Teas Working Group Young Lee Internet Draft Huawei

Intended status: Informational Sergio Belotti

Nokia

Expires: December 30, 2017

Dhruv Dhody Huawei

Daniele Ceccarelli Ericsson

> Bin Young Yun ETRI

> June 30, 2017

#### Information Model for Abstraction and Control of TE Networks (ACTN)

draft-ietf-teas-actn-info-model-02.txt

#### Abstract

This draft provides an information model for Abstraction and Control of Traffic Engineered (TE) networks (ACTN).

#### Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of  $\underline{\text{BCP }78}$  and  $\underline{\text{BCP }79}$ .

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at <a href="http://www.ietf.org/ietf/lid-abstracts.txt">http://www.ietf.org/ietf/lid-abstracts.txt</a>

The list of Internet-Draft Shadow Directories can be accessed at <a href="http://www.ietf.org/shadow.html">http://www.ietf.org/shadow.html</a>.

This Internet-Draft will expire on December 30, 2017.

## Copyright Notice

Copyright (c) 2017 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to <a href="BCP-78">BCP-78</a> and the IETF Trust's Legal Provisions Relating to IETF Documents (<a href="http://trustee.ietf.org/license-info">http://trustee.ietf.org/license-info</a>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the <a href="mailto:Trust Legal Provisions">Trust Legal Provisions</a> and are provided without warranty as described in the Simplified BSD License.

#### Table of Contents

Introduction.	
1.1. Terminol	logy <u>4</u>
ACTN Common 3	Interfaces Information Model <u>4</u>
2.1. VN Actio	on Primitives <u>5</u>
<u>2.1.1</u> . VN	Instantiate
2.1.2. VN	Modify
2.1.3. VN	Delete
	Update
2.1.5. VN	Path Compute
	Query
2.1.7. TE	Update (for TE resources)8
2.2. VN Object	cts
_	Identifier9
	Service Characteristics9
2.2.3. VN	End-Point11
	Objective Function12
2.2.5. VN	Action Status <u>13</u>
	Associated LSP <u>13</u>
	Computed Path
	Service Preference13
	of VN Primitives with VN Objects
	1.1. Terminol ACTN Common 3 2.1. VN Actic 2.1.1. VN 2.1.2. VN 2.1.3. VN 2.1.4. VN 2.1.5. VN 2.1.6. VN 2.1.7. TE 2.2. VN 0bjec 2.2.1. VN 2.2.2. VN 2.2.2. VN 2.2.3. VN 2.2.4. VN 2.2.5. VN 2.2.6. VN 2.2.7. VN 2.2.8. VN 2.2.8. VN 2.3. Mapping

#### 1. Introduction

This draft provides an information model for the requirements identified in the ACTN requirements [ACTN-Req] and the ACTN interfaces identified in the ACTN architecture and framework document [ACTN-Frame].

The purpose of this draft is to put all information elements of ACTN in one place before proceeding to development work necessary for protocol extensions and data models.

The ACTN reference architecture identified a three-tier control hierarchy as depicted in Figure 1:

- Customer Network Controllers (CNC)
- Multi-Domain Service Coordinator (MDSC)
- Physical Network Controllers (PNC).

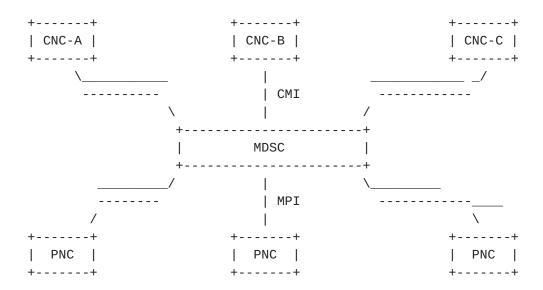


Figure 1: A Three-tier ACTN control hierarchy

The two interfaces with respect to the MDSC, one north of the MDSC and the other south of the MDSC are referred to as CMI (CNC-MDSC Interface) and MPI (MDSC-PNC Interface), respectively. It is intended to model these two interfaces and derivative interfaces thereof (e.g., MDSC to MSDC in a hierarchy of MDSCs) with one common model.

## **1.1**. Terminology

Refer VN, VNS to  $[\underline{\text{ACTN-Frame}}]$  and abstraction and abstract topology to  $[\underline{\text{RFC7926}}]$ .

## 2. ACTN Common Interfaces Information Model

This section provides ACTN common interface information model to describe in terms of primitives, objects, their properties

(represented as attributes), their relationships, and the resources for the service applications needed in the ACTN context.

The standard interface is described between a client controller and a server controller. A client-server relationship is recursive between a CNC and a MDSC and between a MDSC and a PNC. In the CMI, the client is a CNC while the server is a MDSC. In the MPI, the client is a MDSC and the server is a PNC. There may also be MDSC-MDSC interface(s) that need to be supported. This may arise in a hierarchy of MDSCs in which workloads may need to be partitioned to multiple MDSCs.

Basic primitives (messages) are required between the CNC-MDSC and MDSC-PNC controllers. These primitives can then be used to support different ACTN network control functions like network topology request/query, VN service request, path computation and connection control, VN service policy negotiation, enforcement, routing options, etc.

At a minimum, the following VN action primitives should be supported:

- VN Instantiate (See <u>Section 2.1.1</u>. for the description)
- VN Modify (See <u>Section 2.1.2</u>. for the description)
- VN Delete (See <u>Section 2.1.3</u>. for the description)
- VN Update ((See <u>Section 2.1.4</u>. for the description)
- VN Path Compute (See <u>Section 2.1.5</u>. for the description)
- VN Query (See <u>Section 2.1.6</u>. for the description)

In addition to VN action primitives, TE Update primitive should also be supported (See Section 2.1.7. for the description).

Note that all VN action primitives defined above are applicable only to CMI while TE update primitive is applicable to both CMI and MPI."

#### **2.1**. VN Action Primitives

This section provides a list of main primitives necessary to satisfy ACTN requirements specified in [ACTN-REQ].

<VN Action> describes main primitives. VN Action can be one of the
following primitives: (i) VN Instantiate; (ii) VN Modify; (iii) VN
Delete; (iv) VN Update; (v) VN Path Compute; (vi) VN Query.

### 2.1.1. VN Instantiate

<VN Instantiate> refers to an action from customers/applications to
request their VNs. This primitive can also be applied from an MDSC
to a PNC requesting a VN (if the domain the PNC supports can
instantiate the entire VN) or a part of VN elements. Please see the
definition of VN in [ACTN-Frame].

## 2.1.2. VN Modify

<VN Modify> refers to an action from customers/applications to
modify an existing VN (i.e., instantiated VN). This primitive can
also be applied from an MDSC to a PNC requesting a VN (if the domain
the PNC supports can instantiate the entire VN) or a part of VN
elements.

#### 2.1.3. VN Delete

<VN Delete> refers to an action from customers/applications to delete an existing VN. This primitive can also be applied from an MDSC to a PNC requesting a VN (if the domain the PNC supports can instantiate the entire VN) or a part of VN elements.

## **2.1.4**. VN Update

<VN Update> refers to any update to the VN that need to be updated to the subscribers. VN Update fulfills a push model at CMI level, to make aware customers of any specific changes in the topology details related to VN instantiated.

Note the VN Update means the connection-related information (e.g., LSPs) update that has association with VNs.

#### 2.1.5. VN Path Compute

<VN Path Compute> consists of Request and Reply. Request refers to
an action from customers/applications to request a VN path
computation. This primitive can also be applied from an MDSC to a
PNC requesting a VN (if the domain the PNC supports can instantiate
the entire VN) or a part of VN elements.

<VN Path Compute> Reply refers to the reply in response to <VN Path Compute> Request.

<VN Path Compute> Request/Reply is to be differentiated from a VN Instantiate. The purpose of VN Path Compute is a priori exploration to estimate network resources availability and getting a list of possible paths matching customer/applications constraints. To make this type of request Customer/application controller can have a shared (with lower controller) view of an abstract network topology on which to get the constraints used as input in a Path Computation request. The list of paths obtained by the request can be used by customer/applications to give path constrains during VNS connectivity request and to compel the lower level controller (e.g. MDSC) to select the path that Client/application controller has chosen among the set of paths returned by the Path Computation primitives. The importance of this primitives is for example in a scenario like multi-domain in which the optimal path obtained by an orchestrator as sum of optimal paths for different domain controller cannot be the optimal path in the Client/application controller prospective. This only applies between CNC and MDSC.

## <u>2.1.6</u>. VN Query

<VN Query> refers to any query pertaining to the VN that has been already instantiated. VN Query fulfills a pull model and permit to get topology view.

<VN Query Reply> refers to the reply in response to <VN Query>.

#### **2.1.7**. TE Update (for TE resources)

<TE Update> it is a primitives specifically related to MPI
interface to provide TE resource update between any domain
controller towards MDSC regarding the entire content of any "domain
controller" TE topology or an abstracted filtered view of TE
topology depending on negotiated policy.

```
<TE Update> ::= [<Abstraction>]<TE-topology...>
<TE-topology> ::= <TE-Topology-reference> <Node-list> <Link-list>
<Node-list> ::= <Node>[<Node-list>]
<Node> ::= <Node> <TE-Termination Points>
<Link-list> ::= <Link>[<Link-list>]
Where
<Abstraction> provides information on level of abstraction (as
determined a priori).
<TE-topology-reference> ::= information related to the specific te-
topology related to nodes and links present in this TE-topology.
<Node-list> ::= detailed information related to a specific node
belonging to a te-topology e.g. te-node-attributes [TE-TOPO].
<Link-list> ::= information related to the specific link related
belonging to a te-topology e.g. te-link-attributes [TE-TOPO].
<TE-Termination Points> ::= information details associated to the
termination point of te-link related to a specific node e.g.
interface-switching-capability [TE-TOPO].
```

## 2.2. VN Objects

This section provides a list of objects associated to VN action primitives.

#### 2.2.1. VN Identifier

<VN Identifier> is a unique identifier of the VN.

## 2.2.2. VN Service Characteristics (applicable only to CMI)

VN Service Characteristics describes the customer/application requirements against the VNs to be instantiated.

<VN Service Characteristics> ::= <VN Connectivity Type>

(<VN Traffic Matrix>...)

<VN Survivability>

Where

<VN Connectivity Type> ::= <P2P>|<P2MP>|<MP2MP>|<MP2P>|<Multidestination>

The Connectivity Type identifies the type of required VN Service. In addition to the classical type of services (e.g. P2P/P2MP etc.), ACTN defines the "multi-destination" service that is a new P2P service where the end points are not fixed. They can be chosen among a list of pre-configured end points or dynamically provided by the CNC.

<VN Traffic Matrix> ::= <Bandwidth>

[<VN Constraints>]

The VN Traffic Matrix represents the traffic matrix parameters required against the service connectivity required and so the VN request instantiation between service related Access Points [ACTN-Frame]. Bandwidth is a mandatory parameter and a number of optional constrains can be specified in the <VN Constrains> (e.g. diversity, cost). They can include objective functions and TE metrics bounds as specified in [RFC5441].

Further details on the VN constraints are specified below:

<VN Constraints> ::= [<Layer Protocol>]

[<Diversity>]

[<Shared Risk>]

```
( <Metric> | <VN Objective Function> )
```

#### Where:

<Layer Protocol> Identifies the layer topology at which the VN
service is requested. It could be for example MPLS, ODU, and OCh.

<Diversity> This allows asking for diversity constraints for a VN
Instantiate/Modify or a VN Path Compute. For example, a new VN or
a path is requested in total diversity from an existing one (e.g.
diversity exclusion).

<Shared Risk> Based on the realization of VN required, group of
physical resources can be impacted by the same risk. An E2E
tunnel can be impacted by this shared risk. This is used to get
the SRLG associated with the different tunnels composing a VN.

<Metric> can include all the Metrics (cost, delay, delay
variation, latency), bandwidth utilization parameters defined and
referenced by [RFC3630] and [RFC7471].

<VN Objective Function> See <u>Section 2.2.4</u>.

<VN Survivability> describes all attributes related to the VN recovery level and its survivability policy enforced by the customers/applications.

#### Where:

<VN Recovery Level> It is a value representing the requested
level of resiliency required against the VN. The following
values are defined:

- . Unprotected VN
- . VN with per tunnel recovery: The recovery level is defined against the tunnels composing the VN and it is specified in the <VN Tunnel Recovery Level>.

<On the fly restoration>

The VN Tunnel Recovery Level indicates the type of protection or restoration mechanism applied to the VN. It augments the recovery types defined in [RFC4427].

<VN Survivability Policy> ::= [<Local Reroute Allowed>]

[<Domain Preference>]

[<Push Allowed>]

[<Incremental Update>]

#### Where:

<Local Reroute Allowed> is a delegation policy to the Server
to allow or not a local reroute fix upon a failure of the
primary LSP.

<Domain Preference> is only applied on the MPI where the MDSC
(client) provides a domain preference to each PNC
(server).e.g. when a inter-domain link fails, then PNC can
choose the alternative peering with this info.

<Push Allowed> is a policy that allows a server to trigger an updated VN topology upon failure without an explicit request from the client. Push action can be set as default unless otherwise specified.

<Incremental Update> is another policy that triggers an
incremental update from the server since the last period of
update. Incremental update can be set as default unless
otherwise specified.

## 2.2.3. VN End-Point

<VN End-Point> Object describes the VN's customer end-point
characteristics.

<VN End-Point> ::= (<Access Point Identifier>

[<Access Link Capability>]

[<Source Indicator>])...

#### Where:

<Access point identifier> It represents a unique identifier of the client end-point. They are used by the customer to ask for the setup of a virtual network creation. A <VN End-Point> is defined against each AP in the network and is shared between customer and provider. Both the customer and the provider will map it against his own physical resources.

<Access Link Capability> An optional object that identifies the
capabilities of the access link related to the given access point.
(e.g., max-bandwidth, bandwidth availability, etc.)

<Source Indicator> indicates if an End-point is source or not.

#### 2.2.4. VN Objective Function

The VN Objective Function applies to each VN member (i.e., each E2E tunnel) of a VN.

The VN Objective Function can reuse objective functions defined in [RFC5541] section 4.

For a single path computation, the following objective functions are defined:

- o MCP is the Minimum Cost Path with respect to a specific metric (e.g. shortest path).
- o MLP is the Minimum Load Path, that means find a path composted by te-link least loaded.
- o MBP is the Maximum residual Bandwidth Path.

For a concurrent path computation, the following objective functions are defined:

- o MBC is to Minimize aggregate Bandwidth Consumption.
- o MLL is to Minimize the Load of the most loaded Link.
- o MCC is to Minimize the Cumulative Cost of a set of paths.

#### 2.2.5. VN Action Status

<VN Action Status> is the status indicator whether the VN has been successfully instantiated, modified, or deleted in the server network or not in response to a particular VN action.

Note that this action status object can be implicitly indicated and thus not included in any of the VN primitives discussed in <u>Section</u> 2.3.

#### 2.2.6. VN Associated LSP

<VN Associated LSP> describes the instantiated LSPs that is
associated with the VN. <VN Associated LSP> is used between each
domain PNC and the MDSC as part of VN Update once the VN is
instantiated in each domain network and when CNC want to have more
details about the topology instantiated as consequence of a VN
Instantiate.

```
<VN Associated LSP> ::= <VN Identifier> (<LSP>...)
```

#### 2.2.7. VN Computed Path

The VN Computed Path is the list of paths obtained after the VN path computation request from higher controller. Note that the computed path is to be distinguished from the LSP. When the computed path is signaled in the network (and thus the resource is reserved for that path), it becomes an LSP.

```
<VN Computed Path> ::= (<Path>...)
```

## 2.2.8. VN Service Preference

This section provides VN Service preference. VN Service is defined in  $\frac{\text{Section 2}}{\text{Section 2}}$ .

Where

<Location Service Preference describes the End-Point Location's
(e.g. Data Centers) support for certain Virtual Network Functions
(VNFs) (e.g., security function, firewall capability, etc.)and is
used to find the path that satisfies the VNF constraint.</pre>

<Client-specific Preference> describes any preference related to
Virtual Network Service (VNS) that application/client can enforce
via CNC towards lower level controllers. For example, permission
the correct selection from the network of the destination related
to the indicated VNF It is e.g. the case of VM migration among
data center and CNC can enforce specific policy that can permit
MDSC/PNC to calculate the correct path for the connectivity
supporting the data center interconnection required by
application.

<End-Point Dynamic Selection Preference> describes if the EndPoint (e.g. Data Center) can support load balancing, disaster
recovery or VM migration and so can be part of the selection by
MDSC following service Preference enforcement by CNC.

### 2.3. Mapping of VN Primitives with VN Objects

This section describes the mapping of VN Primitives with VN Objects based on <u>Section 2.2</u>.

```
<VN Delete> ::= <VN Identifier>
<VN Update> :: = <VN Identifier>
                <VN Associated LSP>
<VN Path Compute Request> ::= <VN Service Characteristic>
                              <VN End-Point>
<VN Path Compute Reply> ::= <VN Computed Path>
<VN Query> ::= <VN Identifier>
<VN Query Reply> ::= <VN Identifier>
                     <VN Associated LSP>
```

## 3. References

#### **3.1.** Normative References

[DRAFT-SER-AWARE] Dhruv Dhody, Qin Wu, Vishwas Manral, Zafar Ali, and Kenji Kumaki, "Extensions to the Path Computation Element Communication Protocol (PCEP) to compute service aware Label Switched Path (LSP).," June 2016, draft-ietfpce-pcep-service-aware-10.

#### 3.2. Informative References

- [TE-TOPO] Liu, X. et al., "YANG Data Model for TE Topologies", <u>draft-ietf-teas-yang-te-topo</u>, work in progress.Informative References
- [ACTN-Req] Y. Lee, et al., "Requirements for Abstraction and Control of Transport Networks", <u>draft-lee-teas-actn-requirements</u>, work in progress.
- [ACTN-Frame] D. Ceccarelli, et al., "Framework for Abstraction and Control of Transport Networks", <u>draft-ceccarelli-teas-actn-framework</u>, work in progress.
- [Stateful-PCE] E. Crabbe, et al., "PCEP Extensions for Stateful PCE", <a href="mailto:draft-ietf-pce-stateful-pce">draft-ietf-pce-stateful-pce</a>, work in progress.
- [RFC5541] JL. Le Roux, JP. Vasseur and Y. Lee, "Encoding of Objective Functions in the Path Computation Element Communication Protocol (PCEP)", RFC 5541, June 2009.

#### 4. Contributors

Contributors' Addresses

Authors' Addresses

Young Lee (Editor) Huawei Technologies 5340 Legacy Drive Plano, TX 75023, USA Phone: (469)277-5838

Email: leeyoung@huawei.com

Sergio Belotti (Editor) Alcatel Lucent Via Trento, 30 Vimercate, Italy Email: sergio.belotti@alcatel-lucent.com

Dhruv Dhody

Huawei Technologies, Divyashree Technopark, Whitefield Bangalore, India Email: dhruv.ietf@gmail.com

Daniele Ceccarelli Ericsson Torshamnsgatan,48 Stockholm, Sweden Email: daniele.ceccarelli@ericsson.com

Bin Young Yun
ETRI

Email: byyun@etri.re.kr

Haomian Zheng Huawei Technologies

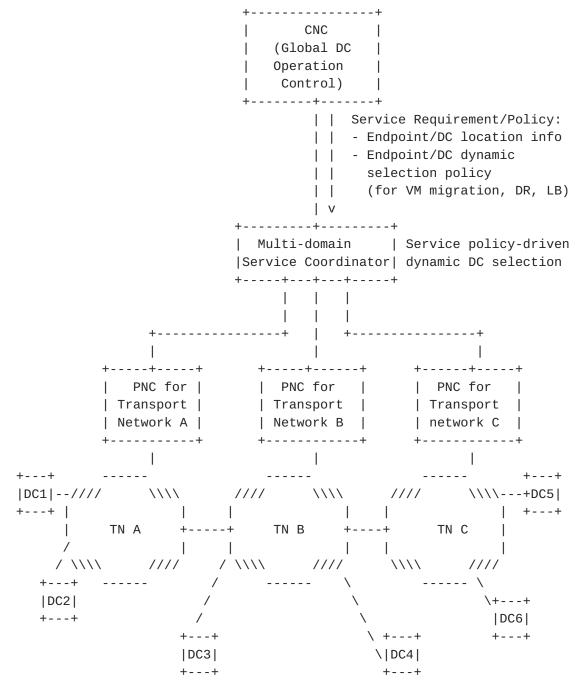
Email: zhenghaomian@huawei.com

Xian Zhang Huawei Technologies

Email: zhang.xian@huawei.com

## Appendix A: ACTN Applications

## A.1. Coordination of Multi-destination Service Requirement/Policy



DR: Disaster Recovery LB: Load Balancing

Figure A.1: Service Policy-driven Data Center Selection

Figure A.1 shows how VN service policies from the CNC are incorporated by the MDSC to support multi-destination applications. Multi-destination applications refer to applications in which the selection of the destination of a network path for a given source needs to be decided dynamically to support such applications.

Data Center selection problems arise for VM mobility, disaster recovery and load balancing cases. VN's service policy plays an important role for virtual network operation. Service policy can be static or dynamic. Dynamic service policy for data center selection may be placed as a result of utilization of data center resources supporting VNs. The MDSC would then incorporate this information to meet the service objective of this application.

## A.2. Application Service Policy-aware Network Operation

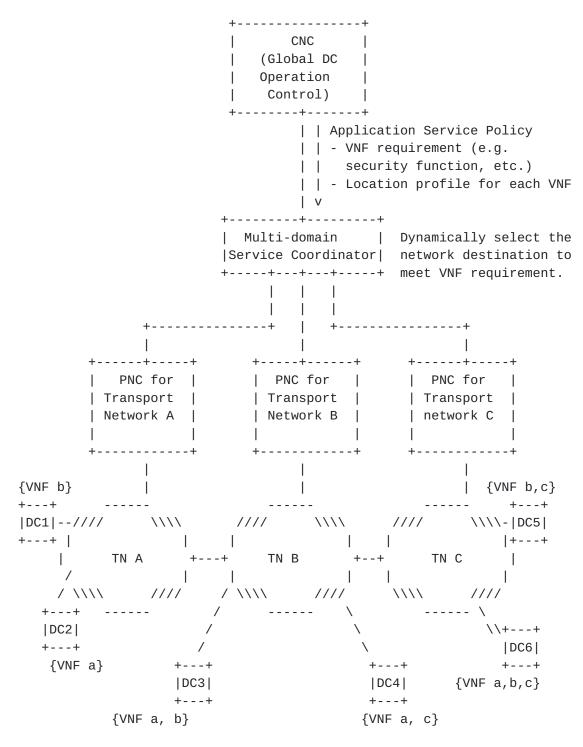


Figure A.2: Application Service Policy-aware Network Operation

This scenario is similar to the previous case in that the VN service policy for the application can be met by a set of multiple destinations that provide the required virtual network functions (VNF). Virtual network functions can be, for example, security functions required by the VN application. The VN service policy by the CNC would indicate the locations of a certain VNF that can be fulfilled. This policy information is critical in finding the optimal network path subject to this constraint. As VNFs can be dynamically moved across different DCs, this policy should be dynamically enforced from the CNC to the MDSC and the PNCs.

## A.3. Network Function Virtualization Service Enabled Connectivity

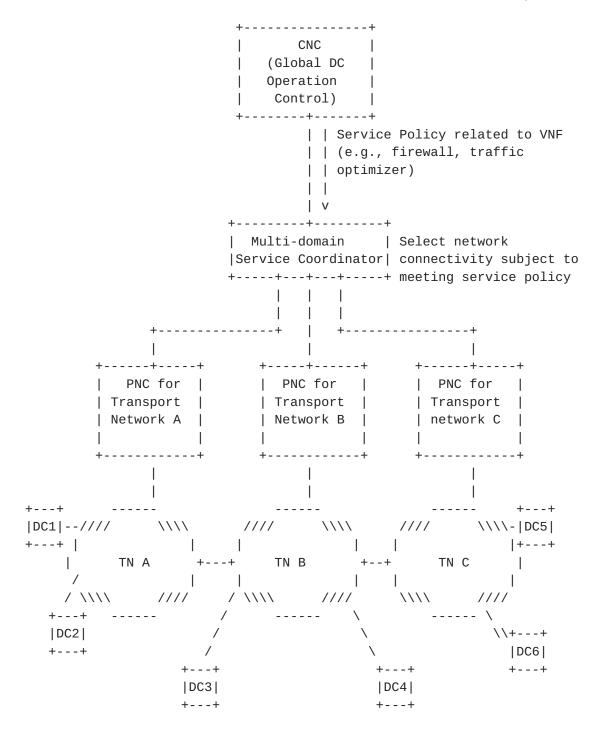


Figure A.3: Network Function Virtualization Service Enabled Connectivity

Network Function Virtualization Services are usually setup between customers' premises and service provider premises and are provided mostly by cloud providers or content delivery providers. The context may include, but not limited to a security function like firewall, a traffic optimizer, the provisioning of storage or computation capacity where the customer does not care whether the service is implemented in a given data center or another. The customer has to provide (and CNC is providing this) the type of VNF he needs and the policy associated with it (e.g. metric like estimated delay to reach where VNF is located in the DC). The policy linked to VNF is requested as part of the VN instantiation. These services may be hosted virtually by the provider or physically part of the network. This allows the service provider to hide his own resources (both network and data centers) and divert customer requests where most suitable. This is also known as "end points mobility" case and introduces new concepts of traffic and service provisioning and resiliency (e.g., Virtual Machine mobility).

# A.4. Dynamic Service Control Policy Enforcement for Performance and Fault Management

+			+
 	Customer Net	work Control	ler
1.Traffic  Monitor&   Optimize   Policy	/ \4.Traffic   Monitor   Result     opt	   5.Service   modify&  imize Req.	/ \   8.Traffic   modify &   optimize
+	/ \3.Traffic   Monitor	ervice Coord	+ / \  7.Path
\ /	   opt	modify&   imize Req.\	
	Physical Netw	ork Controll	+ er   +

Figure A.4: Dynamic Service Control for Performance and Fault Management

Figure A.4 shows the flow of dynamic service control policy enforcement for performance and fault management initiated by customer per VN. The feedback loop and filtering mechanism tailored for VNs performed by the MDSC differentiates this ACTN scope from traditional network management paradigm. VN level dynamic OAM data model is a building block to support this capability.

# A.5. E2E VN Survivability and Multi-Layer (Packet-Optical) Coordination for Protection/Restoration

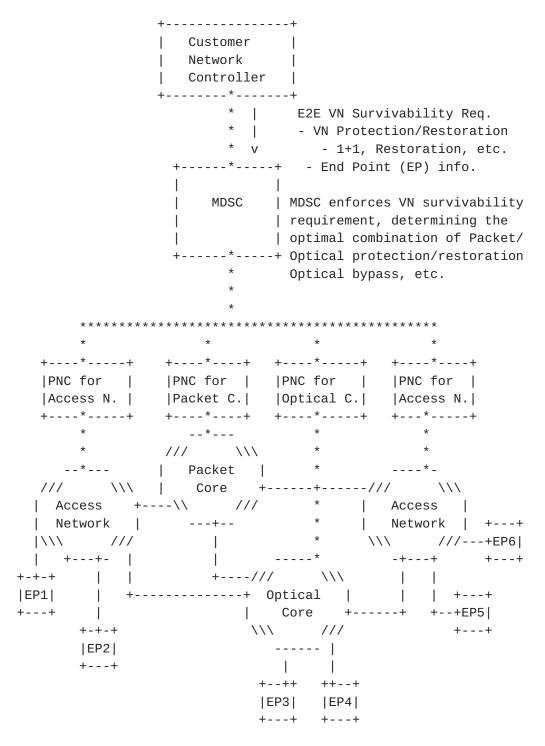


Figure A.5: E2E VN Survivability and Multi-layer Coordination for Protection and Restoration

Figure A.5 shows the need for E2E protection/restoration control coordination that involves CNC, MDSC and PNCs to meet the VN survivability requirement. VN survivability requirement and its policy need to be translated into multi-domain and multi-layer network protection and restoration scenarios across different controller types. After an E2E path is setup successfully, the MDSC has a unique role to enforce policy-based flexible VN survivability requirement by coordinating all PNC domains.

As seen in Figure A.5, multi-layer (i.e., packet/optical) coordination is a subset of this E2E protection/restoration control operation. The MDSC has a role to play in determining an optimal protection/restoration level based on the customer's VN survivability requirement. For instance, the MDSC needs to interface the PNC for packet core as well as the PNC for optical core and enforce protection/restoration policy as part of the E2E protection/restoration. Neither the PNC for packet core nor the PNC for optical core is in a position to be aware of the E2E path and its protection/restoration situation. This role of the MDSC is unique for this reason. In some cases, the MDSC will have to determine and enforce optical bypass to find a feasible reroute path upon packet core network failure which cannot be resolved the packet core network itself.

To coordinate this operation, the PNCs will need to update its domain level abstract topology upon resource changes due to a network failure or other factors. The MDSC will incorporate all these update to determine if an alternative E2E reroute path is necessary or not based on the changes reported from the PNCs. It will need to update the E2E abstract topology and the affected CN's VN topology in real-time. This refers to dynamic synchronization of topology from Physical topology to abstract topology to VN topology.

MDSC will also need to perform the path restoration signaling to the affected PNCs whenever necessary.