IOWN GLOBAL FORUM™

Functional Architecture for Deterministic Networking with Software-Defined Optical Interconnect

Classification: APPROVED Reference Document
Confidentiality: PUBLIC
Version 1.0
[E2E DN FA]
July 24, 2025

Legal

Contents

List of Figures

List of Tables

1. Introduction

1.1. Purpose

1.1.1. Background

In recent years, the demand for data centers has been rising more than ever for processing vast amounts of data. Some examples are large data transfer for AI/cloud applications, data replication for disaster recovery, media streaming, and data transport for cloud-native mobile systems (e.g., vRAN). However, more regions are imposing restrictions on the construction of data centers with the aim of curbing carbon dioxide emissions associated with power consumption. This makes it increasingly challenging to respond rapidly to the ever-growing demand.

This problem can be solved, for example, by building data centers in areas with abundant supplies of renewable energy and combining them with existing data centers to be treated as a single big-scaled data center. In order to utilize computing resources spread across multiple data centers as if they were within a single data center, it is necessary to ensure deterministic Quality of Service (QoS) between computing resources, i.e., Network Interface Cards (NICs) of computing nodes, as opposed to between data centers. On the other hand, while Open APN can provide connections with deterministic QoS, making Open APN connections for all NIC-to-NIC pairs would not be a cost effective nor scalable solution. Therefore, this document defines a functional architecture of networks that provide deterministic-QoS connections for NIC-to-NIC communication, leveraging site-to-site optical transport paths, primarily by Open APN as a foundation technology.

1.1.2. Purpose

The purpose of this functional architecture document is to grow the business of various stakeholders, including enterprise users, cloud service providers, data center operators, and Open APN providers, by enabling unprecedented distributed computing applications that rely on deterministic-QoS connections, e.g., RDMA and PTP.

(1) Enterprise user:

They can utilize remote data centers as extensions of their on-premise environments. By sharing the desired resources among multiple users, they can access remote advanced computing resources more affordably in an 'as a Service' model. As a result, it becomes possible to offer more advanced services to various enterprise users.

(2) Cloud service provider:

The increased options for data center expansion allow for a more rapid and efficient response to customer needs.

(3) Data Center operator:

As the demand for data centers from the above-mentioned enterprise users and cloud service providers increases, it leads to the expansion of business opportunities directed toward enterprise users. Furthermore, it becomes possible to have urban-like value to suburban sites that are rich in renewable energy and relatively inexpensive. This can result in both contributions to environmental issues and an expectation of increased profit margins.

(4) Open APN provider:

By providing Deterministic Network service to data center operators, Open APN providers such as telecommunications carriers can gain new business opportunities.

Another challenge in inter-data center connections is the coordination between Open APNs when traversing multiple service providers. This challenge is addressed in the 'Multi-domain IOWN Networking' document. Depending on specific cases, this architecture can be combined with 'Multi-domain IOWN Networking' to provide Deterministic Networking service across data centers.

1.2. Objective

The objective is to define Deterministic Networking service and define a functional architecture of its infrastructure to enable multi-operator/multi-vendor infrastructure operation.

The Deterministic Network with Software-Defined Optical Interconnect (DN) is a network that provides NIC-to-NIC deterministic connections (connections with deterministic performance) as a Service across optical transport paths, primarily by Open APN, that meet specified requirements with flexibility. It provides Ethernet frame-based logical connection with deterministic QoS (such as bandwidth and delay/jitter guaranteed) between NICs. Today's packet network is typically operated as class-based and difficult to achieve deterministic connections. The DN provides deterministic connections across NIC, network within a site (e.g., data center network) and Wide Area Network (WAN), leveraging Open APN.

Moreover, the DN exposes a service API whereby the users can set up a connection, specifying parameters and capabilities for their requirements. Furthermore, it provides OAM functionalities (such as real-time performance monitoring and fault management) for NIC-to-NIC deterministic connections being established. With this API, various use cases can be realized, which require computing resources across multiple data centers.

Note that Open APN provides high-speed, ultra-reliable, and low-latency connections. However, in typical deployment scenarios of Open APN, Open APN is deployed in WAN connecting multiple sites. The DN intends to extend features of Open APN between NICs - across NIC, network within a site (e.g., data center network) and WAN. Also note that there is no restriction to deploy the DN only in network within a site (e.g., data center network).

1.3. Scope of Work

1.3.1. What is in scope and out of scope

The scope work is as follows.

- Overall functional architecture, including functional components and interfaces.
- Service API semantics.
- Functional definition of data plane components and control and management plane components.

The scope does not include the following.

- Specifying service API syntax, including format for the endpoints of this network service.
- Specifying data plane protocols to be used (examples are described in Section 4.3). Note that recommended data plane protocols are for further study, including evaluation through PoC activities.
- Specifying implementations of data plane components and control and management plane components. (Annex E shows implementation examples.)

1.3.2. Relationship with Other IOWN Global Forum Activities

(1) Open APN

The Open All-Photonic Network (APN) is a network that connects endpoints directly with optical paths. It provides high-speed, ultra-reliable, and low-latency connections. The Open APN is one of the components that constructs the DN, and collaboration is necessary to establish the optimal path that connects the external networks.

(2) DCI

The Data-Centric Infrastructure (DCI) subsystem is intended to provide applications with a distributed and heterogeneous computing and networking environment that spans end-to-end, i.e., across clouds, edges, and customer premises. Similar to the Open APN, it is one of the components that constitute the network of the DN, and collaboration is necessary as DCI Cluster may provide DN endpoints.

(3) Multi-domain IOWN Networking

Multi-domain IOWN Networking aims to realize the interconnection of Open APNs operated by different organizations. Collaboration is necessary in cases where the NIC-to-NIC network traverses multiple Open APNs.

More details are described in Section 5.

Note that Function Dedicated Network (FDN) is defined as a logical network that provides connectivity between application data transport endpoints [1]. The DN is one type of FDN, providing NIC-to-NIC deterministic connections that meet specified requirements.

2. Use Cases, Deterministic Networking Service, and Non-Functional Requirements

This section describes Deterministic Networking service to meet network requirements based on use cases envisioned in the IOWN Global Forum. This section first shows a list of use cases analyzed, followed by Deterministic Networking service, and non-functional requirements to support these use cases.

2.1. Use Cases

It is important that a network providing NIC-to-NIC deterministic performance can support use cases that are envisioned in the IOWN Global Forum.

The following three use cases are analyzed, which are categorized as early adoption use cases in RIM-TF.

- Services Infrastructure for Financial Industry [2]
    - This use case describes multi-data center computing infrastructure with capabilities for financial institutions such as agile and resilient infrastructure. In particular, it describes:
        - Intra-region application deployment and workload migration to improve operational resiliency and agility.
        - Inter-region back-ups and workload migration to improve resiliency (in particular, Recovery Point Objective (RPO) zero).
- Remote Media Production for Broadcast Industry [3]
    - This use case describes cost-effective infrastructure, where broadcasters can optimize existing content development operations and skilled personnel for profitability while serving a larger addressable market. In particular, it describes
        - Media Production from Remote Site (the operator controls the remote media production resources, as if the operator and the media production resource were together at the same site).
        - Network Resource Sharing (the broadcast station requests network resource from the event venue to the broadcast station whenever an event occurs, without utilizing Outdoor Broadcasting (OB) van).
        - Media Production Resource Sharing (the broadcast stations request media production resources at the Media Production Center whenever required for live broadcasting, for cost effectiveness).
- Green Computing with Remote GPU Service for Generative AI / LLM [4]

  - This use case describes infrastructure where users generate LLM by performing training on GPU computing located on green data center using the training data existing on their own site.

Additionally, the following three use cases are analyzed, which are categorized as future looking use cases in RIM-TF.

- Area Management Security Reference Implementation Model (RIM) [5]
  - This RIM describes a reference that continuously analyzes the surveillance camera video image and LiDAR sensor data with AI to identify criminal activities or accidents for a prompt response and/or action.
- Interactive Live Music RIM [6]
  - This RIM describes a reference for one of the future-looking VR use cases that involves a live music concert in virtual space with massive number of users.
- Remote Controlled Robotics Inspection RIM [7]
  - This RIM describes a reference where a maintenance expert can remotely control on-site robots to carry out essential procedures. These procedures include thorough inspections, parts replacement, and valve closure as if the expert was physically present at the plant site.

Figure 2-1 summarizes these use cases. These use cases utilize Open APN-based WAN to interconnect multiple sites, such as user premises (enterprise data center, R&D center, building complex, event venue etc.), Multi-access Edge Computing (MEC) data center (for edge cloud) and cloud/co-location data center (for center cloud, large GPU cluster etc.).

For use cases that utilize GPU-based infrastructure (such as Green Computing with Remote GPU, CPS AM security, ILM, Remote Controlled Robotics Inspection), it is expected to utilize GPU located at MEC data center or cloud/co-location data center, while data is located at user premises and sent or accessed. This allows cost effective and greener use of GPU infrastructure.

For use cases that allow personnel to work remotely (such as Remote Media Production, Remote Controlled Robotics Inspection), it is important that resources at cloud can be accessed remotely, and that personnel can access devices as if they are at the same location.

For use cases that aim at scalability, agility and resiliency by the use of multi-data center infrastructure (such as Services Infrastructure for Financial Industry), it is important to be able to move data and/or application instantly.

Note that in Services Infrastructure for Financial Industry use case, data centers may be located on user premises (such as enterprise private data center), at co-location data center (such as leased-space at co-location data center), or at cloud data center [2].

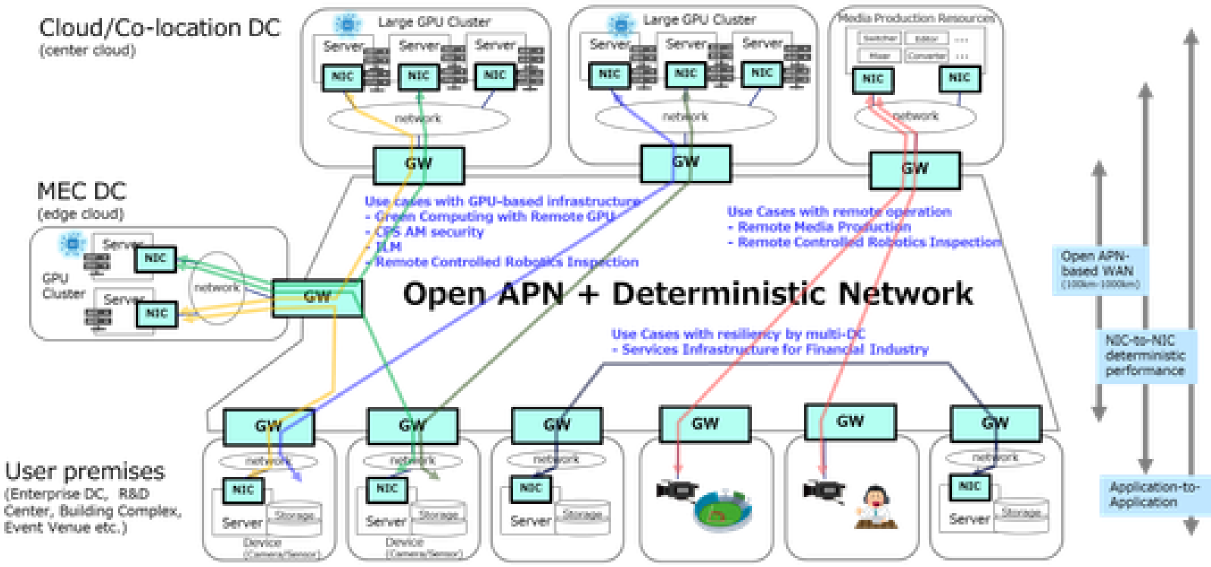


Figure 2-1: Use Case Summary

2.2. Deterministic Networking Service

In order to support use cases mentioned in Section 2.1, the DN should provide NIC-to-NIC deterministic connections as a Service. Deterministic Networking service is a service with the following features:

- Provides an Ethernet frame-based logical connection with deterministic QoS (such as bandwidth and delay/jitter guaranteed) between NICs.
- Exposes a service API whereby the users can set up a connection, specifying the following parameters, which we refer to a logical channel in this document.
    - Endpoints (NOTE: the address space and format are outside the scope of this format)
    - Performance parameters
        - Bandwidth
        - Delay/jitter constraints
        - Packet (or frame)-loss/packet (or frame)-reordering constraints
        - Resiliency/availability constraints
    - Additional capabilities, such as follows.
        - BC/TC [9]: for PTP transport, BC/TC features need to be supported on network devices where necessary.
        - ECN/PFC: for RDMA support, ECN/PFC features need to be supported on network devices where necessary.
- Provides OAM functionalities (such as real-time performance monitoring and fault management) for logical connections being established.

Users/applications can request connections between endpoints through a service API specifying performance parameters and additional capabilities. Network(s) should provide NIC-to-NIC logical connections to meet performance parameters and additional capabilities. Network(s) should provide OAM functionalities for NIC-to-NIC logical connections being established.

Details about service API are described in Section 4.5.

Users may request constant bandwidth for constant traffic, or may request time varying bandwidth for time varying traffic. For time varying traffic, the service API should support bandwidth update.

Delay/jitter constraints, packet-loss/packet-reordering constraints and resiliency/availability constraints are represented by quantitative values (e.g., delay bound). Alternatively, these constraints may be represented by classes (e.g., delay sensitive class or best effort class), when users do not require quantitative values to be satisfied. Even in such cases, the DN should be able to provide more fine-grained control of the network, compared to today's packet-based network.

Note that this document does not exclude cases when NIC control is not performed. Details are described in Section 4.4.

2.3. Non-Functional Requirements

The followings are non-functional requirements for Deterministic Networking service.

- Agility (ability to create connections/modify attributes of connections in timely manner)
  - The network should provide NIC-to-NIC deterministic connections immediately based on user/application requests.
  - It is worth noting that in many cases, there is no strict bound for such provisioning time.
  - This document does not specify definitive values for agility.
- Scalability (number of devices (e.g., servers) per site, and number of sites)
  - The network should provide NIC-to-NIC deterministic connections in certain scalability requirements. Scalability requirements differ per use cases and per deployment scenarios.
  - Typical numbers would be up to 1000+ endpoints (or servers) per site and up to 100+ sites, with 100-1000km distance from use cases.
  - Note that these numbers related to scalability are directly derived from use case requirements, and do not intend to be requirements for architecture itself. The architecture itself should be designed for even higher scalability.
  - This document does not specify definitive values for scalability.

2.4. Applicability to Use Cases

Figure 2-2 shows typical performance parameters requirements and non-functional requirements from use cases being analyzed. Annex A describes more detailed analysis, including early adoption use cases and future looking use cases.

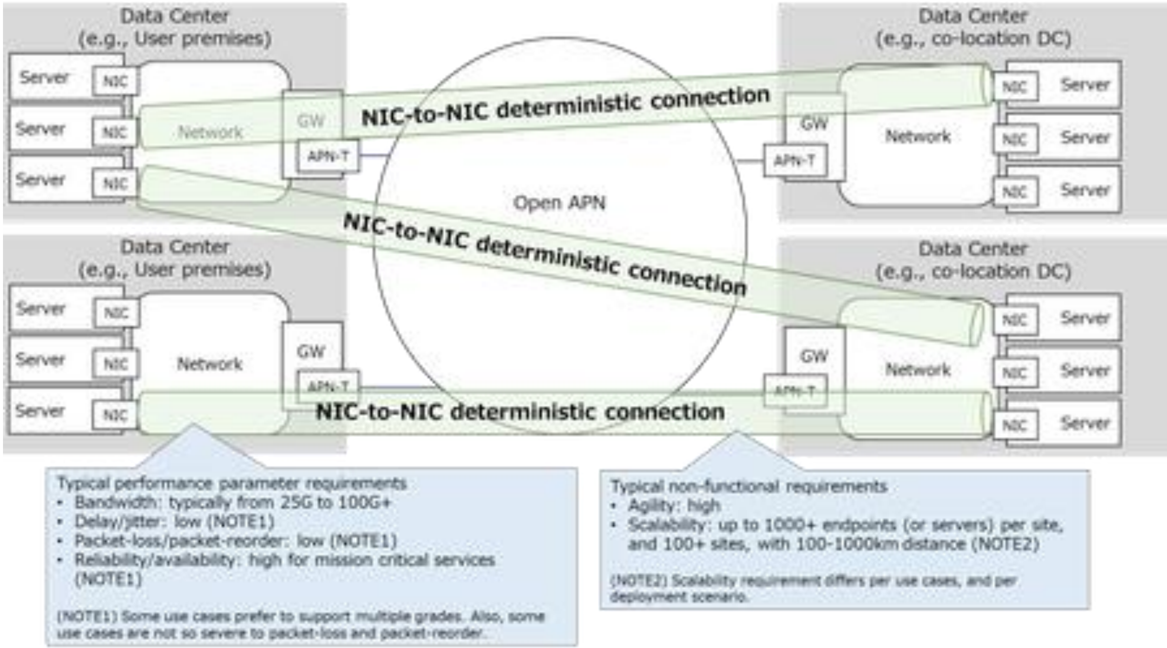


Figure 2-2: Requirements from Use Cases

Note that requirements described above apply to NIC-to-NIC. However, it is beneficial to consider site-to-site (or GW-to-GW) requirements (especially bandwidth) to design overall network architecture. Also, in some deployment scenarios, the network should provide site-to-site (or GW-to-GW) deterministic connections based on user/application requests.

3. Gap Analysis

Today's packet network is difficult to achieve deterministic networking. Today's packet network provides various QoS mechanisms (such as policing, shaping), but is typically operated as class-based (e.g., EF/AF/BE). Network performance provided for an application depends on conditions of the network (e.g., how other applications are sending traffic to the network, how network capacity is designed). In order to ensure performance, tuning of various parameters is needed, and tremendous effort is needed for testing, monitoring and trouble shooting.

On the other hand, Open APN provides high-speed, ultra-reliable, and low-latency connections, allowing end-to-end communication with deterministic performance. However, in typical deployment scenarios of Open APN, Open APN is deployed in WAN connecting multiple sites. There is a network within each site (e.g., data center network). We need a mechanism to provide deterministic connections across NIC, network within a site (e.g., data center network) and WAN.

Note that there are existing works in packet-based networks to achieve deterministic networking, such as Time-Sensitive Network (TSN) and IETF DetNet. In Deterministic Networking with Software-Defined Optical Interconnect, the focus is to provide NIC-to-NIC deterministic connections leveraging optical transport network, primarily Open APN (minimizing packet-based technologies), but TSN and IETF DetNet may be used as component technologies to realize Deterministic Networking with Software-Defined Optical Interconnect.

The followings are specific aspects to be addressed, in order to realize Deterministic Networking with Software-Defined Optical Interconnect.

(1) End-to-end deterministic connection

We need a mechanism to ensure deterministic performance across NIC, network within a site (e.g., data center network) and WAN to meet user/application requirements, including:

- End-to-end design and control
    - End-to-end selection of route (ensuring bandwidth, delay/jitter and reliability constraints)
    - End-to-end selection of reliability options (such as 1+1, N:M protection, or restoration after failure)
    - End-to-end setup, delete, and update of connections (e.g., in-service bandwidth update)
- End-to-end management
    - End-to-end performance monitoring
    - End-to-end fault management

(2) Flexible usage of bandwidth

Many use cases require flexible size of bandwidth (e.g., 20G up to 100G+, depending on use cases and deployment scenarios). Typically, in order to provide transport of above 20-30km physical distance, coherent optics are used. Coherent optics today can typically provide 100Gbps+ (up to 800Gbps or more) bandwidth, which is too big for some applications. Also, even though traffic from one endpoint may be above 100Gbps+, it is desirable to be able to provide connections to multiple endpoints simultaneously (e.g., 50Gbps to endpoint A, 50Gbps to endpoint B).

This means we need a mechanism to allow flexible usage of bandwidth, including:

- Ability to support logical channels over optical transport path (including Open APN path)
    - Flexible bandwidth can be assigned to logical channels.
    - Logical channels allow deterministic data transfer.
    - NIC-to-NIC application data transfer is mapped to logical channels for deterministic performance.

Note that logical channels may be provided over a bundle of optical transport paths (e.g., 2 x 400Gbps paths are bundled as 800Gbps). There are various technologies to provide such bundle (e.g., FlexO, FlexE, Ethernet link aggregation), but abilities to provide deterministic performance may differ depending on technologies.

Logical channels need to be supported at NICs, in cases that (a) a NIC speed is higher than bandwidth requested, or (b) multiple NIC-to-NIC deterministic connections are used simultaneously.

- Ability to support logical channels at NICs

Thus, logical channels are essential components for end-to-end deterministic connection. Example technologies for logical channels are mentioned in Section 4.3.

(3) Agile usage through API

Many use cases require to use end-to-end deterministic connections with agility (ability to create connections/modify attributes of connections in timely manner).

This means we need a mechanism to allow agile usage of end-to-end deterministic connections for applications through a service API.

4. Functional Architecture

4.1. Key Concept: Hub and Spoke Architecture

To achieve the objective, we introduce the hub and spoke architecture, where hub and POP are connected by optical transport paths (spoke), as shown in Figure 4-1. Hub and spoke has been adopted in many other industries such as transportation and airlines. The hub and spoke architecture has several advantages over the traditional many-hop architecture. First, managing service quality is easier because source-to-destination paths traverse fewer hops. Second, the exchange/distribution facility benefits from economies of scale.

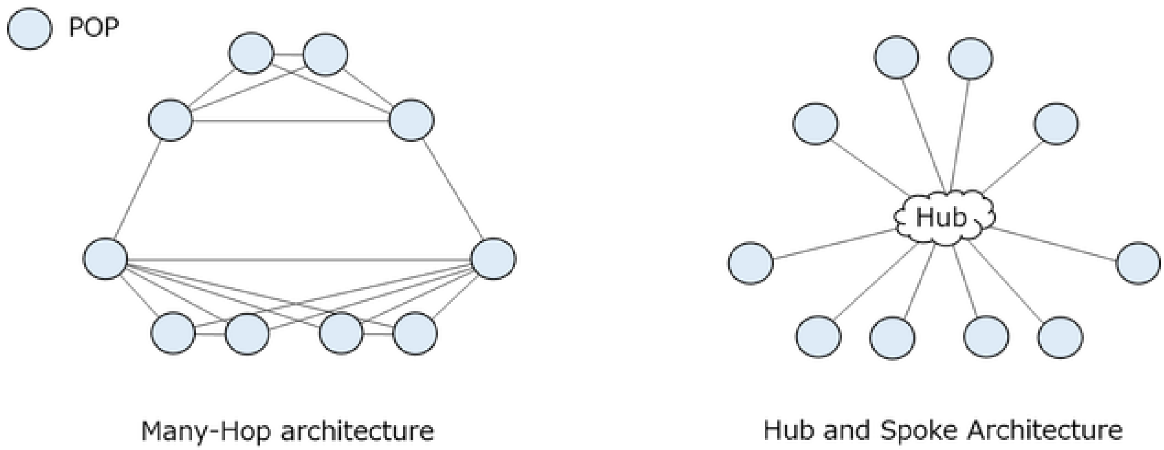


Figure 4-1: Hub and Spoke Architecture

Despite the advantages of the hub and spoke architecture, computer networks used to be built with many hops. There are several reasons. One reason is that the hub and spoke architecture requires longer-distance data transport, resulting in higher transport costs. Another reason is that it is less resilient because the hub becomes a single point of failure (or multiple hubs are necessary for redundancy). However, Open APN's features help us overcome these reasons. First, thanks to the recent evolution of optical transceivers, the distance cost of data transport has decreased dramatically. In addition, for use cases that connect many (hundreds or thousands of) user premises to a few (tens of) cloud/co-location data centers, which are shown in Figure 2-1, hub and spoke tends to be efficient. Second, Open APN's capability of dynamically creating optical transport paths enables us to deploy resilience network, including hub like a cloud (e.g., switching Open APN paths from a primary data center to a secondary data center in failures.)

The above considerations lead to a hub and spoke architecture where POPs are connected to a hub (could be implemented like a cloud) with software-defined optical transport.

Note that optical transport path connecting hub and POP is typically DWDM wavelength path by Open APN.WX or fiber path by Open APN.FX. However, there is no restriction to use statically provisioned wavelength or fiber path for POPs that do not require resilient connectivity.

Also note that hub may be implemented in multiple data centers for resiliency and elasticity like a cloud as mentioned above. In addition, POPs may be directly connected by optical transport paths, bypassing hub.

4.2. High Level Architecture

Figure 4-2 shows a high-level reference architecture of Deterministic Networking with Software-Defined Optical Interconnect.

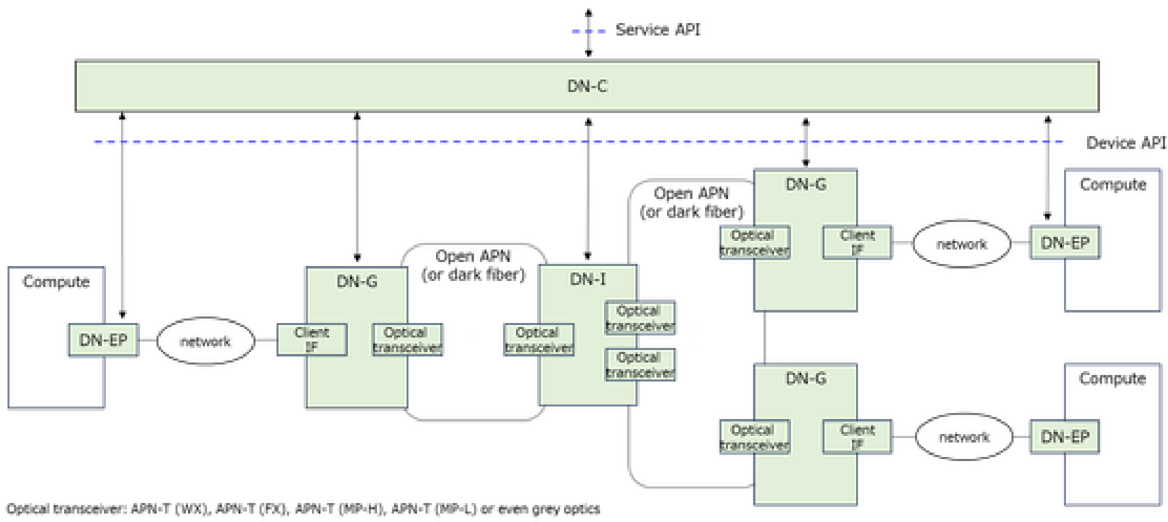


Figure 4-2: High Level Reference Architecture of Deterministic Networking with Software-Defined Optical Interconnect

The data plane components are Deterministic Networking Endpoint (DN-EP), Deterministic Networking Gateway (DN-G) and Deterministic Networking Interchange (DN-I). The control and management plane components are Deterministic Networking Controller (DN-C).

DN-EP is an endpoint and send/receive application data over the deterministic connection provided. Its functional requirements are (1) logical channel endpoint (enforcing route control and bandwidth control, performance management and fault management of logical channels), (2) mapping application flows to logical channels (by looking up header and mapping to logical channel, so that application data is transported over a specific logical channel), (3) communicating with DN-C for control and management, including logical channel endpoint and application flow mapping to logical channel.

Note that (2) mapping application flows to logical channels may be performed as part of service, or may be performed as user/application's role.

DN-G is a gateway between network within a site (e.g., data center network) and WAN (Open APN or even dark fiber). DN-G has optical transceivers (e.g., as pluggable transceivers), which may be APN-T (Open APN.WX), APN-T.FX (Open APN.FX), APN-T (MP-H), APN-T (MP-L) or even grey optics. DN-G functional requirements are (1) optical transport paths endpoints, which may be DWDM wavelength path by Open APN.WX, fiber path by Open APN.FX, or statically provisioned fiber or wavelength path, (2) mapping logical channels to optical transport paths, (3) switching/stitching logical channels, (4) communicating with DN-C for control and management, including logical channels. Stitching is optional and means mapping two different types of logical channels. Details are described in section 4.3. Also, switching/stitching patterns may be restricted due to hardware types. Details are described in Annex C.2.

DN-I is an interchange of logical channels, which terminates optical transport paths from/to DN-Gs. DN-I acts as switching logical channels (as a cross-connect), allowing multiple logical channels on a single optical transport path from a single DN-G can be distributed to multiple optical transport paths towards multiple DN-Gs. DN-I functional requirements are (1) optical transport paths endpoint, (2) mapping logical channels to optical transport paths (3) switching logical channels, (4) communicating with DN-C for control and management, including logical channels.

All functional components in this document are defined as logical instances as opposed to physical instances. In other words, we do not have to implement each functional component as a single physical node. In particular, DN-I should preferably be implemented with nodes in multiple data centers to achieve resiliency and elasticity like a cloud.

DN-C is a controller to receive service requests through a service API, and control and manage end-to-end deterministic connections across NIC (DN-EP), network within a site (e.g., data center network) and WAN (Open APN or even dark fiber) through a device API. Its functional requirements are (1) receiving service requests through a service API, (2) designing end-to-end logical channels, (3) controlling logical channels and controlling application flow mapping to logical channels, by configuring DN-EP and DN-G/DN-I through a device API, (4) managing end-to-end logical channels, by monitoring DN-EP and DN-G/DN-I through a device API. The functions of DN-C can be aggregated at one location or distributed at multiple locations. All functions of DN-C can be implemented together in a single box or separately in multiple boxes.

Note that two DN-Gs can be directly connected by an optical transport path, without passing through a DN-I. This case is applicable when bandwidth requirement between a pair of DN-Gs is sufficient to use capacity of an optical transport path (e.g., 400Gbps). Also, there could be multiple DN-Is, where DN-Is are connected by optical transport paths.

Also note that DN-C may control only DN-Gs/DN-Is (not DN-EPs). In such scenario, deterministic connections are provided between DN-Gs or interfaces of network within a site. Whether deterministic connections can be provided between DN-EPs may be based on design in such scenarios. Details are described in Section 4.4.

Annex C provides implementation examples of DN-EP, DN-G, DN-I and DN-C.

There could be multiple options how to realize the network within a site, i.e., network between DN-EP and DN-G, such as dark fibers, packet switches and optical switches.

- In case of packet switches, it is desirable that packet switches support logical channels and controlled and managed by DN-C for assuring end-to-end performance. Alternatively, the network composed of packet switches have enough capacity and there is virtually no packet loss.
- In case of optical switches, the network composed of optical switches may form Open APN (Open APN.WX or Open APN.FX), and controlled and managed by APN-C.

Figure 4-3 shows how Deterministic Networking with Software-Defined Optical Interconnect can support various use cases mentioned in Section 2. Figure 4-3 is a case where two DN-Gs are interconnected through DN-I. DN-I is deployed in two data centers for resiliency.

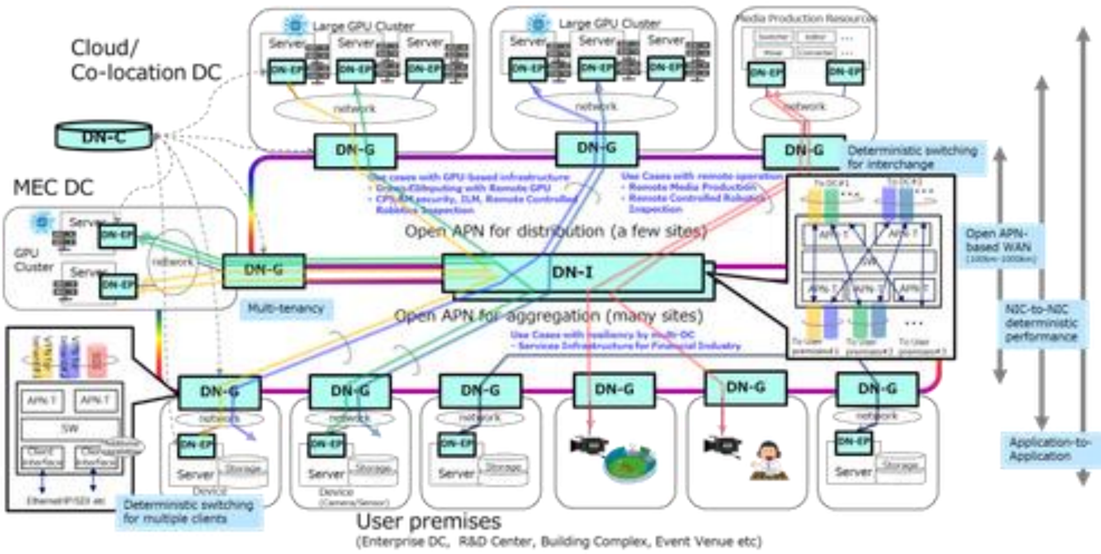


Figure 4-3: Use Case Realization by Deterministic Networking with Software-Defined Optical Interconnect

This network allows multi-cloud/co-location data center exchange for user premises. User premises have one (or a few) APN-T(s), and can connect to multiple clouds/co-location data centers simultaneously, with flexible bandwidth. DN-Gs at user premises aggregate multiple logical channels onto a single optical transport path. Logical channels on a single optical transport path are switched at DN-I towards multiple clouds/data centers. This allows flexible bandwidth usage for user premises, as well as for applications. Users can request/update/delete connections through a service API in agile manner.

4.3. Data Plane Model

This section describes data plane models, in particular how end-to-end deterministic connections are realized by logical channels.

Logical channels are assigned specific bandwidth, allowing deterministic data transfer. Example technologies for logical channels are VLAN, MPLS, SR (Segment Routing), FlexE, OTN. In Deterministic Networking with Software-Defined Optical Interconnect, logical channels are used in the following manner.

- Application flows are mapped to logical channels at DN-EP.
- Multiple logical channels are allowed per DN-EP.
- Logical channels are software-controlled, specified various parameters (such as bandwidth constraints, delay/jitter constraints, and reliability constraints) to satisfy.
- A DN-EP may be restricted to connect by logical channels with a specific group of DN-EPs (like VPN).
- A single type of logical channel (e.g., MPLS LSP or SR) may be used end-to-end, or multiple types of logical channel (e.g., VLAN/SR between DN-EP and DN-G, and MPLS LSP/FlexE/OTN between DN-Gs/DN-Is) may be used end-to-end.

Figure 4-4 shows a data plane model when a single type of logical channel is used end-to-end. Logical channels are switched at DN-Gs. This model typically requires WAN-oriented technologies (such as MPLS) in NIC (DN-EP). Some advanced NIC can support such technologies.

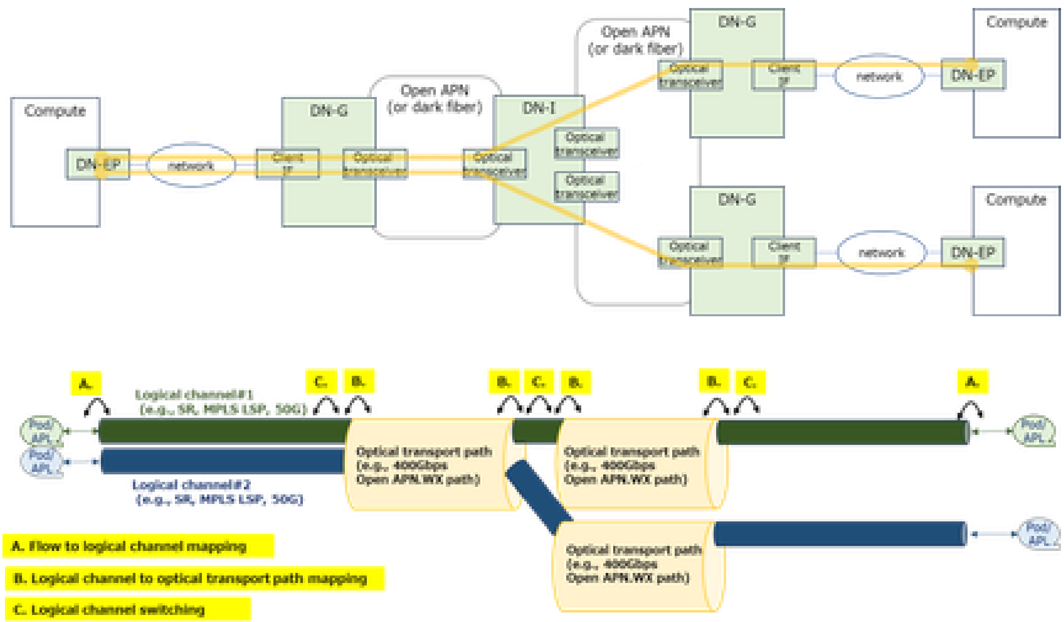


Figure 4-4: Data Plane Model with a Single Type of Logical Channel End-to-End

Figure 4-5 shows a data plane model when multiple types of logical channels are used end-to-end. Logical channels are stitched at DN-Gs where two different types of logical channels are connected. This model allows co-existence of LAN-oriented technologies (such as VLAN) and WAN-oriented technologies (such as MPLS, SR, OTN, FlexE). In particular, OTN and FlexE have benefit of hard traffic separation (allowing very strict delay and jitter control), but are rarely supported on NICs. Also, WAN is usually where bandwidth is scarce, thus strict control is important for assuring end-to-end quality. By allowing co-existence of technologies, it is expected that deterministic connections are provided by economical and technically feasible manner.

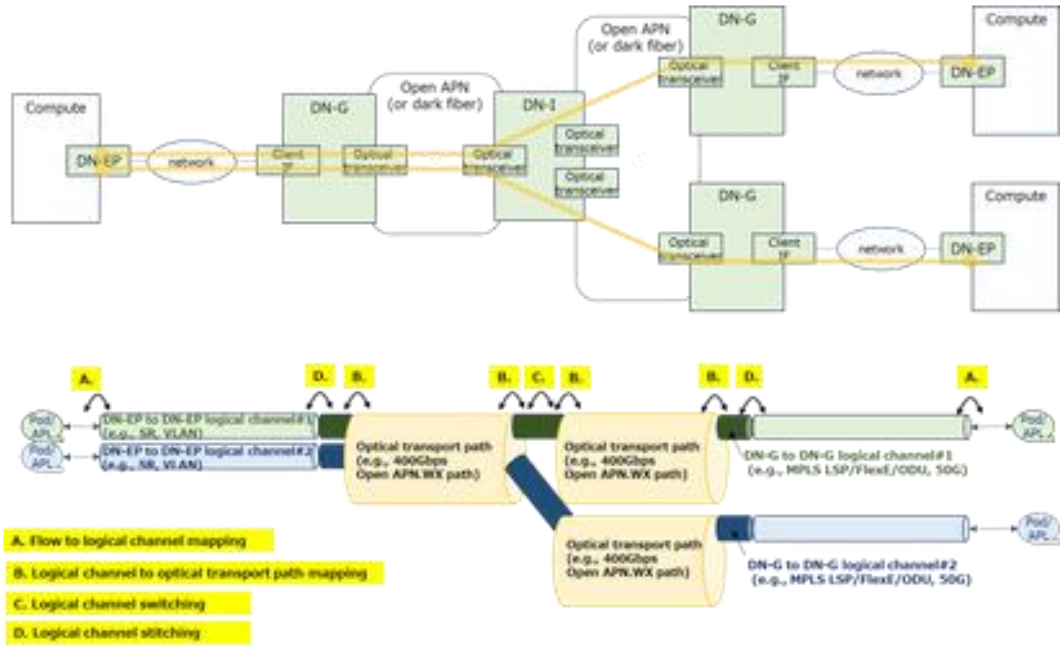


Figure 4-5: Data Plane Model with Multiple Types of Logical Channels End-to-End

Figure 4-6 shows a data plane model where two DN-Gs are directly connected by an optical transport path, without passing through a DN-I. Figure 4-6 described a switching example, but stitching can be equally applicable.

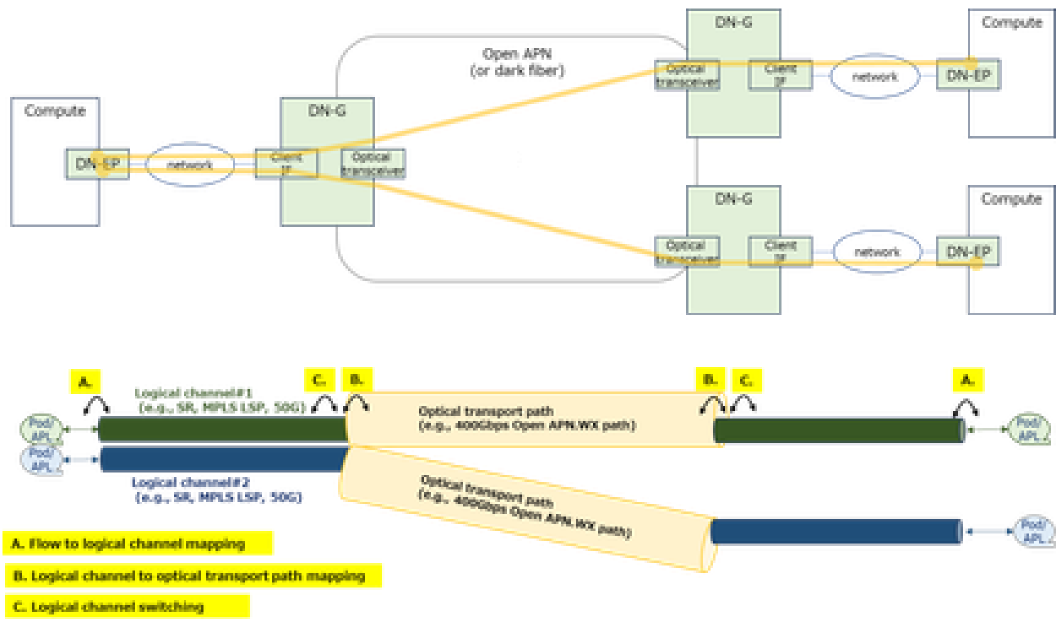


Figure 4-6: Data Plane Model without DN-I

Note that it is possible that a single application flow is mapped to a single logical channel (1:1), or multiple application flows are multiplexed/demultiplexed to multiple logical channels (N:M, M=1 typically as multiplexing/demultiplexing). Also, it is possible that logical channels are switched/stitched 1:1, or N:M (M=1 typically as multiplexing/demultiplexing). Such multiplexing/demultiplexing increases scalability, but care needs to be taken to ensure deterministic performance. In this document, we do not specify how to ensure deterministic performance when multiplexing/demultiplexing is used. Also, multiplexing/demultiplexing may be used to provide best-effort type of data transfer.

4.4. Control and Management Plane Model

As mentioned in Section 4.2, the functions of DN-C can be aggregated at one location or distributed at multiple locations. All functions of DN-C can be implemented together in a single box or separately in multiple boxes. This depends on service model (demarcation), as well as scalability.

In this version of document, the following three models are described, as shown in Figure 4-7.

(1) DN-EP to DN-EP control

(2) Network to Network control & DN-EP control

(3) DN-G to DN-G control & DN-EP + network within a site control

Furthermore, the following two models are described when DN-EP control is not performed.

(4) Network to Network control

(5) DN-G to DN-G control

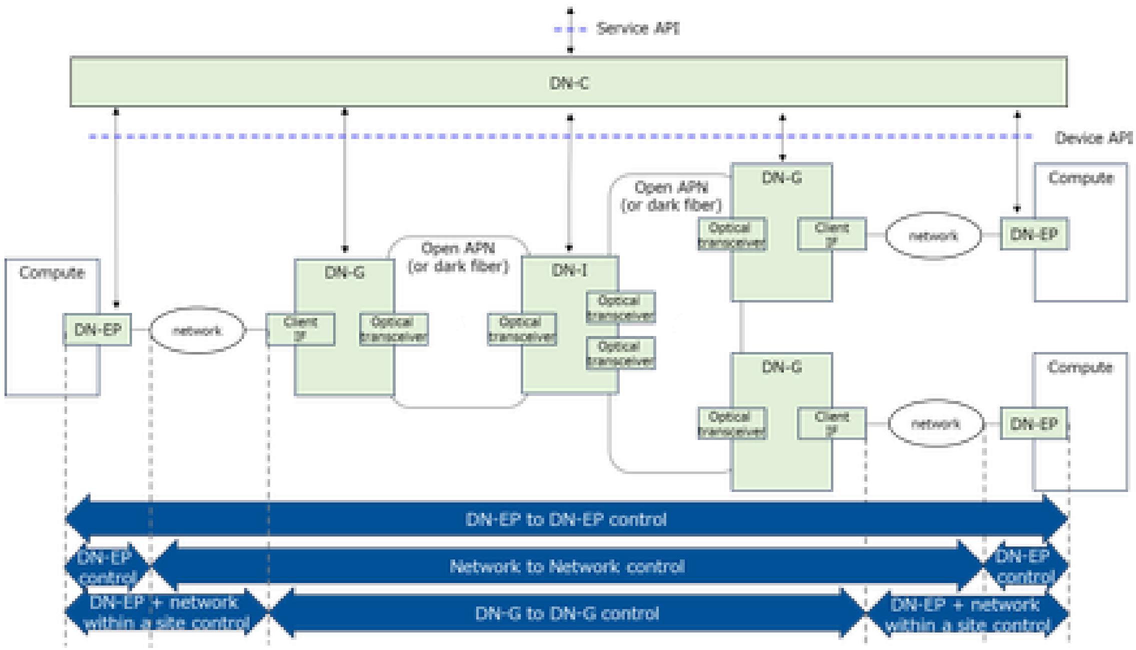


Figure 4-7: DN-C Control and Management Scope

(1) DN-EP to DN-EP control

This model is applicable when compute and network are tightly controlled and managed (e.g., by a single operator, or outsourced to an operator)

A logically single DN-C controls and manages DN-EPs as well as network. DN-EP is configured logical channels endpoints (traffic sending rate, logical channel identifier (such as VLAN)) as well as application flow mapping. DN-G (and network within a site, where necessary) is configured logical channels switching/stitching points. DN-EP configures traffic sending rate based on configuration. DN-G may further perform traffic control in order to improve deterministic performance (such as delay/jitter).

(2) Network to Network control & DN-EP control

This model is applicable when compute and network are loosely controlled and managed (e.g., separated by user and operator).

A logically single DN-C controls and manages network, while logically another DN-C controls and manages DN-EPs. DN-EP is configured logical channels endpoints (traffic sending rate, logical channel identifier (such as VLAN)) as well as application flow mapping. DN-G (and network within a site, where necessary) is configured logical channels switching/stitching points. DN-EP configures traffic sending rate. Network edges drop packets if traffic exceeds configured bandwidth.

Coordination between DN-EP control and network control is required, such as exchanging logical channel identifiers and ensuring end-to-end consistent requirements such as bandwidth, delay and reliability.

In this model, DN-C for DN-EP may be integrated into DCI Cluster Controller, if DCI Cluster is used for compute infrastructure. Details are described in Annex E.

(3) DN-G to DN-G control & DN-EP + network within a site control

This model is applicable when DN-EP + network within a site (e.g., data center network) and WAN are loosely controlled and managed (e.g., separated by data center operator and network operator).

A logically single DN-C controls and manages DN-Gs/DN-Is, while logically another DN-C controls DN-EPs (and network within a site, where necessary). DN-EP is configured logical channels endpoints (traffic sending rate, logical channel identifier (such as VLAN)) as well as application flow mapping. DN-G is configured logical channels switching/stitching points. DN-EP configures traffic sending rate. DN-Gs drop packets if traffic exceeds configured bandwidth.

Coordination between data center control and WAN control is required, such as exchanging logical channel identifiers and ensuring end-to-end consistent requirements such as bandwidth, delay and reliability.

In this model, DN-C for data center may be integrated into DCI Cluster Controller, if DCI Cluster is used for compute infrastructure. Details are described in Annex E.

(4) Network to Network control

This model is a shrinked version of (2), where DN-EP control from DN-C is not performed. One specific example is that DN-EP is embedded into appliances, and DN-EP control from DN-C is not possible. In such scenarios, deterministic performance may still be provided between DN-EPs. One example is that the network knows traffic sending rate from DN-EP (e.g., by interface speed) and can provide deterministic performance.

(5) DN-G to DN-G control

This model is a shrinked version of (3), where DN-EP control + network within a site (e.g., data center network) from DN-C is not performed. Similar to (4), in such scenarios, deterministic performance may still be provided between DN-EPs.

4.5. Service API

Service API consists of the following types.

- Setup DN-EP to DN-EP deterministic connection
- Delete DN-EP to DN-EP deterministic connection
- Update DN-EP to DN-EP deterministic connection
- Status check of already established DN-EP to DN-EP deterministic connection
- Notification of status of already established DN-EP to DN-EP deterministic connection

Setup/delete/update may be requested in the form of scheduling, where start-time and/or end-time of actual service are included as part of request.

For setup, the following parameters are specified.

(1) Endpoints

A pair of identifiers specifying endpoints.

An endpoint may be a DN-EP, or may be a network interface connected to DN-EP or a client interface of DN-G, depending on control and management plane model described in section 4.4.

Note that application flow mapping to logical channel may also be included as part of service API when application flow mapping to logical channel is provided as part of Deterministic Networking service. Alternatively, application flow mapping to logical channel may be performed as user/application’s role. Future version of this document may specify such details.

(2) Performance parameters

In this version of document, the following parameters are defined.

- Bandwidth
- Delay/jitter constraints (in terms of quantitative values or classes)
- Packet-loss/packet-reordering constraints (in terms of quantitative values or classes)
- Reliability/availability constraints (in terms of quantitative values or classes)
    - Examples of classes specifying mechanisms: protection (1+1, 1:1, N:M), restoration
    - Examples of classes specifying diversity level: physical route diversity, operator diversity (Note that SRG (Shared Risk Group) may be used as a specific technology to specify such diversity level)

(3) Additional capabilities, such as follows.

- BC/TC: for PTP transport, BC/TC features need to be supported on network devices where necessary.
- ECN/PFC: for RDMA support, ECN/PFC features need to be supported on network devices where necessary.

Optionally, information identifying user/application of this deterministic connection may be included.

Furthermore, some use cases require specific functions to be performed at endpoints (DN-EPs), such as RDMA and security features on deterministic connections. These specific functions may be requested as part of service API, or may be user/application's role. Future version of this document may specify such details.

Service API may be invoked (a) by user/application software or management/orchestration systems, or (b) by DN-EP, as shown in Figure 4-8. (a) may be used for provisioning type of usage, while (b) may be used for application triggered type of usage.

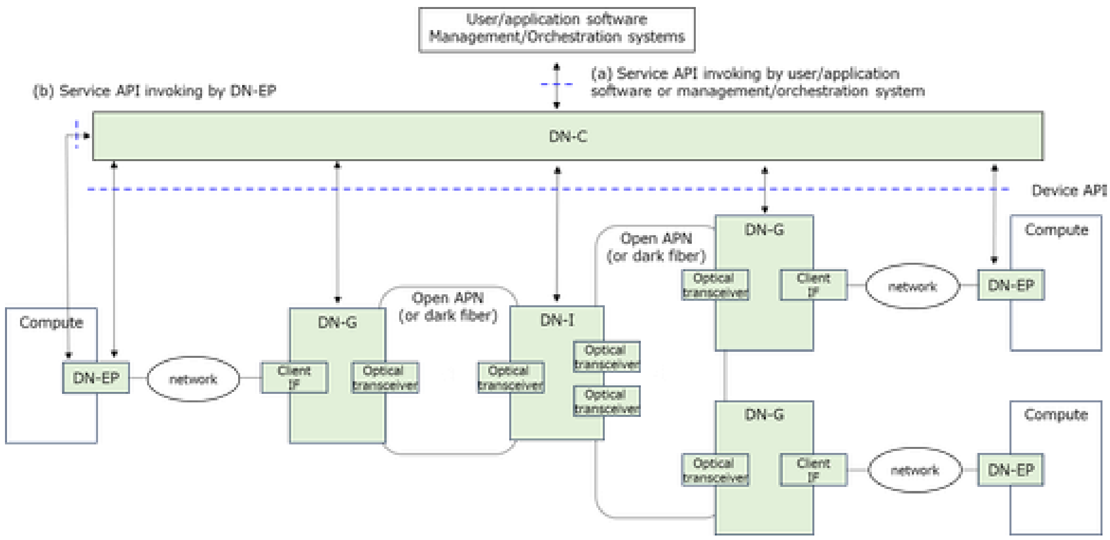


Figure 4-8: Service API Invoking

4.6. Example Procedures

This section describes example procedures for DN-EP to DN-EP deterministic connection setup. As a reference, a network consisting of enterprise user data center, WAN, cloud data center is used, as shown in Figure 4-9. An enterprise user buys cloud data center resources (compute and network), as well as WAN connections. The enterprise user performs remote GPU computing, using GPU computing resources on cloud data center and its own data on enterprise user data center.

The enterprise user owns its own data center, including compute and data center network. WAN operator owns Open APN-based WAN between DN-Gs. Cloud data center operator owns cloud data center, including compute and data center network. Control and management of enterprise user data center is outsourced to WAN operator, using DN-C and orchestrator operated by WAN operator.

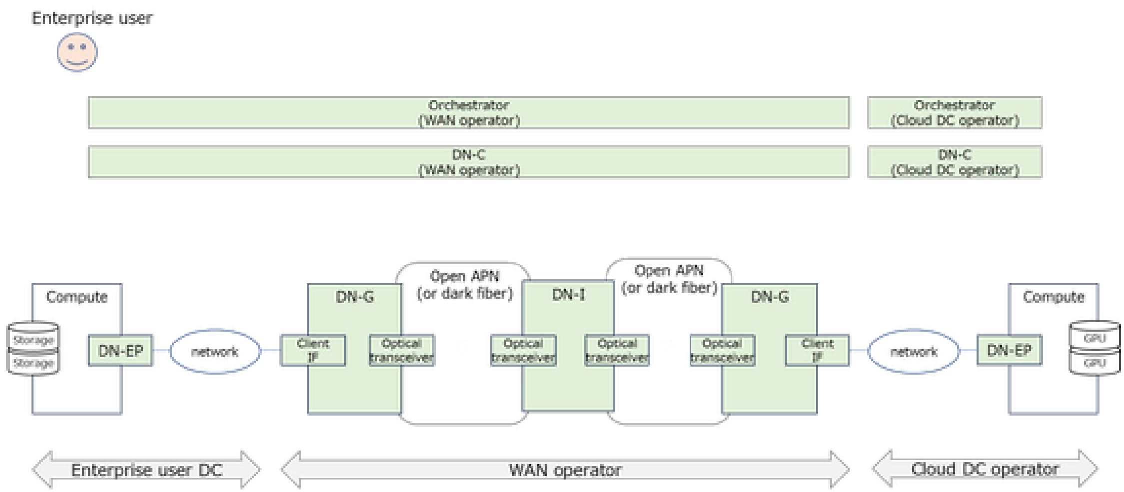


Figure 4-9: Reference Model for Example Procedures

Figure 4-10 shows an example procedure when a service API is invoked by enterprise user via orchestrator. WAN operator orchestrator and Cloud data center orchestrator coordinates for establishing a DN-EP to DN-EP deterministic connection.

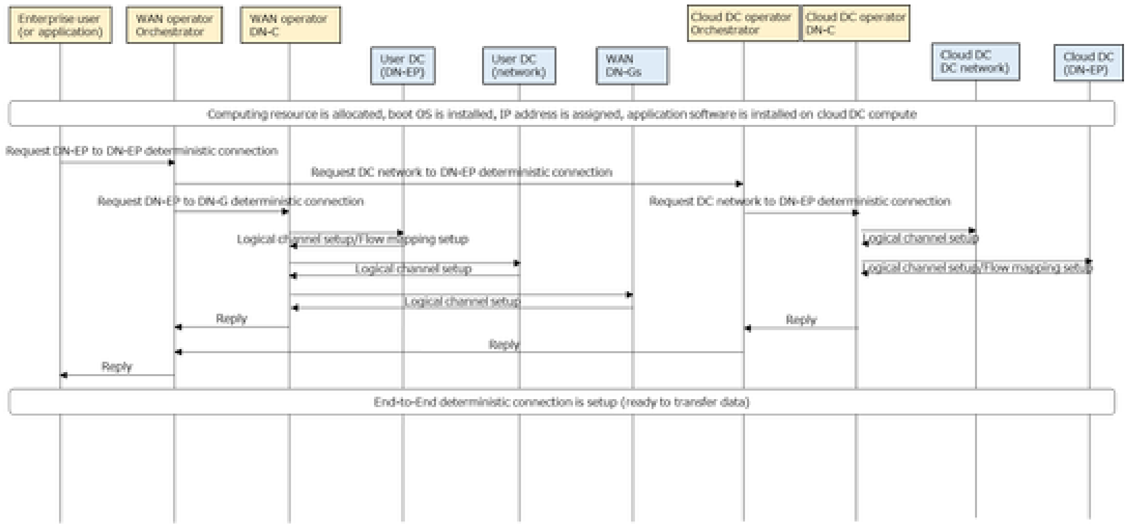


Figure 4-10: An Example Sequence (Service API invoking via Orchestrator)

Figure 4-11 shows another example procedure when a service API is invoked by DN-EP. In this example, first, a DN-G to DN-G logical channel is pre-ordered (as part of WAN connection service provided by WAN operator). Afterwards, a DN-EP to DN-EP deterministic connection is requested from DN-EP when data actually needs to be transferred.

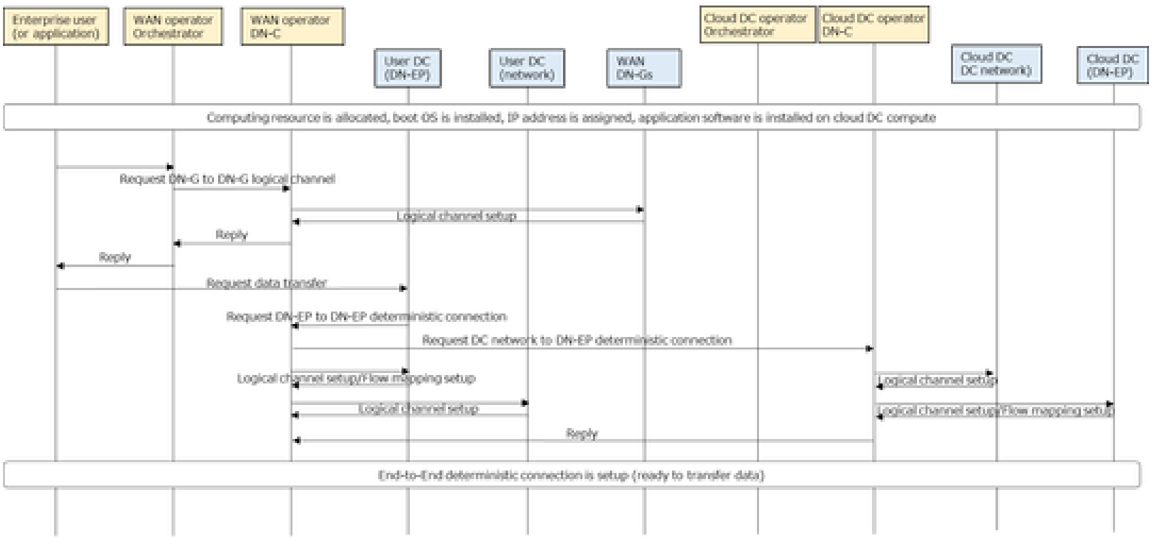


Figure 4-11: An Example Sequence (Service API invoking via DN-EP)

5. Relationship with Other Functional Architectures

5.1. Relationship with Data-Centric Infrastructure (DCI) Functional Architecture

Compute, NIC and data center network may be formed by DCI Cluster. In such scenarios, DN-EP to DN-EP deterministic connections span across DCI Cluster, WAN and DCI Cluster. Annex E describes potential relationship of DCI and Deterministic Networking with Software-Defined Optical Interconnect.

5.2. Relationship with Open APN

DN-C functions for controlling DN-Gs/DN-Is correspond to FlexBr Controller mentioned in Annex A.2 of Open APN Functional Architecture [8]. DN-C functions for controlling DN-Gs/DN-Is may be implemented as part of APN-C, or separately. Annex E.2 of Open APN Functional Architecture [8] describes possible models how APN-Ts are controlled and managed.

5.3. Relationship with Multi-Domain IOWN Networking

When an Open APN path composing DN-EP to DN-EP deterministic connections traverses multiple Open APNs operated by different providers, this Open APN path is provided by Multi-Domain IOWN Networking.

6. Conclusion

Deterministic Networking with Software-Defined Optical Interconnect provides NIC to NIC deterministic connections across NIC, network within a site (e.g., data center network) and WAN leveraging site-to-site optical transport paths, primarily by Open APN as a foundation technology, which are required for various use cases specified in RIM-TF. Deterministic Networking with Software-Defined Optical Interconnect allows flexible bandwidth usage, as well as agile service delivery via a service API. In this document, a high-level reference architecture of Deterministic Networking with Software-Defined Optical Interconnect, service API semantics, functional definitions of key data plane components and control and management plane components (DN-EP, DN-G, DN-I and DN-C) are described.

Deterministic Networking with Software-Defined Optical Interconnect has relationship with other IOWN Global Forum functional architecture components, in particular Data-Centric Infrastructure (DCI) and Open APN. Further work is required for detailing such relationship.

The IOWN Global Forum will work to solve the technical challenges to realize Deterministic Networking with Software-Defined Optical Interconnect, as well as to refine architecture for supporting various use cases. Furthermore, the IOWN Global Forum will develop necessary specifications in collaboration with relevant communities/SDOs.

Annexes

A. Detailed Requirements from Use Cases

Table A-1 shows a full list of network requirements from use cases, including early adoption use cases and future looking use cases.

Table A-1: Requirement Analysis for Use Cases

		BANDWIDTH	DELAY, JITTER	PACKET LOSS, PACKET REORDER	PROTOCOL	RELIABILITY, AVAILABILITY, SECURITY	AGILITY	SCALABILITY
Services Infrastructure for Financial Industry	Intra-regional Application Deployment and Migration	Normal traffic between DCs is less than 100Gbps, and increases up to 200Gbps during data replication etc. 25G between NICs (NOTE1)	Low delay, low jitter	Packet loss-less, packet reorder-less	IPv4/IPv6	High reliability/availability required	Yes	100-1000 servers per site 1000-10000 VMs per site a few sites 50-100km (intra region)
	Inter-regional Back-Ups and Migration	Normal traffic between DCs is less than 10-100Gbps, and increases up to 200Gbps during data replication etc. 25G between NICs (NOTE1)	Low delay, low jitter for Tier 1 systems (NOTE2)	Packet loss-less, packet reorder - less for Tier 1 systems (NOTE2)	IPv4/IPv6	High reliability/availability required (NOTE2)	Yes	100-1000 servers per site 1000-10000 VMs per site a few sites 500-1000km (inter-region)
Remote Media Production	Media production from remote site	1.2G (compressed) or 12G (uncompressed) between NICs and GWs	Low delay, low jitter (16.6ms)	Packet loss-less, packet reorder-less	IPv4/IPv6 (NOTE4)	High reliability/availability required (99.9999% network availability)	Yes	1 media essence stream 1000km radius
	Network resource sharing	20G between GWs (NOTE3)	Low delay, low jitter (16.6ms)	Not so severe compared with other cases	IPv4/IPv6 (NOTE4)	Not so severe compared with other cases	Yes	50 cameras per site 200km 6 sites (broadcast station) 10 sites (event venue)

		BANDWIDTH	DELAY, JITTER	PACKET LOSS, PACKET REORDER	PROTOCOL	RELIABILITY, AVAILABILITY, SECURITY	AGILITY	SCALABILITY
	Media production resource sharing	20G between GWs (NOTE3)	Low delay, low jitter (16.6ms)	Packet loss-less, packet reorder-less	IPv4/IPv6 (NOTE4)	High reliability/availability required (99.9999% network availability)	Yes	50 cameras per site 128 sites (broadcast station) 1000km radius
Green computing		100G+ between NICs	A few ms for 100km distance (RTT)	Not so severe (NOTE5)	IPv4/IPv6	Data security to be expected	Yes	100km+ distance
AM Security RIM	Monitored area to regional edge/center cloud	6Mbps-60Mbps between NICs (LiDAR/Camera) 67Gbps between GWs (per monitored area)	100ms-1s, including processing delay	Packet loss-less, packet reorder-less	IPv4/IPv6 RDMA			1000 cameras/LiDARs per site 100-1000 sites
ILM RIM	Audience to regional edge/center cloud	Upstream: 7.23Gbps between GWs Downstream: 12-69Gbps between GWs (depends on device types)	10ms (motion to photon) 70ms (motion to present) including processing delay (NOTE6)		IPv4/IPv6 (NOTE7)			Number of audience: 1.2million (ultimate goal) 1000km distance
	Artist to regional edge/center cloud	Upstream: 56-113Gbps between GWs Downstream: 6Gbps between GWs	10ms (motion to photon) 70ms (motion to present) including processing delay (NOTE6)		IPv4/IPv6 (NOTE7)			
Remote Controlled Robotic Inspection RIM	Plant site to regional edge/remote site	1Mbps per robot (control) 200kbps per robot (feedback) 2.5Gbps per robot (video, JPEG-XS, 5 cameras + 1 microphone)	100ms (control) 10ms (feedback) 100ms (video) including processing & mobile access delay	Packet loss-less, packet reorder-less	IPv4/IPv6 ATSSS, RDMA	99.999% (control) 99.999% (feedback) 99.999%/99.9% (video)		100 robots per site (working at the same time) 1000km

NOTE1: Bandwidth between GWs may depend on configurations, such as number of VMs to be migrated simultaneously.

NOTE2: It is preferable to be able to provide multiple service grades.

NOTE3: Bandwidth between NICs is around 1Gbps, assuming video compression.

NOTE4: PTP support may be required.

NOTE5: Initial benchmark assumes to use NFS over TCP.

NOTE6: This is the latency between Audience to Audience (via data centers), or the latency between Audience to Artist (via data centers).

NOTE7: There are several protocol options, such as RDMA (RoCE), RTP-based protocol (WebRTC, etc.), ST2110, or proprietary protocol (e.g., DisplayPort signal over ethernet).

B. Additional Data Plane Models

B.1. Coexistence of Various Optical Technologies

Section 3 describes logical channels as a way to enable flexible bandwidth usage of optical transport paths. Multiple logical channels can be multiplexed over an optical transport path. If we use coherent optics, bandwidth of an optical transport path is typically 100Gbps or more. A site (e.g., user premise) can connect to multiple sites (e.g., cloud or co-location data center) over an optical transport path by logical channels (e.g., 50Gbps to site A, 50Gbps to site B).

However, total traffic from/to one site may be much less than 100Gbps, in which case coherent optics are still too big. Deterministic Networking with Software-Defined Optical Interconnect allows cost and power consumption effective choice of optical technologies for each site, as shown in Figure B-1.

- High speed coherent is used between center site and hub site, where traffic is aggregated from/to multiple branch sites.
- Low speed coherent is used between branch site and hub site, where traffic from/to one branch site is around 100Gbps.
- IMDD is used between branch site and hub site, where traffic from/to one branch site is below 25Gbps and its physical distance is around 20km.
- PtMP (XR optics or TDM-PON) is used between branch site and hub site, where traffic from/to one branch site is much smaller than one wavelength. Annex F of the Open APN Functional Architecture [8] describes the detailed model of PtMP.

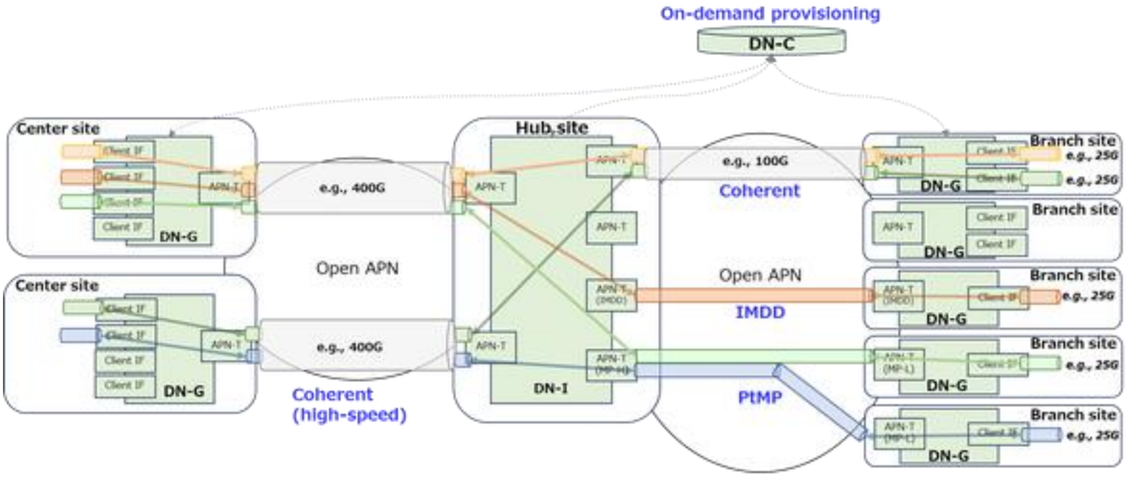


Figure B-1: Coexistence of Various Optical Technologies

B.2. Recovery Models

Mission critical use cases require high reliability/availability. The network needs to provide recovery mechanisms for failures, such as 1+1, N:M protection or recovery after failures. There are several options to provide recovery mechanisms in terms of recovery scope, as shown in Figure B-2.

- (a) Recovery at Open APN
  - A failure within Open APN can be recovered by Open APN recovery mechanisms (excluding a failure of APN-T). An example is OLP (Optical Line Protection). Recovery works per Open APN path. A secondary APN-T is not necessary.
- (b) Recovery at DN-G
  - A failure within Open APN can be recovered at DN-G recovery mechanisms (including a failure of APN-T). An example is OTN-based recovery, or L2/L3-based recovery (such as MPLS Fast ReRoute and Ethernet Linear Protection Switching). Recovery works per Open APN path, or per logical channel. A secondary APN-T is necessary.
- (c) Recovery at Network within a site
  - A failure within Open APN, as well as DN-G can be recovered at network within a site. The recovery point is switches, or may be computes for ensuring DN-EP to DN-EP recovery scope. An example is L2/L3-based recovery (such as MPLS Fast Re-Route and Ethernet Linear Protection Switching), or media specific mechanisms (such as ST2022-7). Recovery works per logical channel.

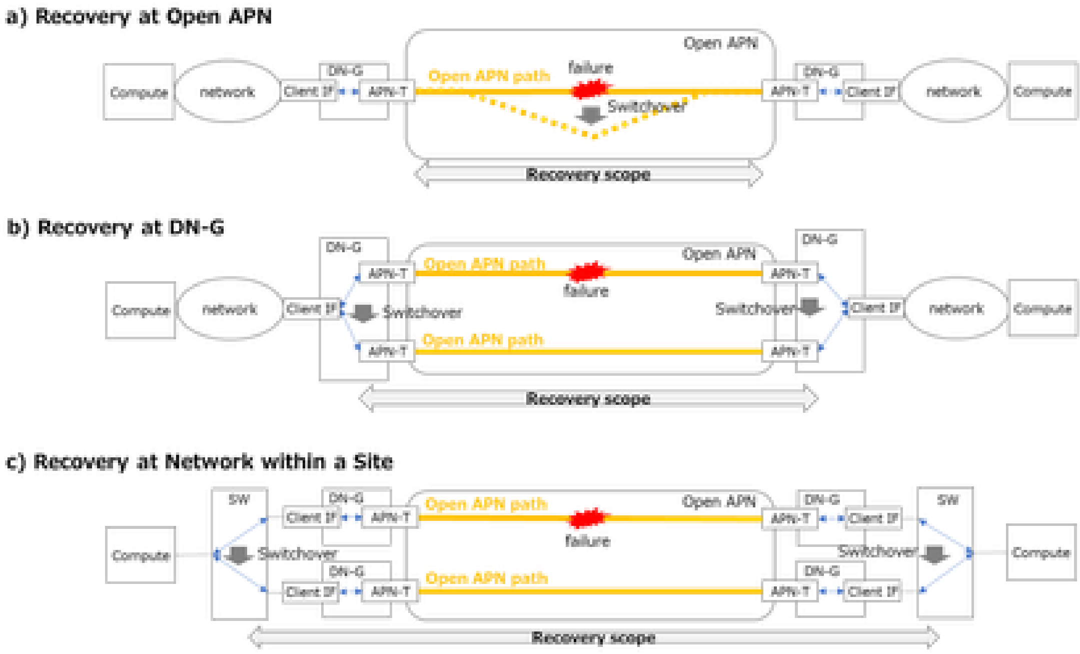


Figure B-2: Recovery Scope

Deterministic Networking with Software-Defined Optical Interconnect should provide mechanisms to allow appropriate recovery models to be used. One example is choosing the recovery model in network design phase to meet specific reliability/availability target. Another example is choosing the recovery model in provisioning phase by parameters specified in a service API.

C. Implementation Examples

C.1. DN-EP Implementation Example

Figure C-1 shows an DN-EP implementation example. In particular, this model assumes IPU/DPU type of DN-EP where it has programmable processors, allowing advanced packet processing features to be supported.

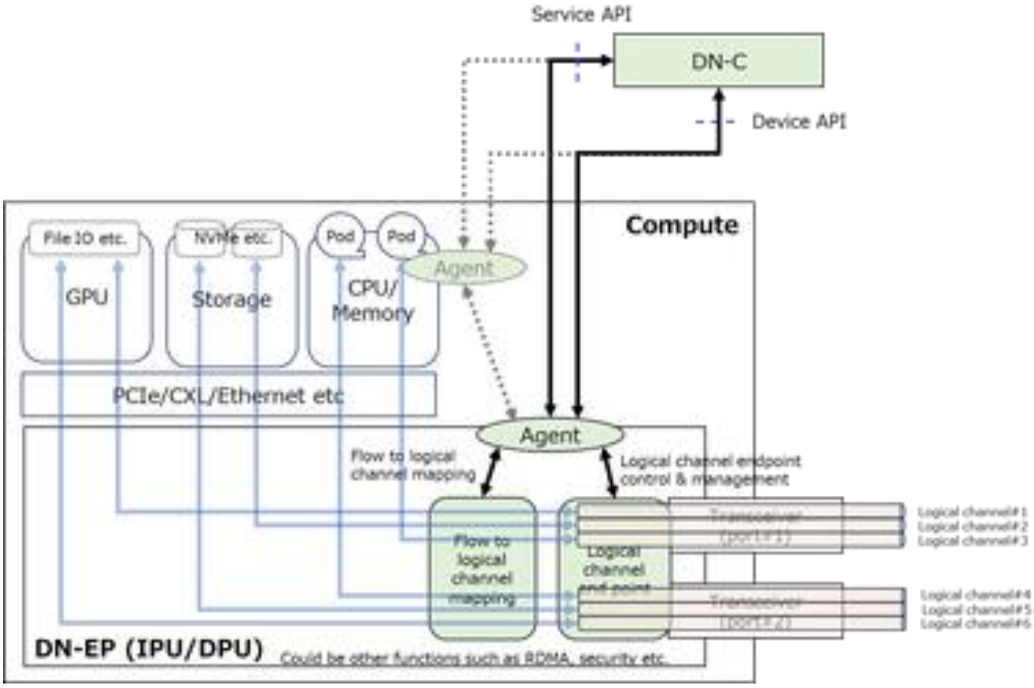


Figure C-1: DN-EP Implementation Example

DN-EP consists of the following functional blocks.

- Agent: a mechanism to communicate with DN-C via a device API, for controlling and managing logical channel endpoints and application flow mapping to logical channels. Additionally, an agent may communicate with DN-C for service request via a service API. An agent may reside on DN-EP, or may reside on host CPU. Communication channel between agent and DN-C could be in-band (the same port is used for logical channels as well as for communication with DN-C), or out-band (a distinct port is used for communication with DN-C).
- Logical channel endpoint: a mechanism to enforce route control, bandwidth control, performance management and fault management per logical channel. An example mechanism to enforce route control is mapping logical channel to physical port by checking logical channel identifier inserted in packet header. An example mechanism to enforce bandwidth control is shaping/policing.
- Flow to logical channel mapping: a mechanism to map application flow to logical channel. An example mechanism is identifying application flow by 5 tuple and inserting logical channel identifiers (such as VLAN, SRH) in packet header.

Note that when compute and data center network are controlled and managed as part of DCI Cluster, the compute is LSN composed by DCI physical nodes controlled by DCI Cluster Controller. Also, DN specific functions on DN-EP may be an application on top of DCI Cluster, or may be embedded as part of DCI Cluster. Further details are described in Annex E.

C.2. DN-G Implementation Example

Figure C-2 shows a reference model DN-G.

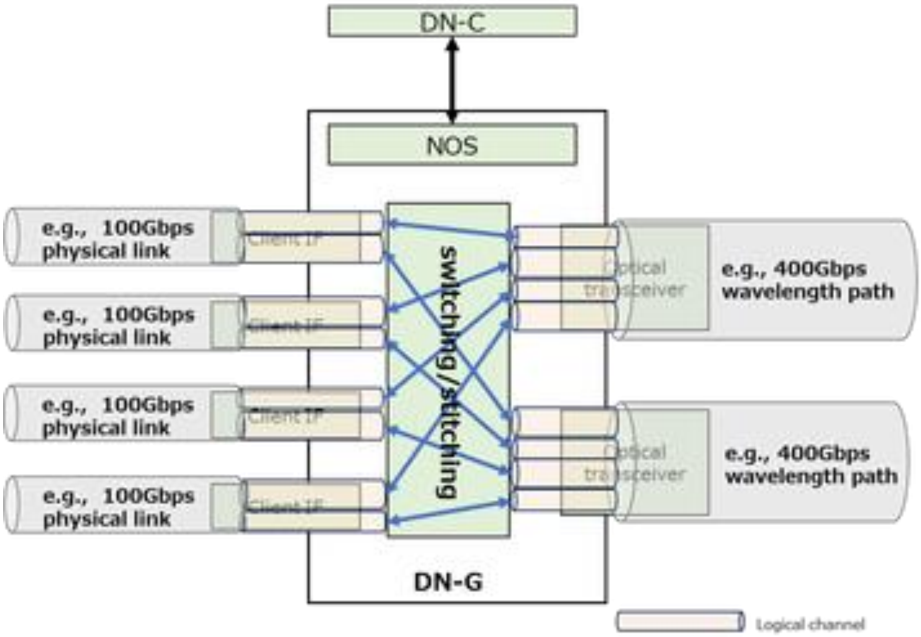


Figure C-2: DN-G Reference Model

DN-G consists of the following functional blocks.

- NOS (Network OS): A mechanism to communicate with DN-C for controlling and managing optical transport path endpoints, mapping logical channels to optical transport paths and switching/stitching logical channels.
- Client interface: A mechanism to send/receive data.
- Optical transceiver: An endpoint of optical transport path, such as APN-T (Open APN.WX), APN-T.FX (Open APN.FX), APN-T (MP-H), APN-T (MP-L) or even grey optics.
- Switching/stitching: A mechanism to enforce connecting client interface side of logical channel and optical transceiver side of logical channel, by mapping logical channels to optical transport paths. Also, a mechanism for enforcing performance management and fault management per logical channel, and for enforcing bandwidth control per logical channel. An example mechanism to enforce bandwidth control is shaping/policing or TDM-based bandwidth allocation. Note that when bandwidth control at DN-EP is performed, it is a choice whether to enforce bandwidth control at DN-G (see Section 4.4).

Note that APN-T may be controlled and managed by DN-C or by APN-C. In the latter case, dual access (DN-C and APN-C) to DN-G should be allowed. Example access methods between NOS of DN-G and APN-T are CMIS and SFF-8636 as Management Interfaces.

Figure C-3 shows several types of DN-G implementation examples, based on available technologies. It is recommended to choose DN-G type based on application requirements.

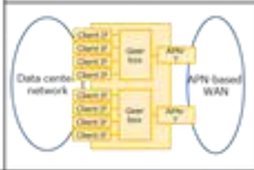
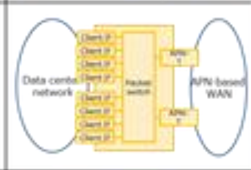
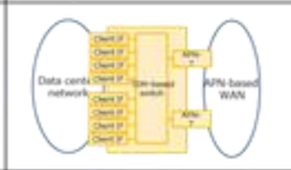
		Type1: Muxponder	Type 2: Packetponder (packet switch + APN-T)	Type3: TDM-based switchponder (assuming OTN SW or FlexE SW + APN-T)
				
Logical Channel	APN-T side	OTN	Packet-based (MPLS LSP, SR)	TDM-based (ODUflex or FlexE)
	Client side	Port	Port/VLAN/Packet-based (MPLS, SR etc)	Port/VLAN/OTN (OTN) Port/VLAN/FlexE (FlexE) and packet-based if packet processing is embedded
	Traffic separation per logical channel (strict bandwidth, delay)	Hard separation	Soft separation	Hard separation
	Logical channel bandwidth unit	per client IF (e.g., 100G)	Very granular	1.25G (ODUflex) 5G (FlexE)
	OAM & recovery	NA (e.g., by optical line protection)	Packet OAM & recovery (e.g.,MPLS)	OTN OAM & recovery FlexE OAM & recovery
Client interface types		Mainly Ethernet, OTN	Mainly Ethernet	Mainly Ethernet, OTN (OTN) Mainly Ethernet (FlexE)
Line interface types		OTN (OpenROADM-based)	OTN (OpenROADM-based), ZR/ZR+	OTN (OpenROADM-based) (OTN) OTN (OpenROADM-based) , ZR/ZR+ (FlexE)
Logical channel switching/switching pattern restriction		Client to Line MUX only	Client to Line Switch Client to Client Switch Line to Line Switch	Client to Line Switch Client to Client switch (optional) Line to Line switch (optional)

Figure C-3: DN-G Implementation Examples

- Type1: Muxponder
  - QoS-guaranteed mux/demux is possible (e.g., 100G x 4 mux/demux 400G).
  - Flexibility is limited (e.g., coarse logical channel bandwidth unit, client to line MUX only).
- Type2: Packetponder
  - Strict QoS control (especially delay/jitter bound) is challenging to achieve.
  - Flexibility is possible (e.g., very granular logical channel bandwidth unit, client to line Switch).
- Type3: TDM-based switchponder
  - Strict QoS control is possible (may depend on implementation).
  - Flexibility is possible (e.g., granular logical channel bandwidth unit, client to line Switch) (may depend on implementation).

C.3. DN-I Implementation Example

Figure C-4 shows a reference model DN-I.

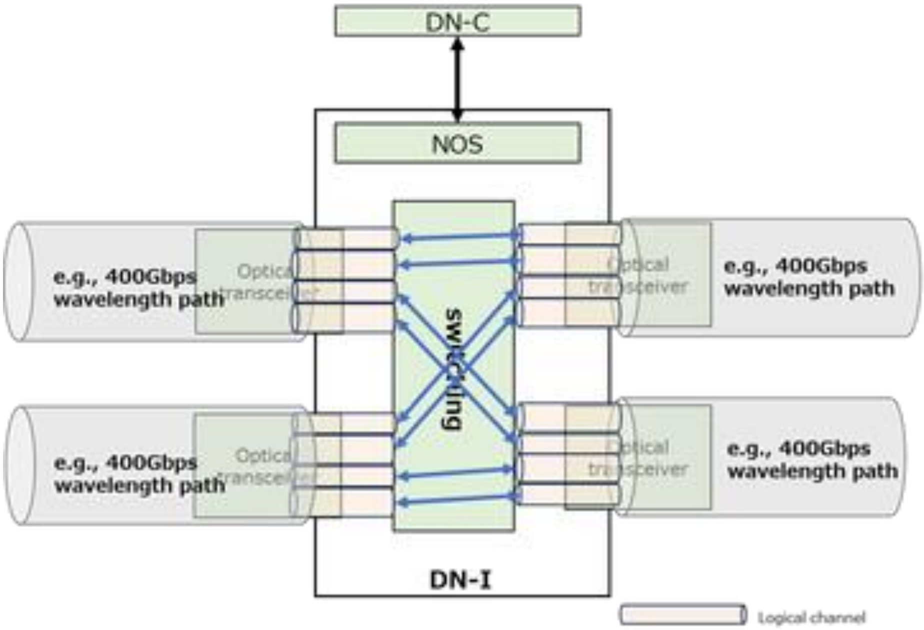


Figure C-4: DN-I Reference Model

Similar to DN-G, there are several types of DN-I implementation examples, by switching technologies used (i.e., packet-based, TDM-based). DN-I requires large switching capacity (as a cross-connect).

C.4. DN-C Implementation Example

Figure C-5 shows an example of DN-C functional model. The DN-C is a logical entity. All the functions of DN-C can be implemented together in a single box or separately in multiple boxes. An example is that DN-EP control functional block may be implemented separately, or may be implemented as part of DCI Cluster Controller. Further details are described in Annex E.

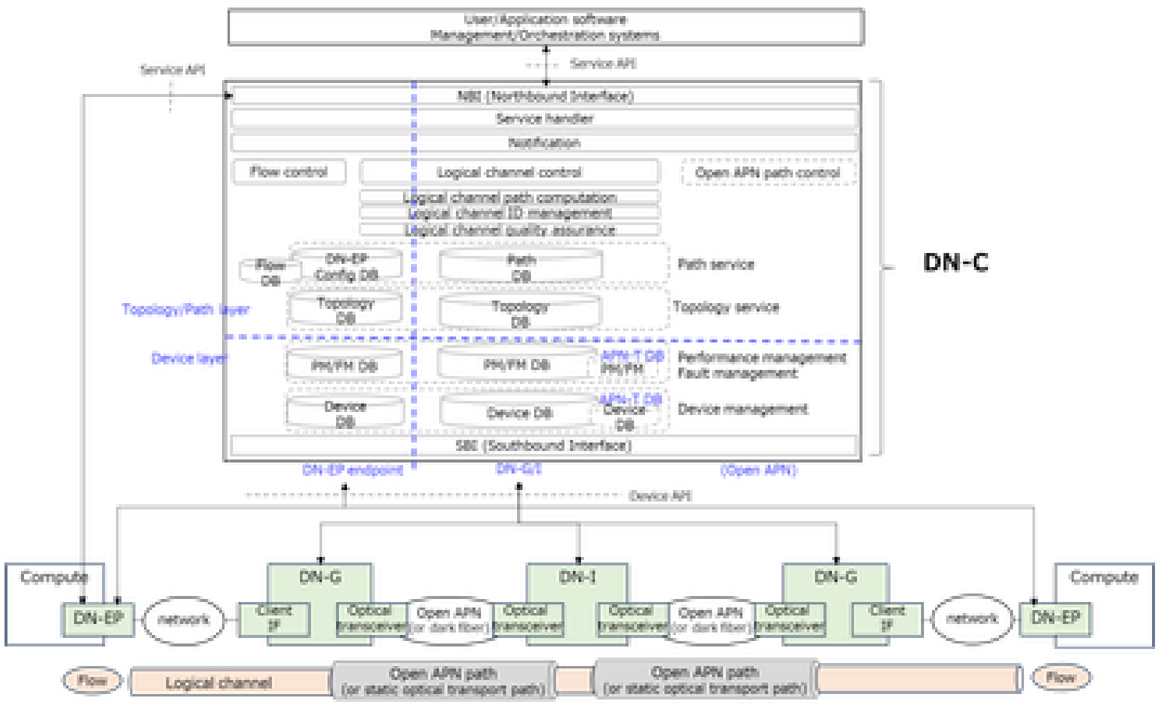


Figure C-5: DN-C Implemetation Example

DN-C needs to control and manage logical channels. Furthermore, DN-C may need to control and manage application flow to logical channel mapping when this is performed as part of Deterministic Networking service. DN-C needs to maintain DN-EP related information, as well as DN-G/DN-I related information.

DN-C consists of the following functional blocks.

- Device management: a mechanism to maintain devices. DN-C needs to maintain DN-EPs as well as DN-Gs/DN-Is.
- Performance management/Fault management: a mechanism to monitor performance information and fault information. DN-C needs to monitor DN-EPs as well as DN-Gs/DN-Is.
- Topology service: a mechanism to maintain topology and to provide topology information. Topology between DN-EP and DN-G, as well as between DN-Gs/DN-Is are maintained.
- Path service: a mechanism to maintain path (i.e., service provided by logical channels through service request) and to provide path information.
- Logical channel path computation: a mechanism to compute route of logical channels, considering performance parameters specified. Note that this document does not specify path computation algorithm itself.
- Logical channel ID management: a mechanism to maintain logical channel identifiers (including switching and stitching).
- Logical channel quality assurance: a mechanism to monitor and analyze PM/FM in real-time and determine whether performance parameters are satisfied for each logical channel, and to enforce any recovery methods to satisfy where necessary.
- Logical channel control: a mechanism to maintain logical channels, by communicating with relevant functional blocks.
- Flow control: a mechanism to maintain application flow to logical channel mapping, by communication with relevant functional blocks.

Note that there are several options how APN-Ts are controlled and managed, as well as how Open APN paths are controlled and managed, as mentioned in Annex C.2.

Service API is described in Section 4.5.

Device API consists of the following.

- Logical channel setup: DN-C communicates with DN-EPs and DN-Gs/DN-Is to setup logical channels. DN-C communicates with DN-EPs and instructs logical channel parameters (such as logical channel identifier, route, bandwidth). DN-C communicates with DN-Gs/DN-Is and instructs logical channel switching/stitching parameters (such as logical channel identifier, route, bandwidth). DN-C may further communicate with DN-EPs and DN-Gs/DN-Is for configuring monitoring (such as telemetry) and reliability mechanisms.

- Logical channel delete: DN-C communicates with DN-EPs and DN-Gs/DN-Is to delete already established logical channels.
- Logical channel update: DN-C communicates with DN-EPs and DN-Gs/DN-Is to update parameters of already established logical channels, including route, bandwidth and reliability mechanisms.
- Flow mapping setup: DN-C communicates with DN-EPs to setup application flow to logical channel mapping.
- Flow mapping delete: DN-C communicates with DN-EPs to delete already established application flow to logical channel mapping.
- Flow mapping update: DN-C communicates with DN-EPs to update already established application flow to logical channel mapping.
- Logical channel monitoring: DN-C communicates with DN-EPs and DN-Gs/DN-Is to obtain monitoring information of logical channels being configured. Monitoring information may be pulled from DN-C, or may be pushed from DN-EP/DN-G/DN-I, based on configuration.
- Device information retrieval: Optionally, DN-C communicates with DN-EPs and DN-Gs/DN-Is to retrieve device information as inventory.

In addition, APN-T control and management may be included as part of device API.

Note that when a service API is invoked by DN-EP (Figure 4-8), the service API may work as a device API as well, meaning reply of the service API (from DN-C to DN-EP) may contain DN-EP configuration related information. This is a choice of implementation.

D. Integration into GPU Cluster Network

A typical GPU cluster network for AI/ML is show in Figure D-1. It has a backend network, which provides loss-less connectivity for GPU-to-GPU communication, as well as GPU-to-storage communication. It also has a frontend network, which provides lossy connectivity for access from/to user premises over WAN.

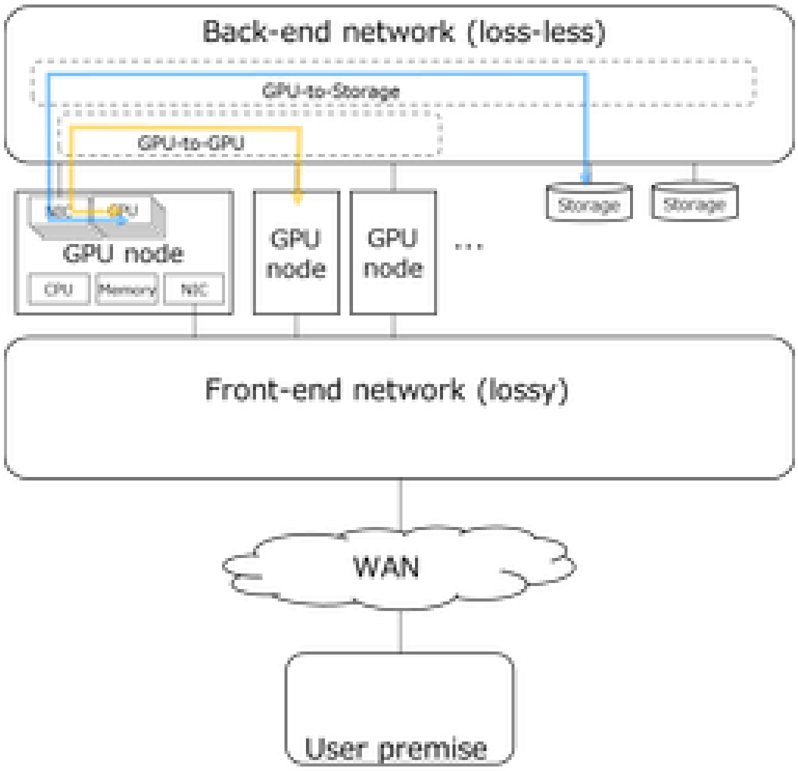


Figure D-1: Typical GPU Cluster Network

For use cases that utilize GPU-based infrastructure, it is expected to utilize GPUs located in cloud/co-location data centers while data is located at user premises and sent or accessed. This allows cost effective and greener use of GPU infrastructure. Here, we need to provide NIC-to-NIC deterministic connections (loss-less) across GPU cluster network, WAN and user premises.

Figure D-2 shows an example how to integrate such NIC-to-NIC deterministic connections into the GPU cluster network. Storage communication from/to user premises is bypassing the front-end network, and directly connected to GPU nodes. Details are for further study, considering performance, cost, and security.

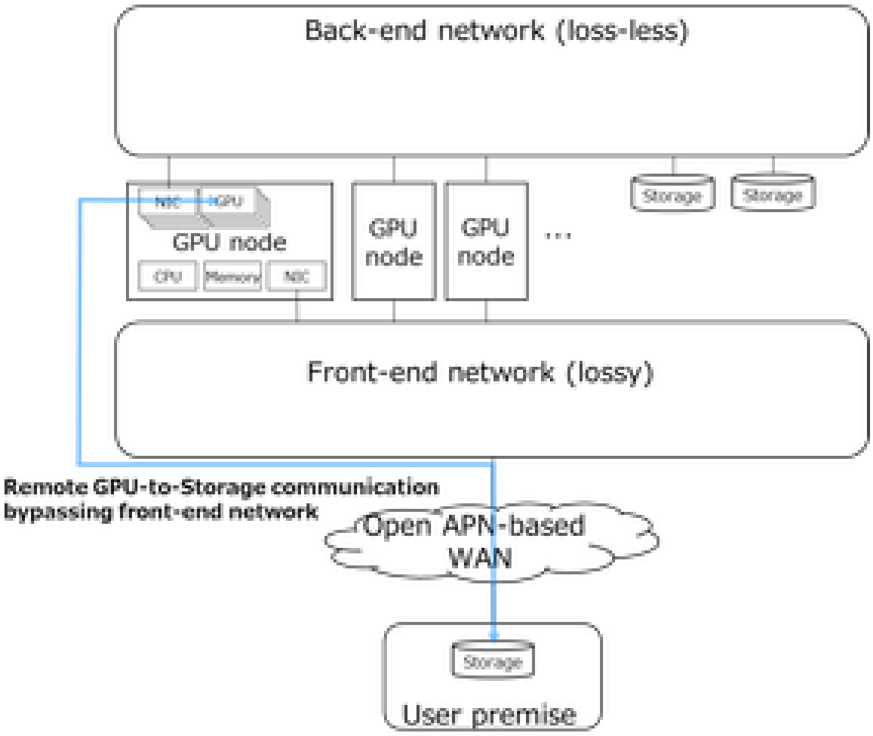


Figure D-2: NIC-to-NIC Deterministic Connection across Open APN-based WAN

E. Details on Relationship with Data-Centric Infrastructure (DCI) Functional Architecture

This annex described two potential models in terms of relationship of DCI and Deterministic Networking with Software-Defined Optical Interconnect. Note that two models are described for further refinement. There could be other models, or two models could be further refined.

The first model is considering DCI Cluster as a physical layer, while considering DN as a logical layer as shown in Figure E-1. In this case, DCI Cluster Controller constructs LSN from DCI physical nodes. DN-C constructs NIC to NIC deterministic connections, by configuring NICs (DN-EPs) on LSNs, DN-Gs and data center network where necessary. This model allows functional split between physical layer and logical layer, and application specific functions can be implemented in DN logical layer.

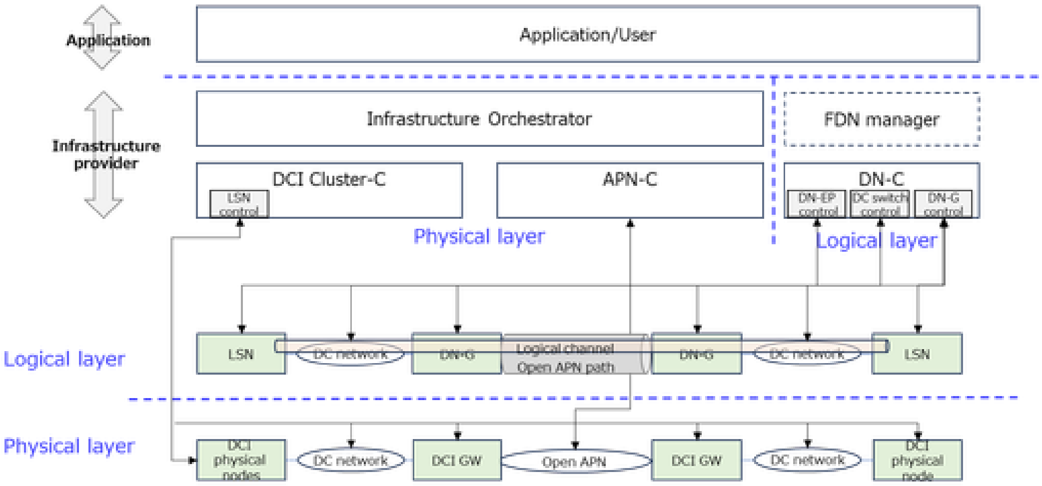


Figure E-1: DCI Cluster as Physical Layer, DN as Logical Layer

The second model is considering DCI Cluster as devices of DCI Cluster, while considering DN as devices of DN, as shown in Figure E-2. In this case, part of DN-C functions is implemented as part of DCI Cluster Controller, namely DN-EP control functions and data center switch control functions. DN-EP control functions may be implemented as tenant platform on top of DCI Cluster [1]. DCI-GW and DN-G may become identical. DCI Cluster Controller controls LSN/DCI physical nodes, as well as data center network. DN-C controls DCI-GWs/DN-Gs. Alternatively, DCI Cluster Controller controls DCI-GWs/DN-G as well, depending on demarcation. This model allows functional split per device.

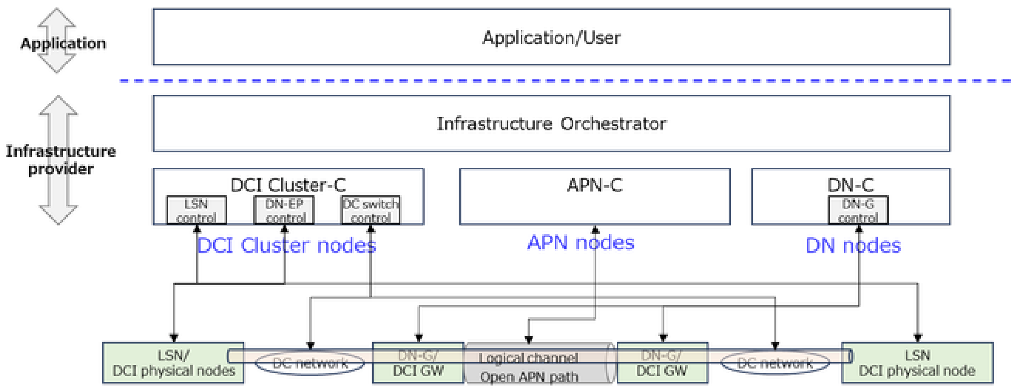


Figure E-2: DCI Cluster as DCI Cluster Nodes, DN as DN Nodes

References

NO	REFERENCE
[1]	Data-Centric Infrastructure Functional Architecture Version 2.0
[2]	Services Infrastructure for Financial Industry Use Case Version 2.0
[3]	Remote Media Production for Broadcast Industry Use Case Version 1.0
[4]	Green Computing with Remote GPU Service for Generative AI / LLM Use Case - Light Speed Data Transfer for AI Training - Version 1.0
[5]	Reference Implementation Model (RIM) for the Area Management Security Use Case Version 1.0
[6]	Reference Implementation Model (RIM) for the Interactive Live Music Entertainment Use Case Version 1.0
[7]	Reference Implementation Model (RIM) for the Remote Controlled Robotic Inspection Use Case Version 1.0
[8]	Open All-Photonic Network Functional Architecture Version 2.0
[9]	1588-2019 - IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems

History

Revision	Release Date	Summary of Changes
1.0	July 24, 2025	Initial version