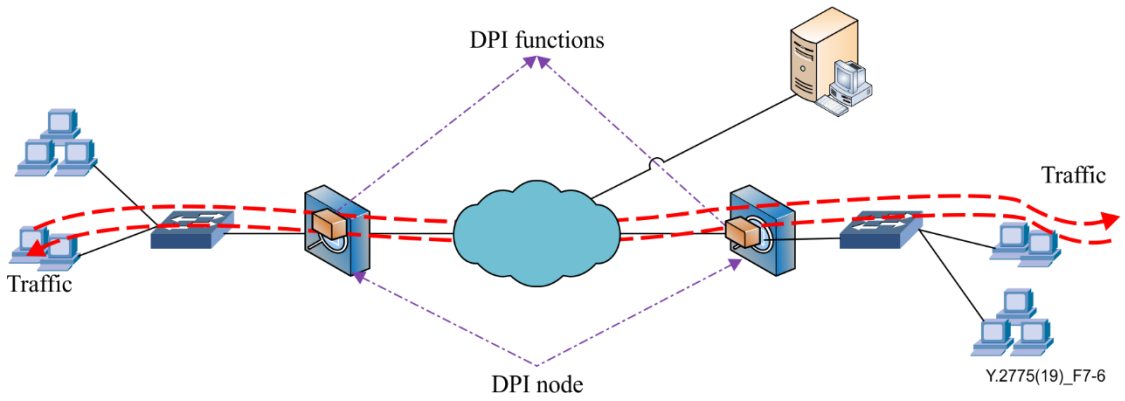


Figure 7-6 – General DPI bidirectional DPI functional model

7.5.2 Single-controller functional model for bidirectional DPI in SDN context

In SDN context, the bidirectional DPI functional model is basically different. The single-controller functional model for bidirectional DPI in SDN context is introduced first, and Figure 7-7 describes the above single-controller functional model.

If the two DPI nodes that cooperate to realization the bidirectional DPI functions can be controlled by an identical controller, then single-controller functional model can be used.

In Figure 7-7, a single controller controls the two DPI nodes while NMS server manages the above DPI nodes through the above controller.

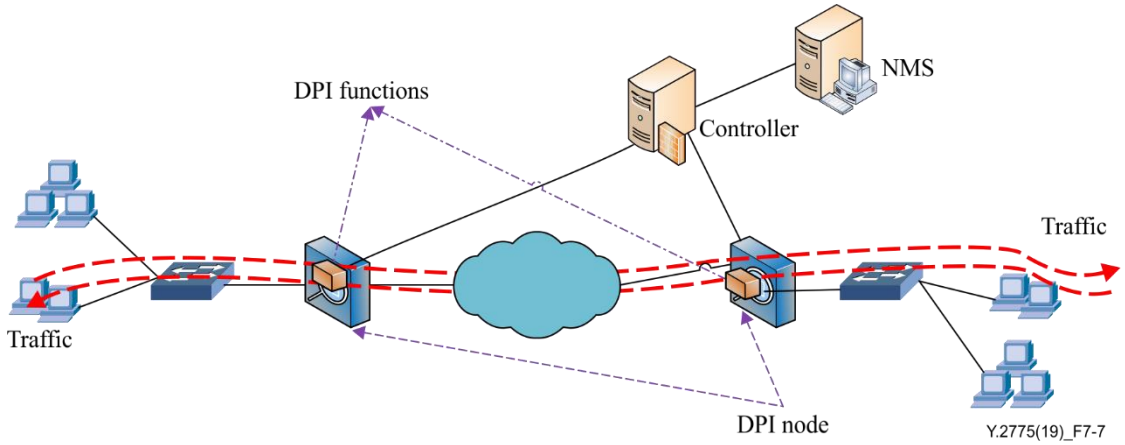


Figure 7-7 – Single-controller bidirectional DPI functional model in SDN context

7.5.3 Multi-controller functional model for bidirectional DPI in SDN context

The multi-controller functional model for bidirectional DPI in SDN context is theoretically different from the single-controller functional model that is described in clause 7.5.2, and Figure 7-8 describes the above multi-controller functional model.

If the two DPI nodes that cooperate to realization the bidirectional DPI functions are controlled by different controllers, then multi-controller functional model can be used.

In Figure 7-8, two controllers control the two DPI nodes respectively. Meanwhile, a NMS server manages the above DPI nodes through the above two controllers. In addition, the two controllers can exchange information (e.g., information about topology and information about policy rule) if needed.

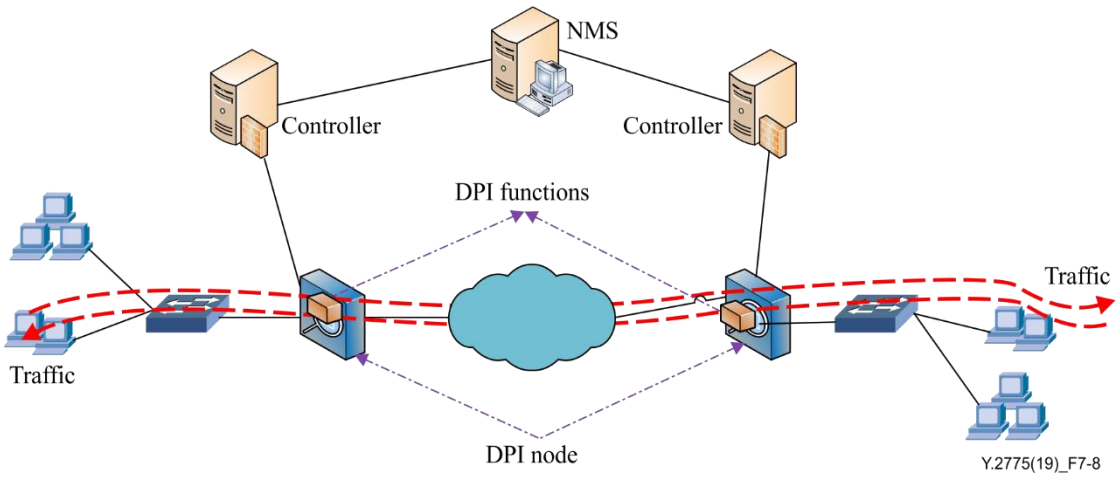


Figure 7-8 – Multi-controller bidirectional DPI functional model in SDN context

8 DPI functional architecture for network function virtualization

8.1 Overview of NFV

Network function virtualization (NFV) is network technology or architecture which is much different from the current network.

NFV aims to decouple software from hardware and virtualizes network functions so they do not have to depend on special hardware, and new functions and services can be deployed through software installation and upgrades. NFV helps carriers increase network flexibility, realize highly efficient network construction and operation, expedite service provisioning, and reduce total cost of ownership (TCO).

It is learned that NFV envisages the implementation of network functions (NFs) as software-only entities that run over the NFV infrastructure (NFVI) from [b-NFV-Framework]. Figure 8-1 (see also Figure 1 of [b-NFV-Framework]) illustrates the high-level NFV framework. As such, three main working domains are identified in NFV:

1) virtualised network function, as the software implementation of a network function which is capable of running over the NFVI;

2) NFVI, including the diversity of physical resources and how these can be virtualised. NFVI gives support for implementation of various VNFs;

3) NFV management and orchestration (MANO), which covers the orchestration and lifecycle management of physical and/or software resources that support the infrastructure virtualisation, and the lifecycle management of VNFs.

NFV MANO focuses on all virtualisation-specific management tasks necessary in the NFV framework.

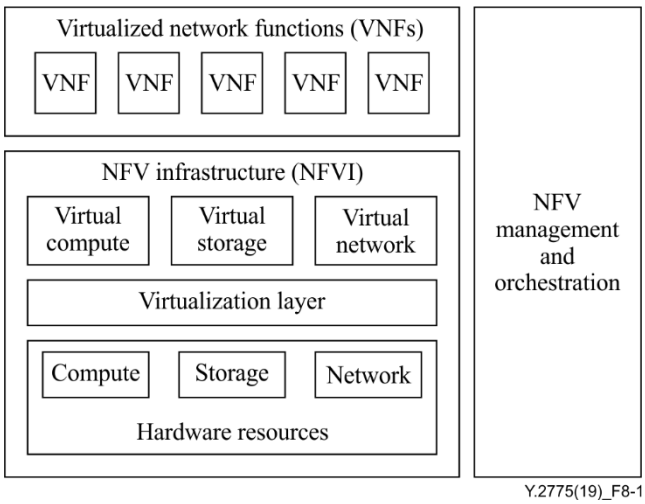


Figure 8-1 – High-level NFV framework

8.2 General model of DPI for NFV

8.2.1 Traditional DPI vs DPI in NFV context

Traditionally, DPI functions can be deployed in the current network through several different ways. One of the typical DPI deployments for current network is embedding the DPI functions into various network devices such as session border controller (SBC), traffic detection function (TDF) and gateway general packet radio service (GPRS) support node (GGSN), etc. The main disadvantage of this deployment is that high costs will be incurred by implementing DPI functions on different hardware platforms and interworking between different applications is more difficult. The left half of Figure 8-2 describes the above deployment mode. Nevertheless, the aforementioned disadvantage should be paid attention to.

It is much different for DPI functions to be deployed in an NFV-based network. In NFV context, DPI functions can be transited from various network devices into which the DPI functions are embedded to the standard server and become common-build function blocks which are executed in VMs. The right side of Figure 8-2 depicts this deployment mode.

NOTE – Based on this deployment mode, cost and power consumption can be reduced and the interworking between different applications can be implemented more flexibly.

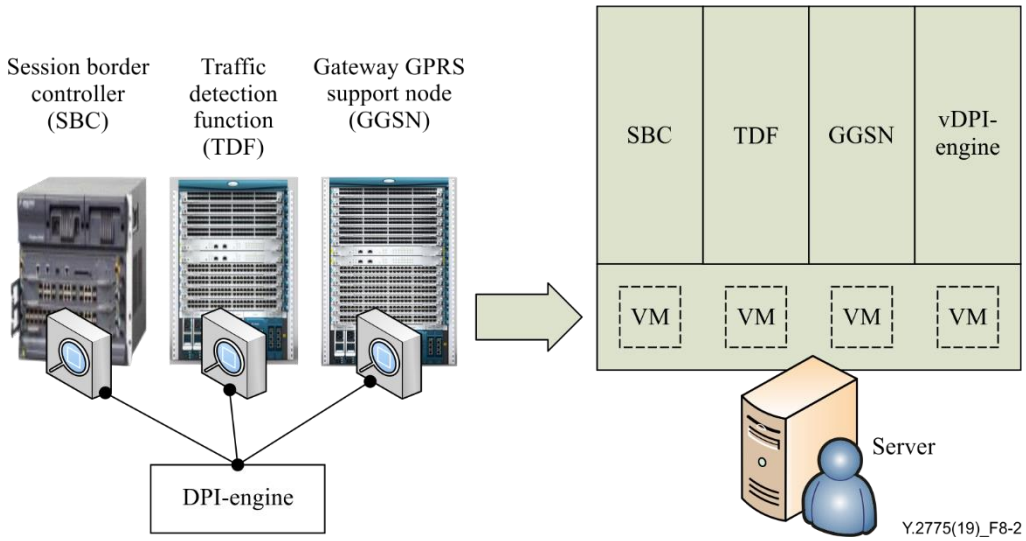


Figure 8-2 – Deployment mode for traditional DPI and DPI in NFV

8.2.2 Functional model for DPI in NFV context

Figure 8-3 describes a general functional model of DPI in NFV context. Within NFV context, a DPI node (DPI entity) can be seen as a virtual network entity named virtual DPI (vDPI) just like virtual gateway (VGW) or virtual machine management entity (VMME) in NFV.

Similar to a physical network entity, a vDPI can receive and transmit data packets from other network entities or from the other network. After packets enter the vDPI, they will be processed by the core functional component, vDPI-engine. As shown in Figure 8-3, the vDPI-engine also has three subfunction components: DPI-ScF, DPI-AnF and DPI-AcEF. After being processed by the vDPI, packets will be transmitted to the PFF to be further processed.

In the course of packet processing, a vDPI node can either apply a pre-defined policy directly or send this information to a network entity (e.g., an outside controller or a management entity) or a network application which can implement the remote policy decision functions, and then receive the policies or rules. In this way, the L-PDF should have the ability to exchange the information with NFV manager and NMS such as EMS.

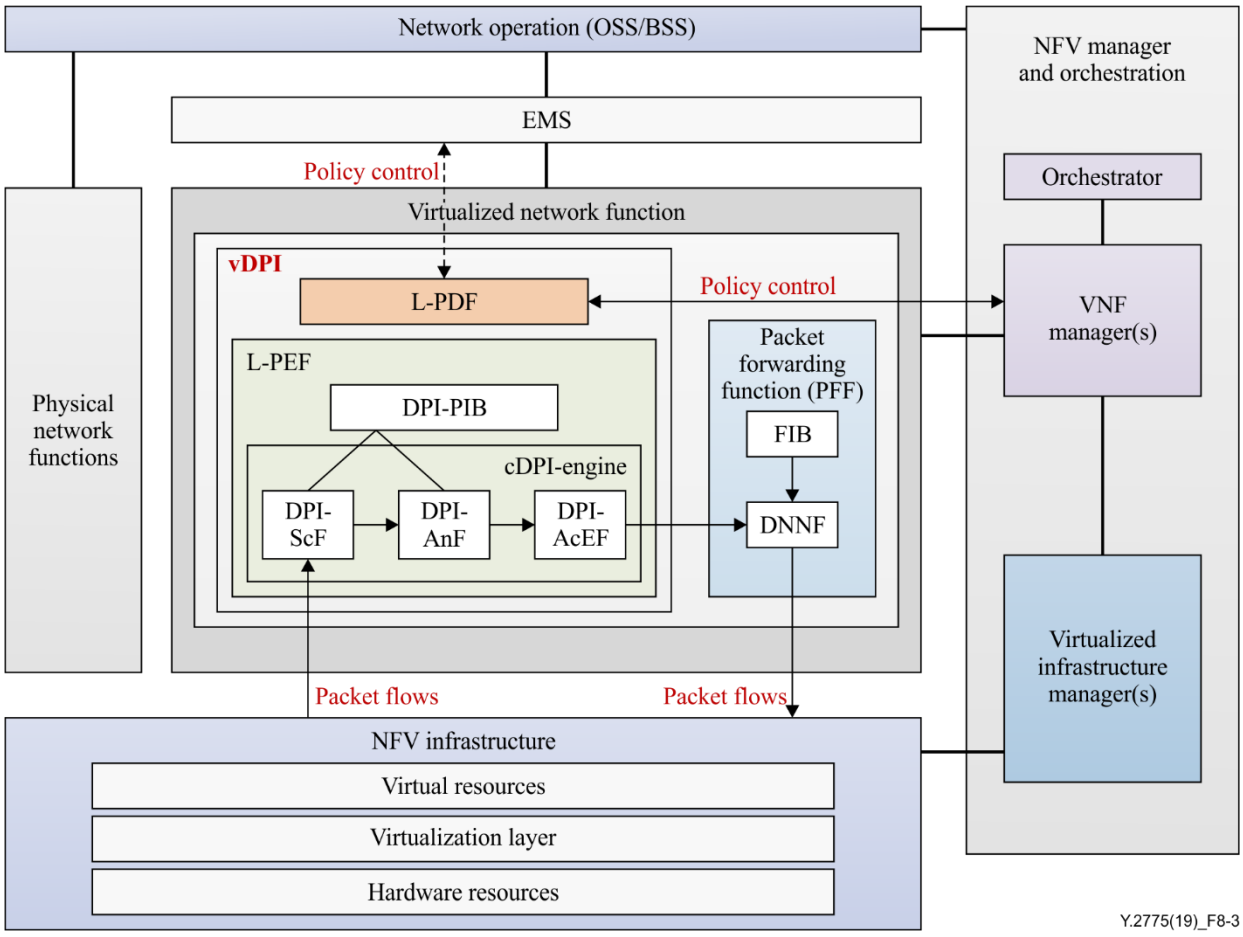


Figure 8-3 – Deployment mode for traditional DPI and DPI for NFV

8.3 Functional architecture for vDPI

Generally, in NFV context, a vDPI entity is carried out by one or more VM, in other words, vDPI functions are mapped into one or more VNFs.

There are several typical functional architectures for mapping between vDPI entities and VMs, in other words, mapping between vDPI functions and VNFs.

8.3.1 Single VMs implements a vDPI entity

This is the simplest and general method to realize vDPI entities, the entire vDPI entity is implemented by a single VM. Figure 8-4 describes the method where a single VM implements a vDPI entity.

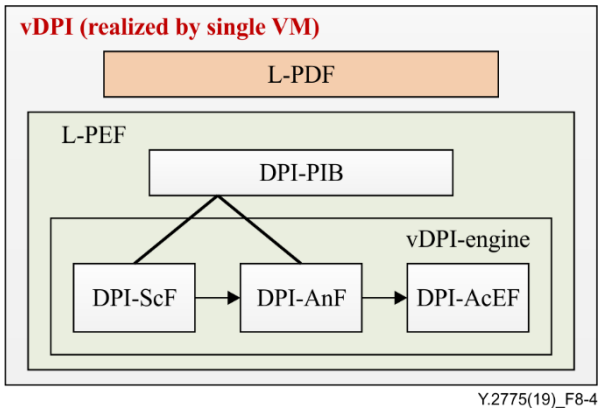


Figure 8-4 – A single VM implements a vDPI entity

8.3.2 Multiple VMs implement a vDPI entity and a single VM implements a vDPI-engine

This is the typical distributed method to realize the vDPI entities, the vDPI entity is realized by several VMs, while the vDPI-engine is carried out by a single VM. Figure 8-5 describes this method. In Figure 8-5, three VMs are used to realize a vDPI entity. VM1-VM3 can be designed in the identical physical machine or different physical machines.

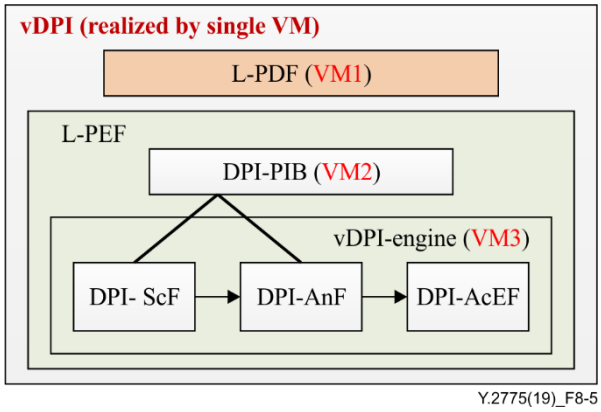


Figure 8-5 – Multiple VMs realize a vDPI entity while a single VM realizes a vDPI-engine

8.3.3 Multiple VMs implement a vDPI entities and multiple VMs implement a vDPI-engine

This is the appropriate method to realize the vDPI entities with the best extensibility and flexibility. When using this method, the vDPI entity is realized by several coordinated VMs, in addition, the vDPI-engine is also implemented by multiple coordinated VMs. Moreover, the DPI-ScF, DPI-AnF and DPI-AcEF are also carried out by multiple coordinated VMs. Figure 8-6 describes this method. In Figure 8-6, The DPI-ScF is realized by VM3-VMi, The DPI-AnF is realized by VMi-VMj, and the DPI-AcEF is realized by VMj-VMk. VM1-VMk can be designed in the identical physical machine or different physical machines.

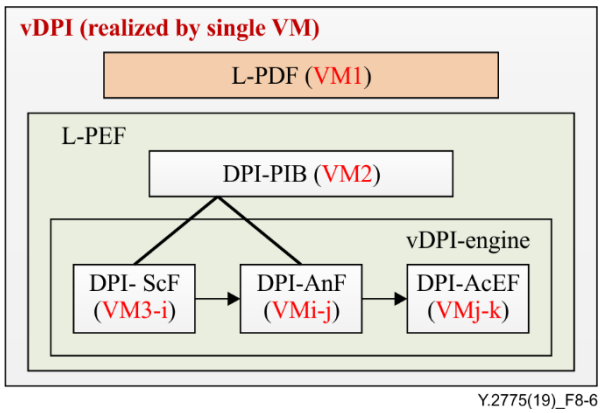


Figure 8-6 – Multiple VMs realize a vDPI entity meanwhile multiple VMs realize a vDPI-engine

9 DPI functional architecture for service function chaining and DPI as a service

9.1 Overview of service function chaining

Service function chaining (SFC) is a new network architecture under research in Internet Engineer Task Force (IETF) [b-IETF RFC 7665]. It is an emerging set of technologies and methods that enable operators to configure network services dynamically in software without having to make changes to the infrastructure of the network. Service function chaining is also called service chaining.

A service function chain defines an ordered or partially ordered set of abstract service functions (SFs) and their ordering constraints that must be applied to packets, frames, and/or flows selected as a result of classification. Figure 9-1 depicts a basic functional architecture of service function chain. There are three important parts of the architecture:

1) classifier: the ability to identify and then classify traffic in order to direct flows into a service function chain is critical. There are several ways to do this:
   - simply directing any traffic to;
   - from a particular destination to a particular service path;
   - a more sophisticated policy-driven scheme.

   In the SFC architecture, there is specifically the packet header definitions: the network service header (NSH). The NSH can carry metadata for managing the chain. Control plane and NSH are basic and important components of SFC architecture;

2) control plane: the SFC control plane includes a topology server and a policy decision point. The topology server talks to the ingress and egress nodes (the classifiers) and locates service function nodes in the network. The policy decision point communicates with the service functions and is a central control/management plane entity used to maintain SFC policy tables;

3) NSH: a service-level data plane encapsulation format (i.e., a new packet header) that specifies the sequence of service functions that make up a service chain. This header format can also be used to communicate context information between nodes that implement service functions.

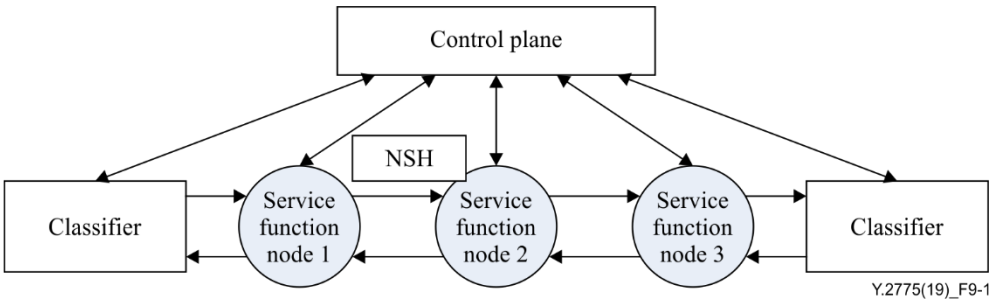


Figure 9-1 – Service chaining architecture

At the ingress to the service function chain, a classifier identifies and classifies traffic before forwarding it into the processing path. The path itself is set up and managed by the control plane.

Network service plays a major role in contemporary networks, as forwarding packets is often not enough to meet operator demands, and other functionalities (such as security, QoS/QoE provisioning, and load balancing) are required. Traffic is usually routed through a service chain, which either resides across the network or in a single, consolidated location.

9.2 Introduction of DPI as a service

Although a vast range of different capabilities can be provided based on service function chaining, there are some functions or functional entities that can be shared among service functions.

DPI is a typical function common to almost all service functions included in a service function chain. DPI can deal with protocols from layer 2 (L2) to layer 7 (L7). Generally, network traffic needs to be scanned and analysed from beginning to end by all service functions of the service function chain. In other words, if there are two or more service functions in the service function chain, the network traffic is possibly scanned and analysed twice or more. However, the duplication of service functions scanning and analysing identical traffic twice or more can be avoided by deploying the DPI function in DPI as a service (DPIaaS) mode.

DPIaaS can be thought of as implementing DPI functions in software as a service (SaaS) mode, similar to how other functions are implemented in an SaaS mode. DPIaaS implements DPI functions in the cloud and provides the aforementioned functions to clients that need those functions. In the service chain context, DPIaaS implies that traffic can be scanned and analysed only once and that the scanning and analysis results of DPIaaS can meet the requirements of all service functions in the service chain. The DPI service thus passes the scanning and analysis results to all appropriate service function instances. Deploying the DPI service in this way has significant advantages in performance, scalability and robustness, and is a catalyst for innovation in the service chaining domain.

9.3 DPI functional architecture for DPIaaS in service function chaining

Figure 9-2 describes a typical architecture of DPIaaS in a service chain. This architecture requires a DPI controller and one or more DPI service nodes providing the DPI service for the service node in service chaining. All service nodes can make direct use of the scanning and analysis results from the DPI service node without reiterating the DPI's function of scanning and analysing packets.

In this manner, the network could avoid repeating DPI-related functions. It is thus possible to decrease network resource consumption as well as improve utilization of network devices and functions.

The DPI controller is a logically centralized entity whose role is to control the DPI functions across the network and to communicate with the SDN controller to realize the appropriate data plane actions. Logically, the DPI controller can reside at the SDN application layer on top of the SDN controller (see Figure 9-2), or the DPI controller can also be a component of SDN controller (i.e., in

Figure 9-2, the DPI controller can be merged with the SDN controller). Physically, the DPI controller and the SDN controller can optionally be designed within an identical physical device.

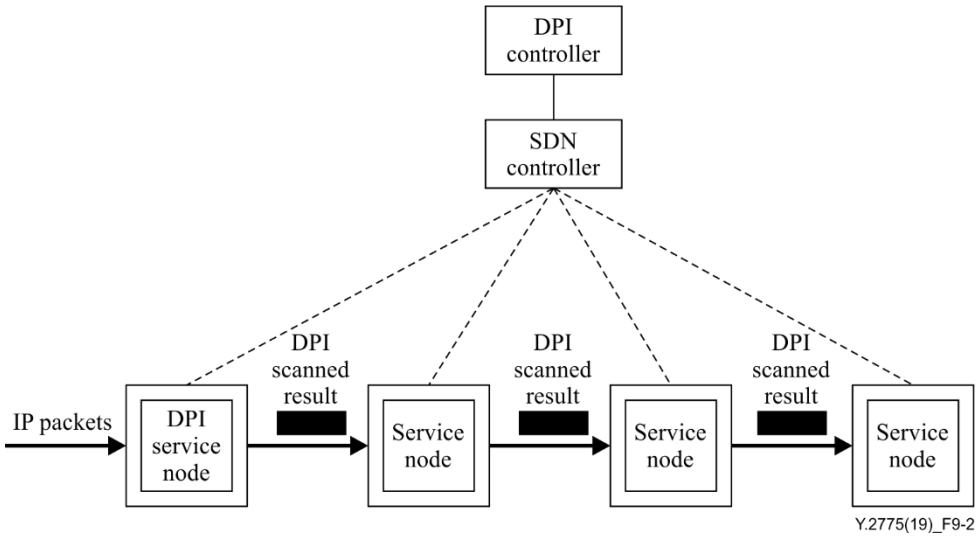


Figure 9-2 – DPIaaS in service chaining

Two kinds of procedures take place between the DPI controller and the service nodes, namely: registration and pattern (signature, condition) set management. The first task of the DPI controller is to register service nodes that use its service. Specifically, a service node registers itself to the DPI service using a registration message. The address (e.g., IP address) of the DPI controller and the service node's unique ID and name are preconfigured.

Generally, the above service node operates by rules that contain actions and conditions that should be matched to activate the actions. The responsibility of the DPI service node is only to scan and analyse packets based upon patterns, while resolving the logic behind a condition and performing the action is the service node's responsibility. The DPI service node only carries out the DPI scanning and analysing function. Patterns are added to and removed from the DPI controller using dedicated messages from service nodes to the DPI controller. The DPI controller maintains a global pattern set with its own internal IDs. If two service nodes register the same pattern (since each has a rule that depends on this pattern), the DPI controller keeps track of each of the rule IDs reported by each service nodes and associates them with its internal ID. For that reason, when a pattern removal request is received, the DPI controller removes the service nodes reference to the corresponding pattern. Only when there are no other service node referrals to that pattern, is the pattern removed.

Another necessary action is passing the pattern match results to the service nodes. A feasible method is adding match results information as an additional layer of information prior to the packet's payload. This allows maximal flexibility and the best performance. Publicly available frameworks such as NSH may be used to encapsulate match results data in an SDN setting.

The downside of this approach is that service nodes that refer to the payload on the service chain should be aware of this additional layer of information. However, if all service nodes that use the DPI functions are grouped and placed immediately after the DPI service instance in the service chain, the last service node can simply remove this layer and forward the original packet.

9.4 DPI functional architecture for extended DPI as a service

DPIaaS can be extended from the service function chain to the general application scenario. Figure 9-3 illustrates an example functional model of extending DPIaaS; the above functional architecture is not limited in service chain domain. Layers are logically classified and all links or relationships between two entities are logical. The DPI part of the model is composed of a DPI

node, a DPI controller and a DPI service instance. The DPI service instance is a logical abstraction of certain DPI functions, i.e., logical DPI function entities. In the model, the DPI controller abstracts the DPI process to other network elements and controls DPI service instances across the network. The DPI service provided by DPI nodes is responsible for scanning and analysing the packets. As a packet is forwarded, each DPI node on its route retrieves the DPI scanning and analysing results instead of performing the DPI task. Therefore, DPI service instances can be deployed all over the network and DPI service in the model is equivalent to that when DPI nodes are deployed locally to a client using DPI services.

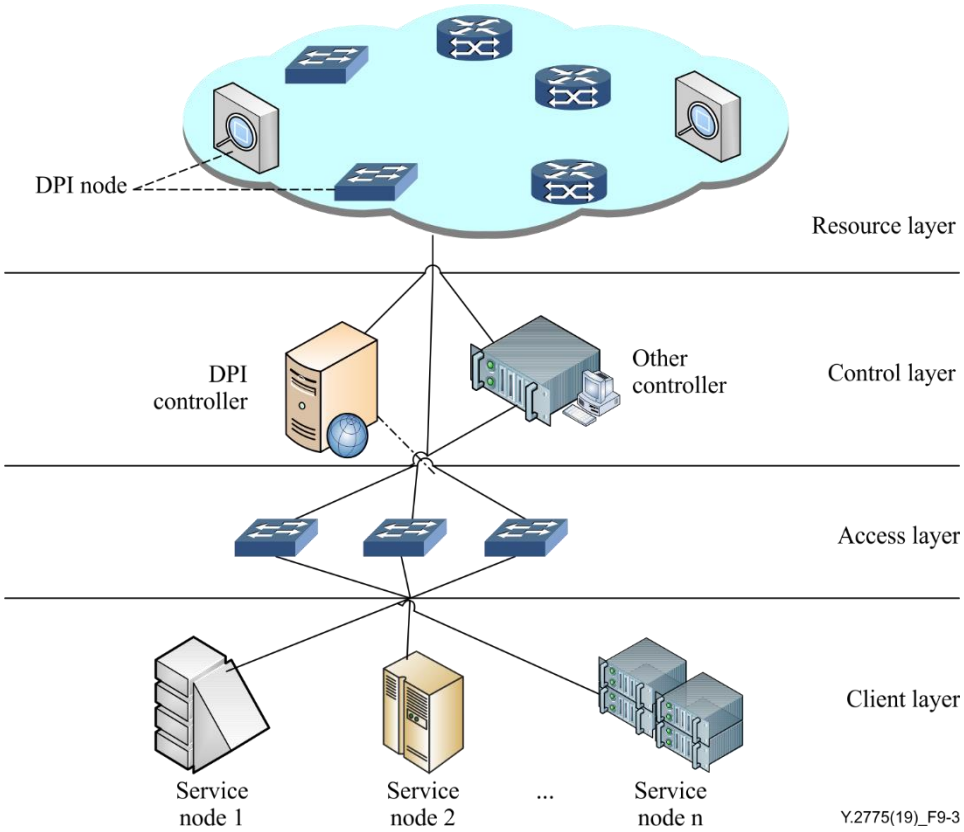


Figure 9-3 – DPIaaS framework illustration

The DPI controller is a logically centralized entity whose role is to control the DPI function or functional entities across the network and to communicate with the other controller to realize the appropriate data plane actions.

The DPI controller is responsible for registering DPI nodes. DPI nodes operate by rules that contain actions and conditions that should be satisfied to activate the actions. In addition, the DPI controller is also responsible for initializing DPI service instances, deployment of different DPI service instances across the network and for advanced features that require a network-wide view.

10 DPI functional architecture in network virtualization environment

10.1 Overview of network virtualization

Network virtualization is a technology that enables the creation of logically isolated network partitions over shared physical networks such that a heterogeneous collection of multiple virtual networks can simultaneously coexist over shared networks. This includes the aggregation of multiple resources in a provider appearing as a single resource [ITU-T Y.3011].

A typical example of network virtualization can be seen in Figure 10-1. In Figure 10-1, a physical network is virtualized to several logical networks (virtual network). Although virtualized logical networks in Figure 10-1 are homogeneous, virtualized logical networks based on the identical physical network can be homogeneous or heterogeneous.

The relationship between the physical network and the virtual network is not limited to 1:n, and several physical networks can also be virtualized to a virtual network. Mapping between physical networks and virtual networks lies in requirements and design.

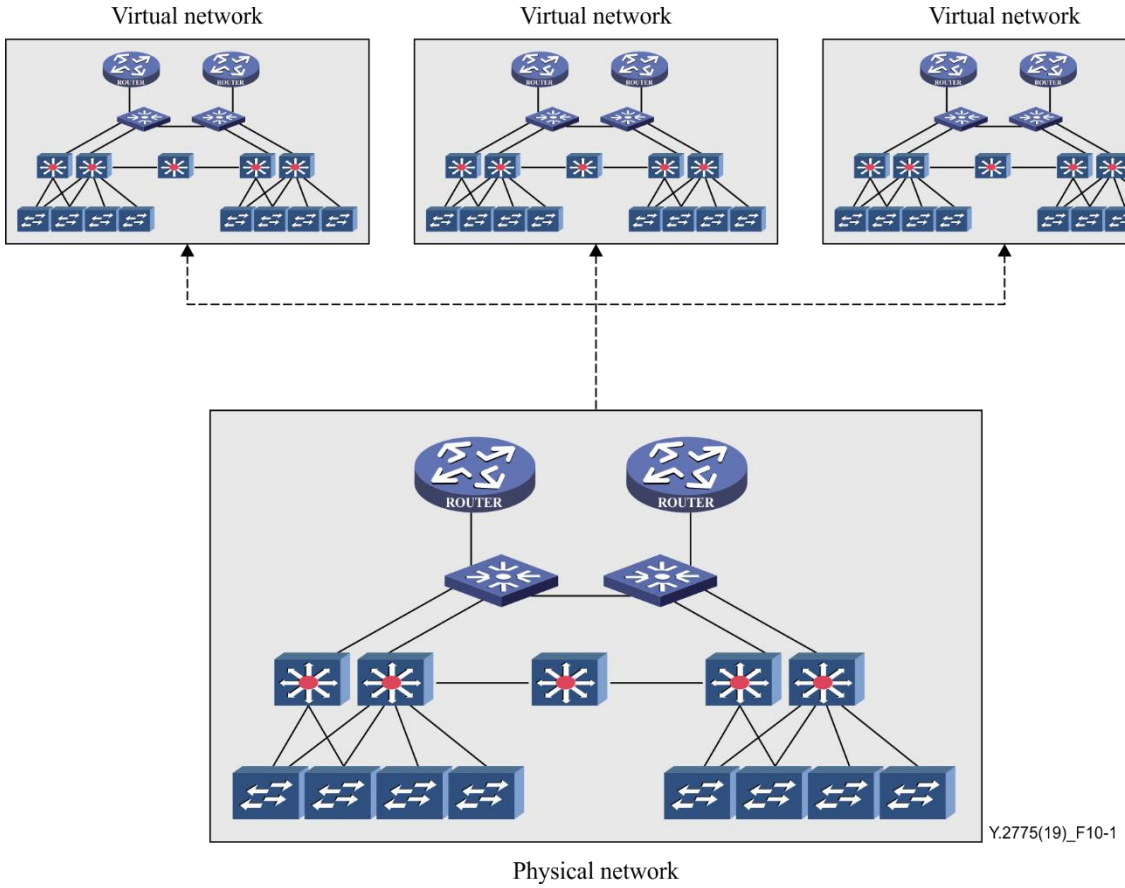


Figure 10-1 – An example of network virtualization

10.2 DPI functional architecture for network virtualization

10.2.1 Overview of DPI functional architecture for network virtualization

The general and core feature of a DPI functional architecture for network virtualization is virtualization from physical components to virtual components. If the physical components have been virtualized to the virtual components, logically and functionally, the virtual components are equivalent to the physical components. Thus, mapping physical components to virtual components is the core of DPI functional architecture for network virtualization.

10.2.2 DPI physical components

A list of DPI-related physical components is listed as follows:

- DPI nodes;
- DPI engine;
- DPI policy information base (DPI-PIB);
- DPI-ScF;
- DPI-AcF;

- DPI-AcEF.

10.2.3 DPI virtual components

Not all DPI physical components can be virtualized, because some of the physical components are not physically independent.

However, the DPI node, DPI engine and DPI-PIB can be virtualized as a v-DPI node, v-DPI engine and v-DPI-PIB, respectively.

10.2.4 Mapping physical components to virtual components

There are several methods for mapping physical components to virtual components: 1:n, m:1 and m:n (where m and n are positive integers, and m > 1 and n > 1).

Figure 10-2 describes the 1:n mode.

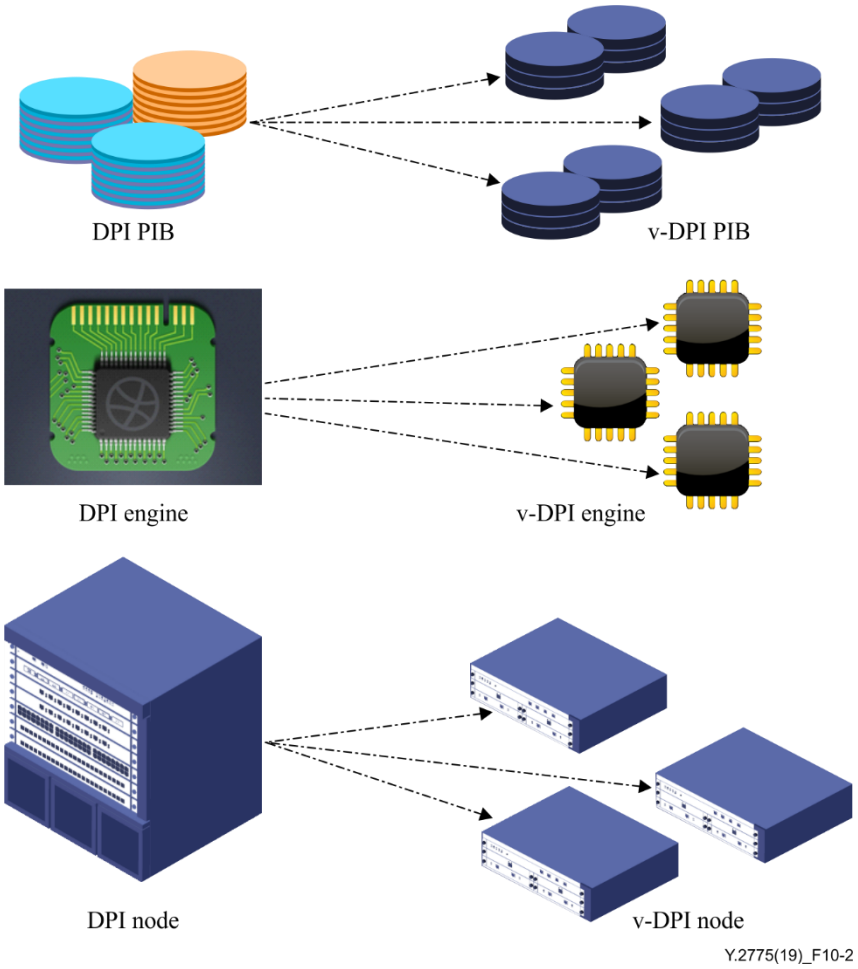


Figure 10-2 – 1:n mapping between physical component and virtual components

Figure 10-3 describes the m:1 mode.

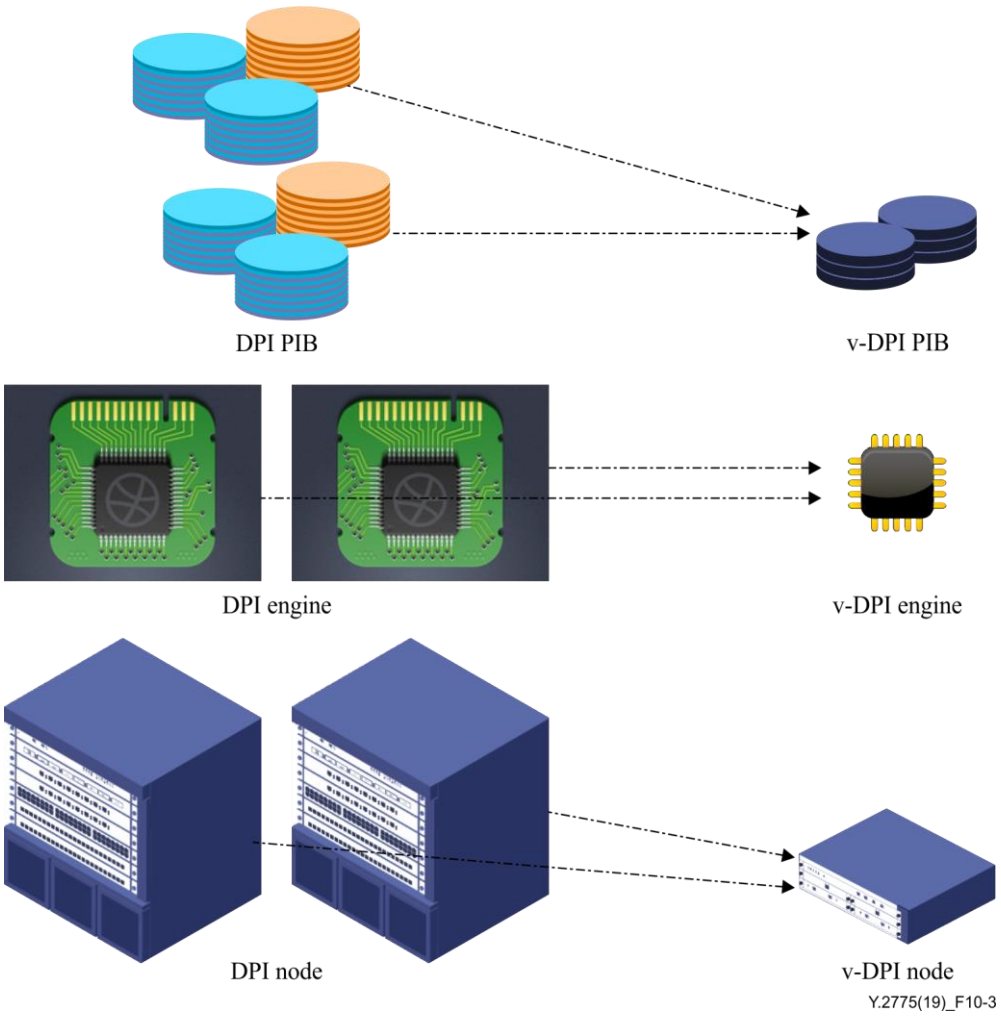


Figure 10-3 – m:1 mapping between physical components and virtual component

Figure 10-4 describes the m:n mode.

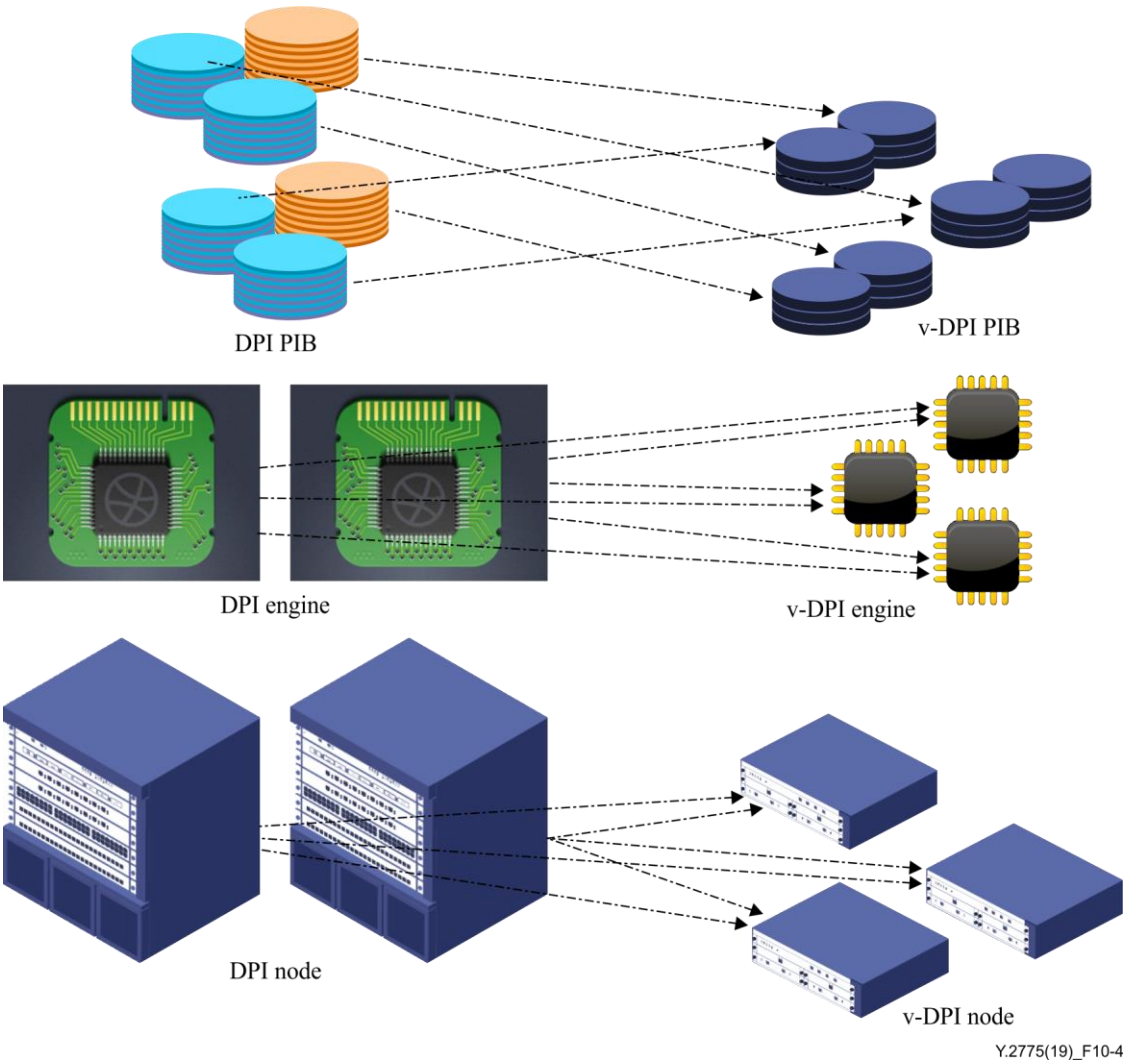


Figure 10-4 m:n mapping between physical components and virtual components

11 DPI functional architecture for evolving mobile networks

11.1 Introduction of evolving mobile networks

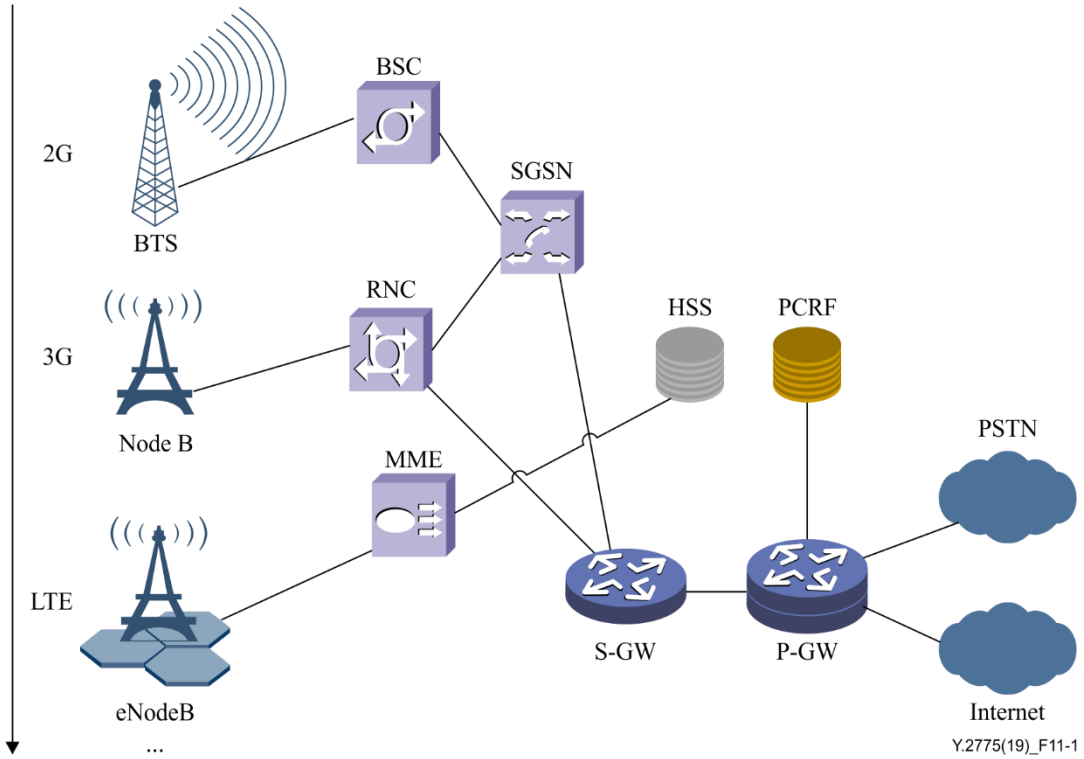
Evolution of mobile networks refers to the process where mobile networks are updated to a new generation. For example, when the second generation (2G) mobile network is updated to the third generation (3G) mobile network, that can be called a kind of evolution.

Therefore, an evolving mobile network is one where the mobile network is developing to a new-generation mobile network with the development of related technologies. For example, the current mobile network has been evolving to a fourth generation (4G) mobile network and will evolve to a fifth generation (5G), and so on, in the future.

11.2 General information for evolving mobile networks

The development of mobile networks is very rapid, and at present, mobile networks have evolved from 2G to 3G, and then from 3G to 4G and so on. To be certain, mobile networks will continually be evolving. Figure 11-1 depicts the evolving process and the main differences among mobile networks of 2G, 3G and long-term evolution (LTE).

It is certain that huge changes related to mobile networks have taken place during this evolving process. Then related technologies applied in mobile networks should also change with the change of the network.



BSC – base station controller
BTS – base transceiver station
HSS – home subscriber server
MME – mobility management entity
PCRF – policy and charging rules function
P-GW – packet data network gateway
PSTN – public switched telephone network
RNC – radio network controller
SGSN – serving GPRS support node
S-GW – serving gateway

Figure 11-1 – An example model of evolving mobile network

11.3 DPI applied in evolving mobile network

Deep packet inspection technology is a widely used network technology, and it is also widely applied in mobile networks. Figure 11-2 represents a current mobile network deployed with DPI functions (it can be realized by a DPI component or by a single DPI node).

It can be shown from Figure 11-2 that DPI functions can be deployed in four modes:

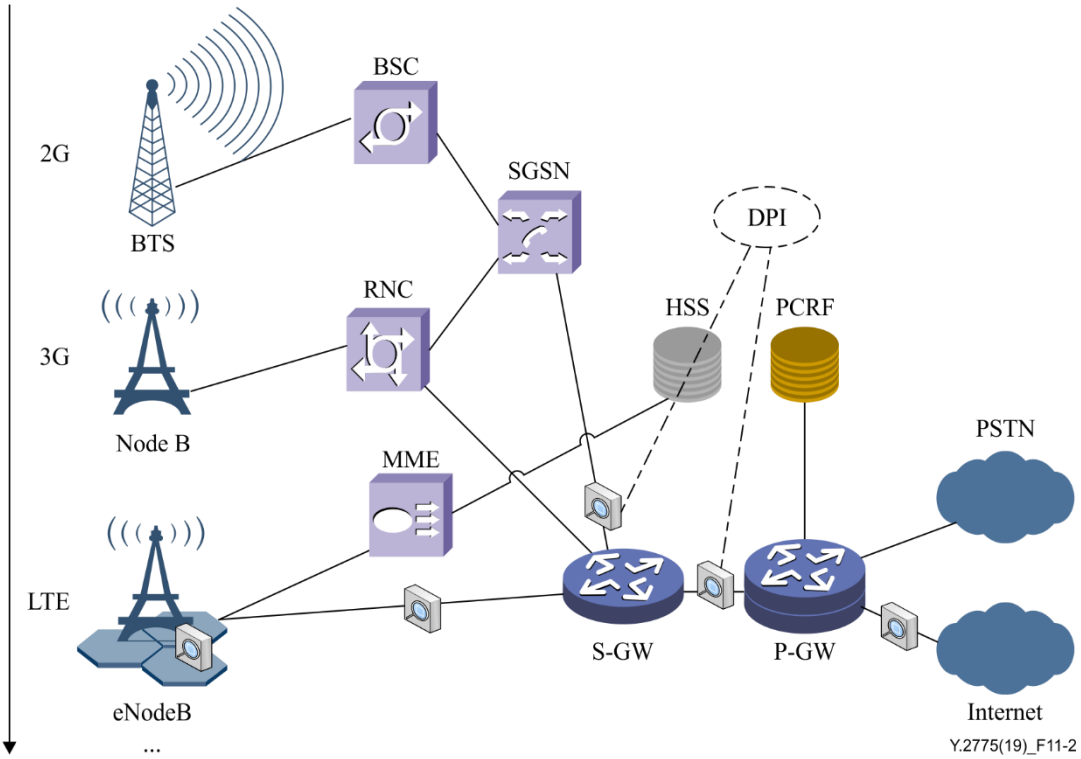
1) within radio access network (RAN);

2) at the access edge of packet-based core network;

3) within packet-based core network;

4) at the uplink edge of packet-based core network.

It should be noted that the above four deployment modes are not mutually exclusive.



BSC – base station controller
BTS – base transceiver station
HSS – home subscriber server
MME – mobility management entity
PCRF – policy and charging rules function
P-GW – packet data network gateway
PSTN – public switched telephone network
RNC – radio network controller
SGSN – serving GPRS support node
S-GW – serving gateway

Figure 11-2 – An example deployment of evolving mobile network supported DPI functions

11.4 Specification of DPI functions in evolving mobile networks

11.4.1 Implementing the DPI functions in the single logical level

In Figure11-2, the mobile network can be divided into four logical levels: RAN, access edge of packet-based core network, within packet-based core network and the uplink edge of packet-based core network. Implementing DPI functions in a single logical level means that DPI functions shall be carried out only in that one single logical level. Thus, DPI functions can be deployed according to the following mode:

- within the RAN;
- at the access edge of packet-based core network;
- within the packet-based core network;
- at the uplink edge of packet-based core network.

11.4.2 Implementing DPI functions in the hierarchical mode

Implementing DPI functions in a hierarchical mode means that DPI functions can be deployed in two or more logical levels but with collaboration between DPI functions deployed in the different logical levels.

For example, if DPI function, R, is deployed in the logical level RAN and DPI function, S, is deployed in the logical level access edge of packet-based core network, then the results of DPI function R should be used by DPI function S.

11.4.3 Implementing DPI functions in the distribution mode

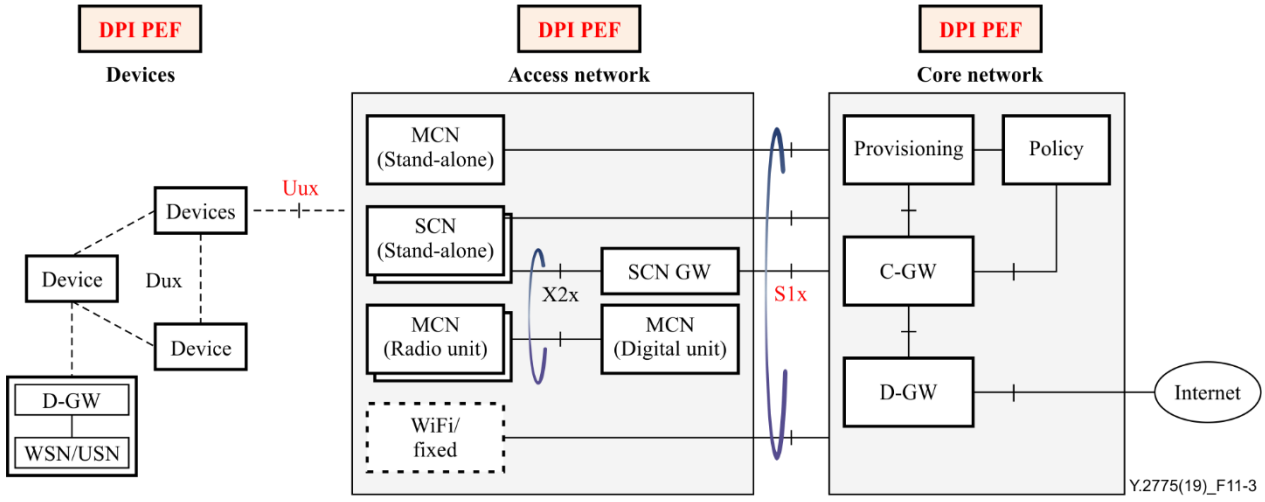
Implementing DPI functions in a distributed mode means that it is possible that some DPI components are deployed in the logical level, A, while some other DPI components are deployed in the logical level, B. All of the DPI components distributed in different logical levels cooperate to realize the functions of a DPI node.

For example, it is possible the DPI-Scf is deployed in the RAN, the DPI-Anf is deployed within the packet-based core network and the DPI-AceF is deployed at the uplink edge of packet-based core network.

11.5 Example DPI functional architecture for IMT-2020

At present, the mobile network is evolving to International Mobile Telecommunication (IMT)-2020. Figure 11-3 is an example of the reference model in the G Forum 2016 5G white paper [b-5Gforum-WP]. It is intended to provide a common understanding on what a reference model looks like.

Figure 11-3 also depicts an example DPI functional architecture for IMT-2020. As illustrated, the DPI in IMT-2020 is deployed in stand-alone and distributed models, the DPI-PDF is located in the core network, while the DPI PEF is located in the devices and access networks.



MCN – management communication network
SCN – signalling communication networks
USN – ubiquitous sensor network
WSN – wireless sensor network

Figure 11-3 – An example DPI functional architecture for IMT-2020

12 Security and other considerations

According to [ITU-T Y.2770], the DPI-FE and the information pertaining to DPI operations should be under protection against threats. It shall be guaranteed that information exchanged between remote policy decision function (R-PDF) and L-PDF is accomplished safely.

The mechanisms specified in [ITU-T Y.2704] address security requirements of this Recommendation.

Regulations, privacy and security application aspects of DPI are outside the scope of this Recommendation. Vendors, operators and service providers are required to take into account national regulatory and policy requirements when implementing this Recommendation.

Bibliography

[b-ITU-T Y.3001] Recommendation ITU-T Y.3001 (2011), *Future networks: Objectives and design goals*.

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

[b-IETF RFC 7665] IETF RFC 7665 (2015), *Service Function Chaining (SFC) Architecture*.

[b-5Gforum-WP] 5G Technical Study – 2015~2016 5G Technical White paper. <https://committee.tta.or.kr/include/Download.jsp?filename=choan%2F%282017-842%29+5G+%B1%E2%BC%FA+%BF%AC%B1%B8+-+2015%7E2016+5G+%B1%E2%BC%FA%B9%E9%BC%AD%28%B1%E2%BC%FA%BA%B8%B0%ED%BC%AD%29_%C0%C7%B0%DF%BC%F6%B7%C5_%5B3%5D.pdf>

[b-NFV-Framework] ETSI GS NFV 002 V1.2.1 (2014), *Network function virtualization (NFV) architectural framework*.

SERIES OF ITU-T RECOMMENDATIONS

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

Printed in Switzerland
Geneva, 2019