CCAMP Working Group
Internet Draft
Intended status: Informational
Expires: January 2017

Italo Busi
Huawei
Sergio Belotti
Nokia
Victor Lopez
Oscar Gonzalez de Dios
Telefonica
Anurag Sharma
Infinera
Yan Shi
China Unicom
Ricard Vilalta
CTTC
Karthik Sethuraman
NEC

July 7, 2016

Path Computation API
draft-busibel-ccamp-path-computation-api-00.txt

Status of this Memo

Abstract

There are scenarios, typically in a hierarchical SDN context, in which an orchestrator may not have detailed information to be able to perform an end-to-end path computation and would need to request lower layer/domain controllers to calculate some (partial) feasible paths.

Multiple protocol solutions can be used for communication between different controller hierarchical levels. This document assumes that the controllers are communicating using YANG-based Application Programming Interface (APIs).

This document describes some use cases for an Application Programming Interface for path computation. A related yang model will be proposed in a next version or in another document.

Table of Contents

1. Introduction

There are scenarios, typically in a hierarchical SDN context, in which an orchestrator may not have detailed information to be able to perform an end-to-end path computation and would need to request lower layer/domain controllers to calculate some (partial) feasible paths.

Multiple protocol solutions can be used for communication between different controller hierarchical levels. This document assumes that the controllers are communicating using YANG-based Application Programming Interface (APIs).

Path Computation Elements, Controllers and Orchestrators perform their operations based on Traffic Engineering Databases (TED). Such TEDs can be described, in a technology agnostic way, with the YANG Data Model for TE Topologies [TE-TOPO]. Furthermore, the technology specific details of the TED are modeled in the augmented TE topology models (e.g. [L1-TOPO] for Layer-1 ODU technologies).

The availability of such topology models allows providing the TED via Netconf or Restconf API. Furthermore, it enables that a PCE/Controller performs the necessary abstractions or modifications and offer this customized topology to another PCE/Controller or high level orchestrator.

The tunnels that can be provided over the networks described with the topology models can be also set-up, deleted and modified via Netconf or Restconf API using the TE-Tunnel Yang model [TE-TUNNEL].

This document describes some use cases where a path computation function, also using Netconf or Restconf API, can be needed. A related yang model will be proposed in a next version or in another document.

2. Use Cases

This document presents different use cases, where an API for path computation is required. The presented uses cases have been grouped, depending on the different underlying topologies: a) IP-Optical integration; b) Multi-domain Optical Networks; and c) Data center interconnections.

2.1. IP-Optical integration

In these use cases, there is an Optical domain which is used to provide connectivity between IP routers which are connected with the Optical domains using access links (see Figure 1).

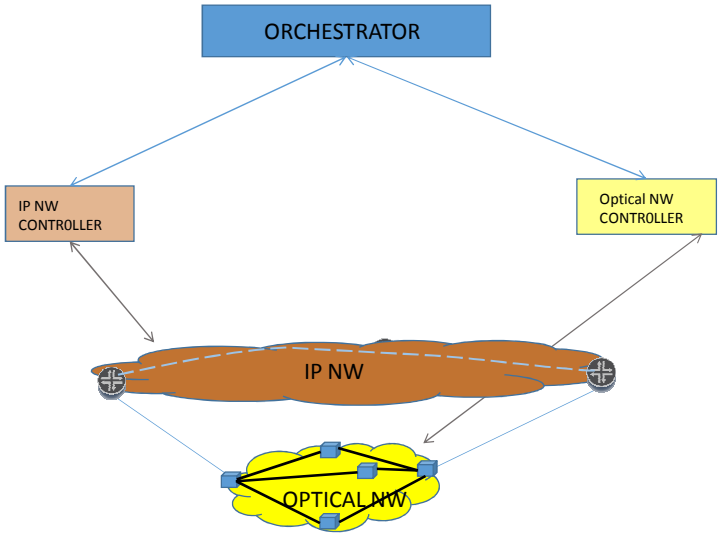

Figure 1- IP+Optical Use Cases

It is assumed that the Optical domain controller provides to the orchestrator an abstracted view of the Optical network. A possible abstraction shall be representing the optical domain as one "virtual node" with "virtual ports" connected to the access links.

The path computation request helps the orchestrator to know which are the real connections that can be provided at the optical domain.

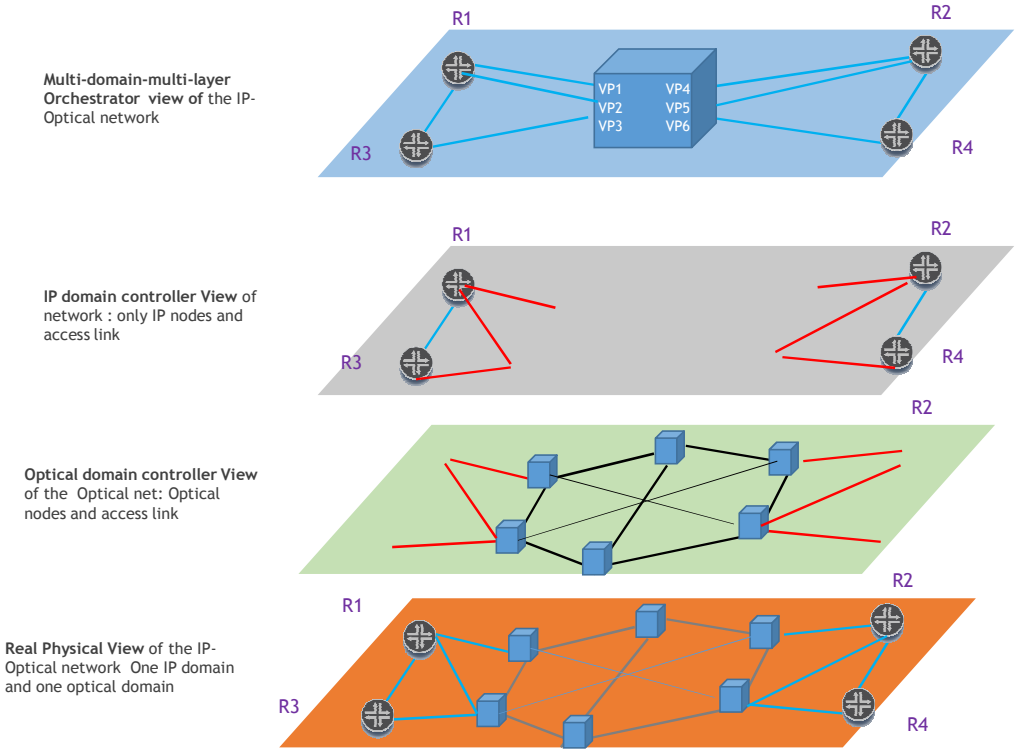

Figure 2- IP+Optical Topology Abstraction

2.1.1. Inter-layer path computation

In this use case the orchestrator needs to setup an optimal path between two IP routers R1 and R2.

As depicted in Figure 2, the Orchestrator has only an "abstracted view" of the physical network, and it does not know the feasibility or the cost of the possible optical paths (e.g., VP1-VP4 and VP2-VP5), which depend from the current status of the physical resources within the optical network and on vendor-specific optical attributes.

However, the orchestrator can ask the underlying Optical domain controller to compute a set of potential optimal paths, taking into account optical constraints Then, based on its own constraints, policy and knowledge (e.g. cost of the access links), it can choose

which one of these potential paths to use to setup the optimal e2e path crossing optical network.

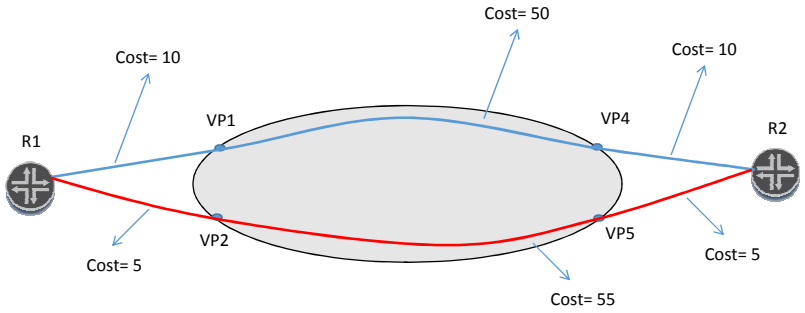

Figure 3- IP+Optical Path Computation Example

For example, in Figure 3, the Orchestrator can request the Optical domain controller to compute the paths between VP1-VP4 and VP2-VP5 and then decide to setup the optimal end-to-end path which passes through the VP2-VP5 Optical path even this is not the optimal path from the Optical domain perspective.

An alternative approach could be to have the Optical domain controller making the information shown in Figure 3 available to the Orchestrator.

One possibility, under discussion within the TEAS WG, is to provide a "detailed connectivity matrix" which extends the "connectivity matrix" defined in [RFC7446] and describes not only the valid inbound-outbound TE link switching combinations, but also specifies a vector of various costs (in terms of delay, OSNR, intra-node SRLGs and summary TE metrics) a potential TE path associated with the connectivity matrix entry.

The information provided by the “detailed abstract connectivity matrix” would be equivalent to the information that should be provided by “virtual link model” as defined in [TE-INTERCONNECT].

In this case, the Path Computation Element (PCE) within the Orchestrator could use this information to calculate by its own the optimal path between routers R1 and R2, without requesting any additional information to the Optical Domain Controller.

However, there is a tradeoff between the accuracy (i.e., providing "all" the information that might be needed by the Orchestrator's PCE) and scalability to be considered when designing the amount of information to provide within the "detailed abstract connectivity matrix".

Figure 4 below shows another example, similar to the one in Figure 3, but where there are two possible Optical paths between VP1 and VP4 with different properties (e.g., available bandwidth and cost).

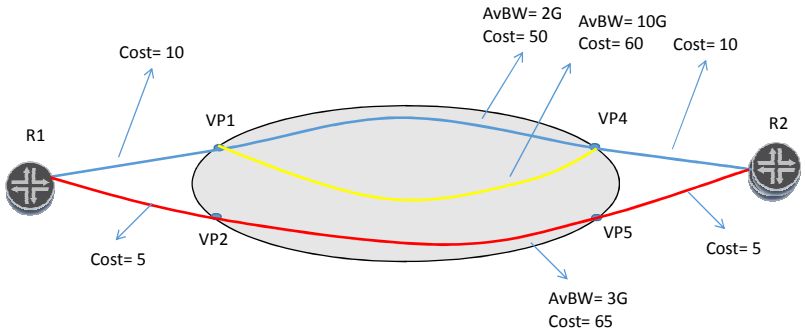

Figure 4- IP+Optical Path Computation Example with multiple choices

Reporting all the information, as in Figure 4, using the "detailed abstract connectivity matrix" is quite challenging from a scalability perspective since the amount of this information is not just based on number of end points (which would scale as N-square), but also on many other parameters, including client rate, user constraints / policies for the service, e.g. max latency < N ms, max cost, etc., exclusion policies to route around busy links, min OSNR margin, max preFEC BER etc. All these constraints could be different based on connectivity requirements.

It is also worth noting that the "connectivity matrix" has been originally defined in WSON, [RFC7446] to report the connectivity constrains of a physical node within the WDM network: the information it contains is pretty "static" and therefore, once taken and stored in the TE data base, it can be always being considered valid and up-to-date in path computation request.

Using the "connectivity matrix" with an abstract node to abstract the information regarding the connectivity constraints of an Optical domain, would make this information more "dynamic" since the

connectivity constraints of an Optical domain can change over time because some optical paths that are feasible at a given time may become unfeasible at a later time when e.g., another optical path is established. The information in the "detailed abstract connectivity matrix" is even more dynamic since the establishment of another optical path may change some of the parameters (e.g., delay or available bandwidth) in the "detailed abstract connectivity matrix" while not changing the feasibility of the path.

“Connectivity matrix” is sometimes confused with optical reach table that contain multiple (e.g. k-shortest) regen-free reachable paths for every A-Z node combination in the network. Optical reach tables can be calculated offline, utilizing vendor optical design and planning tools,and periodically uploaded to the Controller: these optical path reach tables are fairly static. However, to get the connectivity matrix, between any two sites, either a regen free path can be used, if one is available, or multiple regen free paths are concatenated to get from src to dest, which can be a very large combination. Additionally, when the optical path within optical domain needs to be computed, it can result in different paths based on input objective, constraints, and network conditions. In summary, even though “optical reachability table” is fairly static, which regen free paths to build the connectivity matrix between any source and destination is very dynamic, and is done using very sophisticated routing algorithms.

There is therefore the need to keep the information in the "connectivity matrix" updated which means that there another tradeoff between the accuracy (i.e., providing "all" the information that might be needed by the Orchestrator’s PCE) and having up-to-date information. The more the information is provided and the longer it takes to keep it up-to-date which increases the likelihood that the Orchestrator’s PCE computes paths using not updated information.

It seems therefore quite challenging to have a "detailed abstract connectivity matrix" that provides accurate, scalable and updated information to allow the Orchestrator’s PCE to take optimal decisions by its own.

If the information in the "detailed abstract connectivity matrix” is not complete/accurate, we can have the following drawbacks considering for example the case in Figure 4:

- If only the VP1-VP4 path with available bandwidth of 2 Gb/s and cost 50 is reported, the Orchestrator's PCE will fail to compute a 5 Gb/s path between routers R1 and R2, although this would be feasible;

- If only the VP1-VP4 path with available bandwidth of 10 Gb/s and cost 60 is reported, the Orchestrator's PCE will compute, as optimal, the 1 Gb/s path between R1 and R2 going through the VP2-VP5 path within the Optical domain while the optimal path would actually be the one going thought the VP1-VP4 sub-path (with cost 50) within the Optical domain.

Instead, using the approach proposed in this document, the Orchestrator, when it needs to setup an end-to-end path, it can request the Optical domain controller to compute a set of optimal paths (e.g., for VP1-VP4 and VP2-VP5) and take decisions based on the information received:

- When setting up a 5 Gb/s path between routers R1 and R2, the Optical domain controller may report only the VP1-VP4 path as the only feasible path: the Orchestrator can successfully setup the end-to-end path passing though this Optical path;

- When setting up a 1 Gb/s path between routers R1 and R2, the Optical domain controller (knowing that the path requires only 1 Gb/s) can report both the VP1-VP4 path, with cost 50, and the VP2-VP5 path, with cost 65. The Orchestrator can then compute the optimal path which is passing thought the VP1-VP4 sub-path (with cost 50) within the Optical domain.

Considering the dynamicity of the connectivity constraints of an Optical domain, it is possible that a path computed by the Optical domain controller when requested by the Orchestrator is no longer valid when the Orchestrator requests it to be setup up.

It is worth noting that with the approach proposed in this document, the likelihood for this issue to happen can be quite small since the time window between the path computation request and the path setup request should be quite short (especially if compared with the time that would be needed to update the information of a very detailed abstract connectivity matrix).

If this risk is still not acceptable, the Orchestrator may also optionally request the Optical domain controller not only to compute the path but also to keep track of its resources (e.g., these

resources can be reserved to avoid being used by any other connection). In this case, some mechanism (e.g., a timeout) needs to be defined to avoid having stranded resources within the Optical domain.

These issues and solutions can be fine-tuned during the design of the Path Computation API.

2.1.2. Route Diverse IP Services

This is for further study.

2.2. Multi-domain Optical Networks

In this use case there are two optical domains which are interconnected together by multiple inter-domains links.

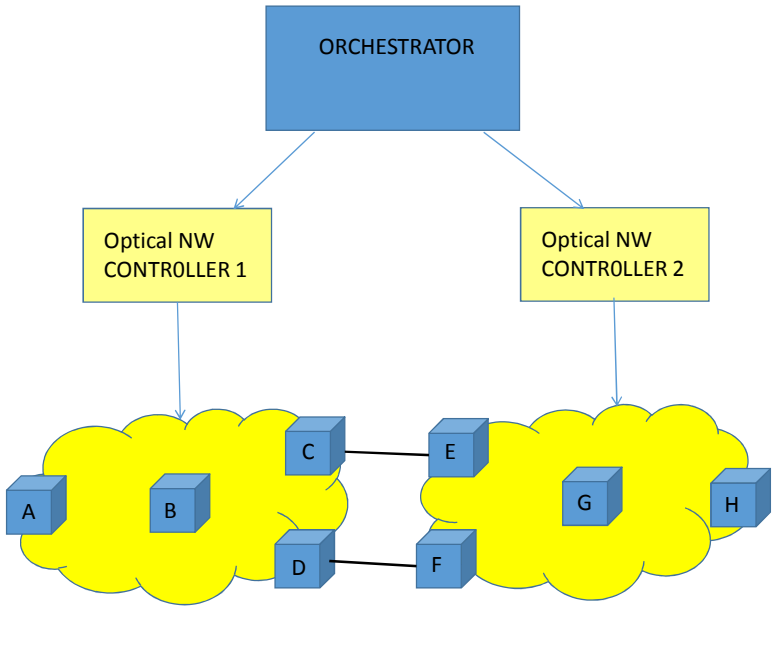

Figure 5 Multi-domain multi-link interconnection

In order to setup an end-to-end multi-domain Optical path (e.g., between nodes A and H), the orchestrator needs to know the

feasibility or the cost of the possible optical paths within the two optical domains, which depend from the current status of the physical resources within each optical network and on vendor-specific optical attributes (which may be different in the two domains if they are provided by different vendors).

There is a trade-off between having the Orchestrator's PCE being able to take path computation decisions by its own versus having the Orchestrator being able to ask the Domain Controllers to provide a set of feasible optimal optical paths.

Orchestrator could want to select/optimize end-to-end path based on abstract topology information provided by the domain controllers. For example:

- Need to compute a path between A and H
- That path can go through inter-domain link C-E or through inter-domain link D-F
- Orchestrator's PCE, based on its own information, can compute the optimal multi-domain path being A-B-C-E-G-H
- But, during path setup, the domain controller may find out that A-B-C is not optically feasible, while only the path A-B-D is feasible
- So what the hierarchical controller computed is not good and need to re-start the path computation from scratch

As discussed in section 3.1, providing more extensive abstract information from the Optical domain controllers to the multi-domain Orchestator may lead to scalability problems.

Alternatively the Orchestrator can request the Optical domain controllers to compute a set of optimal paths and take decisions based on the information received. For example:

- Need to compute a path between A and H
- The Orchestrator asks Optical domain controllers to provide set of paths between A-C, A-D, E-H and F-H
- Optical domain controllers return a set of feasible paths with the associated costs: the path A-C would not be part of this set

- The Orchestrator will select the path A-B-D-F-G-H since it is the only feasible path and then request the Optical domain controllers to setup the A-B-D and F-G-H paths

- If there are multiple feasible paths, the Orchestrator can select the optimal path knowing the cost of the intra-domain paths (provided by the Optical domain controllers) and the cost of the inter-domain links (known by the Orchestrator)

In a sense this is similar to the problem of routing and wavelength assignment within an Optical domain. It is possible to do first routing (step 1) and then wavelength assignment (step 2), but the chances of ending up with a good path is low. Alternatively, it is possible to do combined routing and wavelength assignment, which is known to be a more optimal and effective way for Optical path setup. Similarly, it is possible to first compute an abstract end-to-end path within the multi-domain Orchestrator (step 1) and then compute an intra-domain path within each Optical domain (step 2), but there are more chances not to find a path or to get a suboptimal path that performing per-domain path computation and then stitch them.

The approach to request each Optical domain controllers to compute a set of optimal paths and take decisions based on the information received may still have some scalability issues when the number of Optical domains is quite big (e.g. 20).

In this case, it would be worthwhile combining the two approaches and use the abstract topology information provided by the domain controllers to limit the number of potential optimal end-to-end paths and then the Path Computation to decide what is the optimal path within this limited set.

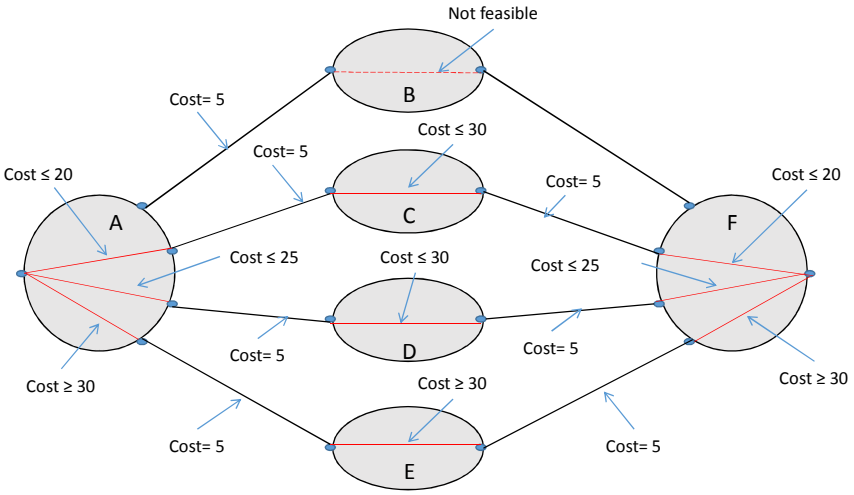

Figure 6 – Multi-domain with many domains (Topology information)

An example can be described considering multi-domain abstract topology shown in Figure 6. In this example an end-to-end Optical path between domains A and F needs to be setup. The transit domain should be selected between domains B, C, D and E.

The actual cost of each intra-domain path is not known a priori from the abstract topology information. The Orchestrator only knows the feasibility of some intra-domain paths and some upper-bound and/or lower-bound cost information. With this information, together with the cost of inter-domain links, the Orchestrator can decide that:

- Domain B cannot be selected as the path connecting domains A and E is not feasible;

- Domain E cannot be selected as a transit domain since it is know from the abstract topology information provided by domain controllers that the cost of the multi-domain path A-E-F (which is 100, in the best case) will be always be higher than the cost of the multi-domain paths A-D-F (which is 90, in the worst case) and A-E-F (which is 80, in the worst case)

Therefore, the Orchestrator can decide by its own that the optimal multi-domain path could be either A-D-F or A-E-F.

The Orchestrator can therefore request only the Optical domain controllers A, D, E and F to provide a set of optimal paths.

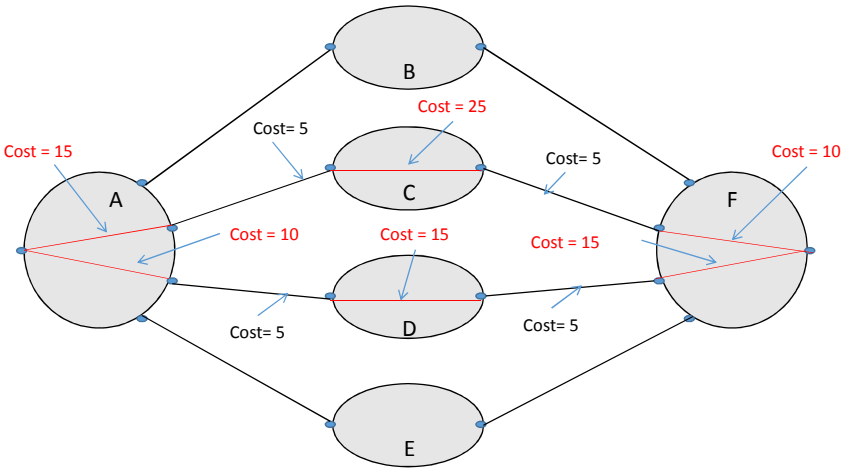

Figure 7- Multi-domain with many domains (Path Computation information)

Based on these requests, the Orchestrator can know the actual cost of each intra-domain paths which belongs to potential optimal end-to-end paths, as shown in Figure 7, and then compute the optimal end-to-end path (e.g., A-D-F, having total cost of 50, instead of A-C-F having a total cost of 70).

2.3. Data center interconnections

In these use case, there is an Optical domain which is used to provide connectivity between data centers which are connected with the Optical domains using access links.

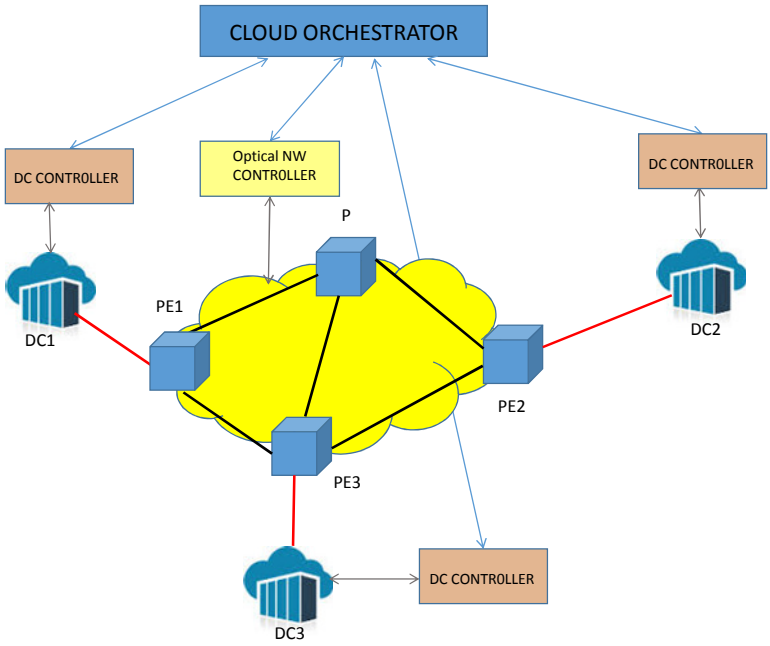

Figure 8- Data Center Interconnection Use Case

In this use case, a virtual machine within Data Center 1 (DC1) needs to transfer data to another virtual machine that can reside either in DC2 or in DC3.

The optimal decision depends both on the cost of the optical path (DC1-DC2 or DC1-DC3) and of the computing power (data center resources) within DC2 or DC3.

The Cloud Orchestrator may not be able to make this decision because it has only an abstract view of the optical network (as in use case in 3.1).

The cloud orchestrator can request to the Optical domain controller to compute the cost of the possible optical paths (e.g., DC1-DC2 and DC1-DC3) and to the DC controller to compute the cost of the computing power (DC resources) within DC2 and DC3 and then it can take the decision about the optimal solution based on this information and its policy.

3. Security Considerations

This is for further study

4. IANA Considerations

This document requires no IANA actions.

5. References

5.1. Normative References

[RFC7446] Lee, Y. et al., "Routing and Wavelength Assignment Information Model for Wavelength Switched Optical Networks", RFC 7446, February 2015.

5.2. Informative References

[TE-TOPO] Liu, X. et al., "YANG Data Model for TE Topologies", draft-ietf-teas-yang-te-topo, work in progress.

[L1-TOPO] Zhang, X. et al., "A YANG Data Model for Layer 1 (ODU) Network Topology", draft-zhang-ccamp-l1-topo-yang, work in progress.

[TE-TUNNEL] Xhang, X. et al., "A YANG Data Model for Traffic Engineering Tunnels and Interfaces", draft-ietf-teas-yang-te, work in progress.

[TE-INTERCONNECT] Farrel, A. et al., "Problem Statement and Architecture for Information Exchange Between Interconnected Traffic Engineered Networks", draft-ietf-teas-interconnected-te-info-exchange, work in progress.

6. Acknowledgments

The authors would like to thank Igor Bryskin and Xian Zhang for participating in discussions and providing valuable insights.

This document was prepared using 2-Word-v2.0.template.dot.

Contributors

Dieter Beller
Nokia
Email: dieter.beller@nokia.com

Authors' Addresses

Italo Busi
Huawei
Email: italo.busi@huawei.com

Sergio Belotti
Nokia
Email: sergio.belotti@nokia.com

Victor Lopez
Telefonica
Email: victor.lopezalvarez@telefonica.com

Oscar Gonzalez de Dios
Telefonica
Email: oscar.gonzalezdedios@telefonica.com

Anurag Sharma
Infinera
Email: AnSharma@infinera.com

Yan Shi
China Unicom
Email: shiyan49@chinaunicom.cn

Ricard Vilalta
CTTC
Email: ricard.vilalta@cttc.es

Karthik Sethuraman
NEC
Email: karthik.sethuraman@necam.com