Internet Engineering Task Force
Internet-Draft
Intended status: Informational
Expires: June 3, 2017

A. Veitch, Ed.
Netcracker, Inc.

November 30, 2016

# Use Cases and Analysis on Integrated NFV and Network Optimization
draft-veitch-nfvrg-nfv-nw-optimization-00

## Abstract

This is a review of issues related to the optimized deployment of network services, aka service function chains, composed of virtual and physical network functions, where these functions may be instantiated in multiple distributed data centers.  Criteria for optimization are introduced, and use cases are described and expanded, in order to establish the need for coordinated computation of NF deployment sites and the connectivity among them.  Some methods for addressing optimization pain points are described, and potential new requirements of the PCE and PCEP are discussed.

One point to make clear is that the exploration of specific algorithms for NF/VNF placement and path computation is outside the scope of this draft.  We are not looking at computational or optimal greedy search algorithms.  The goals for this draft are 1) to provide justification through use cases for more tightly integrating path and placement computation algorithms, and 2) looking at how some of the optimization requirements might be addressed, mostly through pre-computation and caching, and how these might affect management and orchestration functions and the protocols (e.g.  PCEP) that are used.

## Status of This Memo

Copyright Notice

Table of Contents

## 1.  Introduction

With the advent of Network Function Virtualization (NFV), network services, consisting of an ordered set of network functions (NFs), may be composed of physical (PNF) and virtual network functions (VNF).  These VNF will execute in one or more virtual machines or containers operating on standard high performance servers.  The set of network functions for a given network service might be instantiated within a single site or the network service might be multi-site.

This document is a brief examination of issues related to the optimized deployment and operations of network services, and in particular, multi-site network services.  It includes a review of use cases that reveal possible limitations in current NFV and SDN management architectures, as defined in some standards and open source projects, and related communications protocols.

The use cases described include the virtualization of support for Content Delivery Networks (CDN), Internet of Things, video narrowcasting, and high performance computing.  In this analysis, an important consideration is the shared infrastructure among a variety of applications and services.  Sharing the infrastructure has the advantage of increasing overall utilization of the infrastructure vs. support for dedicated solutions for each service type.  One can imagine disjoint networks for each of the services being implemented, with the result being a very high total cost to the provider and consumer.

Another advantage is that the resource utilization profiles for each of these services will differ.  High performance computing might require a good deal of computational power and memory, which translates into requiring many CPU, virtualized CDN require a large amount of storage, and the main requirements for video services might be low latency and jitter in the delivery of its traffic from its source to its consumers.

An understanding of the expected resource usage for different services and network functions will allow intelligent placement of these services, not just to provide the user a high quality of experience (QoE), but also provide the operator with a lower CAPEX through smarter utilization of the resources.

One of the goals in reviewing these use cases is to understand the common information that is necessary for an operator to effectively provide optimal solutions for the various services to be supported, and their disparate requirements.

## 2. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

## 3. Definitions

Note: This document includes concepts and terminology used in both the IETF and in the ETSI NFV ISG. A number of concepts are shared or have very similar counterparts between the groups. Here are some of the key terms used and their definitions.

- o Management and Orchestration (MANO) - Describes the architecture framework to manage NFVI and orchestrate the allocation of resources needed by the NSs and VNFs.

- o Network Function (NF) - A functional block within a network infrastructure, that has well-defined external interfaces and a well-defined functional behavior.

- o Virtual Network Function (VNF) - Implementation of an NF that can be deployed on a Network Function Virtualisation Infrastructure (NFVI). IETF corollary is the Service Function or SF in SFC.

- o NFV Infrastructure (NFVI) - The NFV-Infrastructure is the totality of all hardware and software components which build up the environment in which VNFs are deployed. It may span across several sites, e.g. where data centers operate.

- o NFVI-PoP - Where a Network Function is or could be deployed as Virtual Network Function (VNF)e

- o Network Function Virtualization Orchestrator (NFVO) - functional block that manages the Network Service (NS)lifecycle and coordinates the management of NS lifecycle, VNF life cycle and

NFVI resources to ensure an optimized allocation of the necessary resources and connectivity

- o Network service (NS) - A composition of network functions and defined by its functional and behavioral specification.

- o VNF Forwarding Graph (VNFFG) - A NF forwarding graph where at least one node is a VNF.

- o Multi-site network service - A service that has component network functions operating in infrastructure located in separate geographical sites. These sites may consist of a combination of customer sites and operator data centers. Service components are therefore connected across wide area networks (WANs.)

## 4. Optimization Criteria

One question to be asked, when we mention optimization, is, what is it that we want or need to optimize when a network service or services are to be deployed? And the answer is that there are a number of criteria we would like to see maximized or minimized, some of which lead in opposite directions.

- o Maximize the likelihood a requested network service will be honored and instantiated, and not rejected due to a lack of available resources to meet the service's SLAs.

  For a multi-site service, it is important that the site of the network functions and the connectivity between them be selected to ensure any SLA requirements are met. The search for a solution should not fail when adequate resources exist to support the service request.

- o Maximize the health of a network service and minimize the likelihood that service might fail to meet its service level agreements, or SLAs, during its lifetime.

  Each network service deployed must meet an SLA, with a penalty likely imposed should the service fail to do so. These SLAs will include metrics such as Availability, throughput or bandwidth, loss, latency, maximum outage time, mean time between failures, etc. Goals include minimizing the risk of any single service failing to meet its SLAs, as well as minimizing the overall number of services that fail to meet their SLAs.

- o Minimize the time it takes to respond to a network service request with an instantiation of that service. Alternatively, a request may ask only for the set of resources to support a network service, but not to actually deploy the services. The response time for this request should also be minimized.

- o Maximize the utilization of the network and compute infrastructure, or NFVI, to achieve OPEX and CAPEX savings.

  One promise of NFV is the reduction of OPEX and CAPEX. In other words, with greater flexibility and finer grained control, the better a management system can utilize the NFVI. CAPEX is clearly lower when less total NFVI is used, and OPEX should be too. For example, excess hardware may be hibernated when not needed, saving energy costs and management costs.

## 5. Challenge of Network Service (or Service Function Chain) Optimization Across Multiple Sites

When deploying a network service, it is important to place the component network functions (PNF/VNF) in sites to deliver an optimized network service, as based on the criteria listed above. Deciding how and where to deploy the VNFs is made more complex when the network service is to be deployed across multiple sites.

A network service descriptor includes a set of NFs and links that represent connections between pairs of the network functions. Associated with each link might be a set of criteria, e.g. latency, bandwidth, etc., that an instance of the service must meet. When NFs are to be placed in different sites and separated by one or more networks, both the sites of the NFs and the connectivity paths between the pairs must be selected to meet the NF and connectivity requirements defined for the network service and its components. This means that choosing the placement of the VNFs and selecting their connectivity are interdependent activities.

## 6. Review of Current NFV / SDN Management Solutions

Today a number of management and orchestration architectures for NFV based services have been defined and are described by standards organizations, open source projects, as well as some service providers. The ETSI NFV ISG described the original NFV management architecture: The Management and Orchestration of NFV based network services, commonly referred to as MANO, was defined in phase 1 of the ISG work program.

Since then, other standards bodies have addressed the architecture, and a number of software projects undertaken to develop management systems, based on, or strongly influenced, by the ETSI MANO architecture. These include, but are not limited to:

- Lifecycle Service Orchestration (LSO) from the Metro Ethernet Forum (MEF)

- Open Source MANO (OSM), also sponsored by ETSI

- Open Platform NFV (OPNFV) from Linux Foundation

- SONATA

- Openstack, including Tacker, and networking-sfc

- Open-O

- Enhanced Control, Orchestration, Management and Policy (ECOMP) from ATT

- TMForum architecture

These architectures generally follow two paradigms. Each includes an End-2-end (E2E) orchestrator that is responsible for overall orchestration of network services, including the VNFs, PNFs and network connectivity. This E2E orchestrator then communicates with an NFVO, an orchestrator for the network services based on the VNFs. Where they differ is in how the network controller, often an SDN controller, fits. In most of these, the E2E controller communicates directly to both the NFVO and the network controller. In others the NFVO communicates to the network controller.

```
+----------------------------------------------------------------------+
|                           E2E Orchestration                          |
+----------+------------------------+------------------------+---------+
           |                        |                        |
           |                        |                        |
+----------+---------+   +----------+---------+    +---------+---------+
| WAN SDN Controller |   | DC SDN Controller  |    |       NFVO        |
+----------+---------+   +----------+---------+    +---------+---------+
           |                        |                        |
           |                        |              +---------+---------+
           |                        |     ---------+       VIM         |
           |                        |     |        +---------+---------+
           |                        |     |                  |
 ----------+--------      ----------+-----+--      +---------+---------+
(   WAN Networks    )    (    DC Network     )     |       VNFs        |
 -------------------      -------------------      +-------------------+


+----------------------------------------------------------------------+
|                           E2E Orchestration                          |
+-----------------------------------+----------------------------------+
                                    |
                                    |
                         +----------+---------+
           +-------------+         NFVO       +--------------+
           |             +----------+---------+              |
           |                        |                        |
           |                        |                        |
+----------+---------+   +----------+---------+    +---------+---------+
| WAN SDN Controller |   | DC SDN Controller  +----+       VIM         |
+----------+---------+   +----------+---------+    +---------+---------+
           |                        |                        |
           |                        |                        |
 ----------+--------      ----------+--------      +---------+---------+
(   WAN Networks    )    (    DC Network     )     |       VNFs        |
 -------------------      -------------------      +-------------------+
```

Two typical paradigms of network service orchestration seen in open source and SDO architectures.

Figure 1

The key point is that in both paradigms the NF deployment decisions and connectivity decisions are controlled by separate functions. As described in literature to date, the functionality described for each, and the information to be exchanged, is not adequate to avoid potential inefficiencies and unnecessary network service deployment failures, due to the real interdependence these functions have upon each other.

Of course, these architectures are currently undergoing active development through open and iterative design processes.  Therefore, the state of these designs is very fluid, with new capabilities continuously evolving.  Therefore, the description provided here represents only a moment in time.  This document is intended to provide insight into possible limitations that exist in computing deployment solutions and to recommend updates to functional block capabilities and communications (perhaps leading to protocol updates), that will lead to improving these computations.  It is expected that as these projects evolve, this draft too will evolve, further clarifying roles, communications, and needed extensions.

## 7.  Use Cases

### 7.1.  Introduction

The following are a set of network service examples that demonstrate the value of coordinated or unified compute and network deployment and configuration planning.

### 7.2.  Virtual Content Delivery Network (vCDN) - Dynamic (Flash) Delivery

The following are a set of network service examples that demonstrate the value of coordinated or unified compute and network deployment and configuration planning.

ETSI and other groups have identified virtual CDN (vCDN) as an application well suited to operate as an NFV network service.  As 5G is adopted and bandwidth greedy applications such as UHD video proliferate, mobile data will have a huge impact as a source of network traffic.  CDNs have a proven track record as an effective way to provide high quality and low latency delivery of content to users, while simultaneously limiting the overall utilization of network bandwidth.  Operators need to continue to provide a high quality service while limiting provisioning unnecessary infrastructure. Virtual CDNs are seen as an excellent way to address these new pressures.

There are some obvious advantages to a virtualized solution.  The costs will shift from CAPEX to OPEX as the vCDN can operate on standard high capacity servers.  More importantly, a software-based solution enables rapid configuration of the vCDN, faster upgrades of the software, and mobility of the vCDN vCaches.  With vCDN it is possible to dynamically and intelligently size the vCaches, and move them to sites where they provide the best value, based upon the current and anticipated near-term traffic volumes and patterns. More ephemeral demands for content can be more effectively managed.

The SONATA Use Cases and Requirements document (http://www.sonata-nfv.eu/content/d21-use-cases-and-requirements) categorizes two types of vCDN scenarios.

First is the more traditional distribution of popular content. Content originates at a content provider and it is distributed on vCaches located as needed for a large number of subscribers.

The second scenario is concerned with user-generated content and is much more dynamic. This is the one that is examined here. This is a 'flash' vCDN, one where the need for a CDN is due to a sudden burst of content, one with a limited expected lifetime. One example might be any large sporting event that many people attend. These attendees will likely take and post multiple videos to the Web with a social media application, and then they and their friends will view and download these videos multiple times. During the event there will a real need to support caching this content and making it available to consumers while limiting network impact.

```
                 +-----------+
              +--+ End Users |                     +-----------+
              |  +-----------+---------------------| End Users +---+
   +---+------+  ------/         WAN         \---  +-----------+   |
   |   Data   |-/                                   \---   +-------+--+
   |  Center  +-                                         \ |   Data   |
   | NFVI-PoP | \@                                        -+  Center  |
   +----------+    \@                                   @/ | NFVI-PoP |
      |               \@                             @/     +----------+
        -\               \@                      @/        /-
          --\               \@               @/         /--
             ---\              \@       @/         /---
                 ---+-------------\---/--------+----
                    |  Data Center (NFVI-PoP)  |
                    | +---------+  +---------+ |
                    | | vCache  |  | vCache  | |
           ----     | +---------+  +---------+ |     -----
        --/         +---+-------------------+--+          \--
       (                |  Access Network   \                )
        --\            /                     |            /--
           --------\  |                      \   /--------
                    --/-----------------------\--
                    |                      +---------+-----+
          +----------+----+                | Media Content |
          | Media Content |                |    Creator    |
          |    Creator    |                +---------------+
          +---------------+
```

Topology of vCDN use case with a rapid short term need for content caching.

Figure 2

The challenge for this use is for the provider to be able to rapidly deploy and configure the a vCDN solution so that the users' QoE is high, both the sources and the consumers, while the resource utilization, mostly the WAN connectivity, is minimized.

In this use case the placement of the vCaches is driven by a number of factors, including the proximity to the users posting the videos, the proximity to users consuming the videos, the available network capacity among the data centers, reliability (availability) depending upon the service, the costs of the data center servers, and the cost of the network, if connectivity requires transit across a third party's network. This is not simply a matter of identifying the data centers and then connecting them. Candidate sites must be identified, connectivity options must be identified, and the combination of these must be evaluated in the context of the factors listed above, past and current performance measurements, and any provider or user policies.  Multiple deployment options may be considered, with one selected to provide an optimal solution.

As this will be a dynamic vCDN deployment, frequent monitoring of demand and delivery metrics is a major necessity, and redeployments and upgrades, or possible reductions when demand wanes, must be addressed with no unnecessary delay.

## 7.3.  Video Narrowcasting Use Case

Video narrowcasting is a targeted delivery of video (and audio) content to multiple consumers, perhaps from the tens to low thousands, and delivery is generally expected to be broadcast quality.  This is not broadcast television, but might be used for distance training, or education, or internal communications within large organizations.  It is also a term used to describe the delivery of commercial television to a niche audience, e.g. from a cable service provider.  This differs from popular microcomputer based applications such as Skype or weChat, as high quality (high definition), high availability and very low latency are expected.

In some cases, there will be more than one video source; you can imagine a presentation where presenters are based in different sites, and in some of these cases, the video encodings might differ from the different sources.  In addition, lower quality video and audio might be available for users to send feedback to the primary sources.

A number of factors must be examined when choosing how to support this service.  These include, the number of sources, their sites, their encoding capabilities, and the bandwidth available over which each can send.  Also to be considered are the receivers, their number and site, their bandwidth capacity over which to receive the videos, and their codecs.  How many incoming video streams can the receiver process at once.

Today there are commercial hardware and software products and packages to provide this capability.  These, however, can be costly and are not very flexible.

You can imagine a number of situations.  The simplest would be if all senders have the same video codec, all receiver codecs can decode as many incoming video streams as needed, and all senders and receivers have more than enough bandwidth for sending or receiving, and high usage of WAN links is not a problem.  When this is true there is very little service management needed.  Unfortunately, this is unlikely.

```
                 +-----------+
              +--+ End Users |                     +-----------+
              |  +-----------+---------------------| End Users +---+
   +---+------+  ------/         WAN         \---  +-----------+   |
   |   Data   |-/                                   \---   +-------+--+
   |  Center  +-                                         \ |   Data   |
   | NFVI-PoP | \@@                                       -+  Center  |
   +----------+    \@                                   @/ | NFVI-PoP |
      |               \@                            @/     +----------+
        -\               \@                    @/       /-
          --\               \@             @/        /--
             ---\              \@     @@/         /---
                 +----------------\---/-------------+
                 |     Data Center (NFVI-PoP)       |
                 | +---------+ +-------+ +--------+ |
                 | |  video  | | video | | video  | |
                 | | process | |  mgmt | | server | |
           ----  | +---------+ +-------+ +--------+ | ----
        --/      +-----+--------------------+-------+      \--
       (               |  Access Network    \                )
        --\            /                     |            /--
           --------\  |                      \   /--------
                    --/-----------------------\--
                    |                      +---------+-----+
          +----------+----+                |     Video     |
          |     Video     |                |     Source    |
          |     Source    |                +---------------+
          +---------------+
```

Topology of video narrowcasting>.

Figure 3

More likely is that some of the following are true and must be addressed.

There are multiple senders with different coding capabilities, or different bandwidths available over which to send.  In addition, some of the receivers do not have the video codecs to decode some of the sender encodings, or lack the bandwidth over which to receive the packets of the high definition video.

When this is true, functions must be installed into the network to manage these issues properly.  Video processing can be added to mix video of different sources, if necessary for some consumers to see multiple senders.  Consumers may even remotely control this processing, on demand.  Other processing may be needed to compress the video, with the effect of lowering its quality, but making it possible for consumers with limited bandwidth (e.g. mobile) to see it. These and many other factors will drive selecting where to place these functions, connect them to one another, and how to forward these to the end consumers (e.g. what IP multicast groups, etc.), all to provide the best QoE while limiting resource consumption.

The evaluation of these factors may quickly become complex, and may be continuous as consumers join and leave. The placement of the VNFs will depend greatly on network connectivity, capacity and latency, as well as the potential to provide connections directly over electrical or optical, vs. multi-hop packet.

## 7.4. Real-Time Telemetry Use Case - Internet of Things

The Internet of Things (IoT) is growing exponentially and will be a huge source of information to be communicated across the Intenet in the coming years. This use case in concerned with a large number of sensors being activated over a short amount of time. One example of this could include a large sporting event, e.g. the Boston Marathon, with multiple thousands of competitors wearing health-monitoring applications that communicate vital signs as telemetry over the Internet. Or this could be relevant for a natural disaster (or man-made disaster.) Sensors might be deployed, or simply activated to report much more data, because of an incoming hurricane, or a volcanic eruption, tornado, forest fire, or tsunami and infrastructure emergencies. This could necessitate the creation of a new IoT gateway and other associated components located nearer to the sensors and with the capacity to forward the data in real-time to its consumers.

This use case is quite similar to the flash vCDN use case described above. Here, instead of vCaches, we have IoT gateways. Instead of cameras taking videos to be uploaded and downloaded, we have sensors, to be processes and forwarded in real-time to the consumers. This is a case where low latency, and perhaps low jitter, is required, though high throughput less so. Again, these network requirements might impact the placement of the gateways and their connectivity.

A further complexity is introduced when we consider these sensors to be mobile. One example will be with self-driving cars, or even human driven cars, but where there are many more sensors. The vehicles will generate a great deal of data which must travel over the wireless access networks to the nodeBs, behind which will be IoT gateways to process the data and forward the relevant data in real time to other functions. These might automatically change driving routes, maximum speeds for some roads, etc.

In this scenario it seems likely that different areas will have different loads of network sensor data based upon the time of day. There will be low volumes during the day, with greater volumes during rush hour.  And these will migrate from city or industrial areas, to suburbs.  (Just as, no doubt, today the cell phone calls managed in different areas depend upon the time of day.)  Virtual IoT gateway deployments and the connectivity among the gateways and telemetry consumers will there for also be varied.  This is more predictable, but deployment options must still consider connectivity, network utilization, etc. when placing the vIoT gateways, etc.  There are also likely more dynamic mobile scenarios to be reviewed, e.g. travelers evacuating an area due to a hurricane, etc.

## 7.5.  High Performance Computing (HPC)

Note: A number of concepts here are taken from 'HPC-Aware VM Placement in Infrastructure Clouds', by Gupta et al.

High performance computing (HPC) in the cloud promises to expand the number of computing applications that can be supported and will lower costs over all.  However, HPC has a number of requirements that place significant demands upon the compute and networking infrastructure.

Included among these are the need to 1) periodically move huge volumes of data among distributed applications as well as 2) support multiple processes that are tightly coupled and require frequent communications and synchronizations.  In addition, for some large computations, there may be the need to rapidly expand to very large numbers of collocated VNFs for large computations with very strict synchronization time windows.

There is active research today for improving VM scheduling, hardware utilization, how best to partition clusters for different application types, and map functions to sites to maximize utilization.

When considering some of these demands, a few things are clear.  For communicating large volumes of data, it will be necessary to dedicate significant bandwidth at times and limit latency.  When supporting HPC as a service a provider must anticipate these sorts of needs, and ensure the infrastructure resources supporting other services are utilized such that the capacity to meet this sort of demand is possible without the need to migrate services and VNFs.

When the resource requirements of cooperating applications recommend locating the applications in separate data centers and the application's synchronized communications requirement demands connectivity of a high bandwidth circuit, the provider must ensure both the data center infrastructure and connectivity between the datacenters meet the service requirements.  In this case, while a circuit may not exist, the function computing the connectivity should be aware of the potential circuit, for situations such as supporting this network service.

## 8.  Orchestration of Mixed Applications and Network Services

The use cases listed above place a variety of disparate demands upon the data center (NFV-IPoP) and network infrastructure.  For each individual network service instance, a service deployment algorithm may compute the sites and connectivity for that specific instance. These computations are made in the isolation of the context of that specific service instance, perhaps over its lifecycle.

However, there is a need to consider and manage overall delivery and support of these individual services.  As has been described above, it is beneficial to deploy across the data centers a balanced distribution of network service components with different resource requirement profiles.  This will help to optimize the utilization of the data center and network infrastructure.  And in circumstances when some of the services are known to place specific large demands on infrastructure, for example, instantiating an ODU circuit to provide high bandwidth real-time communications, some agent must be aware of this sort of demand to ensure it is met.

This implies that there could be some sort of higher-level agent, one that works with the agents (e.g.  NFVO and SDN controller) computing compute and network deployments for individual requested service definitions.  This higher layer agent must ensure that resources are allocated properly so that those agents compute deployments that also honor the greater goals, e.g. the balanced distribution of heterogeneous services, resources ready for specific demands of some of the services, general reduced energy consumption and lower costs, etc.  These should lead to the optimization goals described at the beginning of this paper.

The exact communications that might be required to support this should be explored.  These could impact the controller NBI or the PCE pro tocol.

```
   +-----------------------------+
   |     E2E Orchestration       |
   |              +------------+ |
   |              |   General  | |
   +------+------ |Optimization|-+
          |       +-------+----+
          |     -/        |
          |   -/          |
   +------+--+--+   +-----+------+
   |            |   |  Network   |
   |    NFVO    |   | Controller |
   |            |   |(PCE / SDN) |
   +------------+   +------------+
```

Shows an agent that tracks usage to provide guidance for general optimization.

Figure 4

## 9. Generalized Architecture Options for Coordinated or Unified SDN/NFV Deployment Planning

To achieve the integrated NFV/SDN optimization goals requires a software system that computes the optimizations for compute and network.

One option would be to replace the separated agents, the NFVO and SDN controller, with a single unified agent that processes the network topology, configuration and utilization, as the PCE / SDN controller does today, and computes the VNF placement, as the NFVO does today. If we look at the E2E Orchestrator in Figure 1, one option is for the network SDN controller and the NFVO to send full topology and NFVI (data center topology, compute et al resources and existing use and utilization) information to the E2E Orchestrator, where it is the unified agent, computing the optimal deployment, computing the path and NFVI allocations, and then directing specific actions of the PCE / SDN agent and NFVO to deploy the updates.

Alternatively, the functionality of the NFVO and SDN Controller remain mostly consistent.  Each fills its current set of responsibilities, the PCE computing paths and the NFVO planning NFVI-PoP updates with the new VNF instances.  If this is done, then enhancements must be made so these agents coordinate and compute the best overall deployment.

## 10. Analysis of Integrated Network and Compute Optimization with a Simple Use Case

### 10.1. Introduction

Here is presented a simple network service, composed of two VNFs (SFs) across two data centers. This could be a vCPE service with a remote VNF. This simple case is used here to provide a basis on which to explore how network services (SFCs) may be orchestrated. Solutions for the simple case are examined iteratively, each new version presenting more advanced orchestration capabilities that address deficiencies identified in the previous version. The purpose here is not to examine any path computation optimization algorithms. It is intended to demonstrate how other methods, including pre-computation and/or memoization, might aid in meeting the optimization goals, and what this might mean for network topology modeling and protocol definitions.

Note: Many concepts in the TE Topology model currently being defined in the TEAS WG, can be considered relevant to this work. The intention of this section is to describe the general problem and approach. How this relates to the concepts and models defined in the TE Topology is explored in the next section of this document.

```
+------------------+       +-------+           +-------+
| Service Endpoint |       |       |           |       |
| (Cust. Premises) +-------+ VNF A +-----------+ VNF B |
|                  |       |       | 3GB req.  |       |
+------------------+       +-------+           +-------+
```

Shows an architecture.

Figure 5

### 10.2. Network Topology

The topology includes three data centers and a user enterprise site, where the a user service endpoint is located.

Two variations are used. In the first, tunnels are already established among the data centers. In the second, a tunnel is established specifically to support the link between the network functions.

```
                      +-------------+                    +-------------+
+-----------------+   |             |                    |             |
|Service Endpoint |   | Data Center |   --------  2 GB   | Data Center |
|(Cust. Premises) +--+ (NFVI-PoP)  +--(   WAN  )---------+  (NFVI-PoP) |
+-----------------+   |             |   ----+--- Available|             |
                      +-----|-------+       |            +------+------+
                                            |
                                            |
                                            |
                                            |            +------+------+
                                            |            |             |
                                            |   5 GB     | Data Center |
                                            +------------+ (NFVI-PoP)  |
                                               Available |             |
                                                         +-------------+
```

The network topology available for establishing connectivity for the network service.

Figure 6

## 10.3. Simple Orchestration

In this example the computation of the VNF deployment sites occurs first, followed by the connectivity solution. There is only one link property to be met, minimum bandwidth, and for the link between VNF A and VNF B, the minimum acceptable bandwidth of 2 GB . (Note: This is shown in the labels of the links in Figure 6.)

Here are the actions and events involved in planning the creation of a service instance.

o Orchestration software receives the network service request.

  The request includes descriptors, including resource requirements, for each VNF (SF) and the links between them. For the VNFs, this includes the number of needed containers (a VNF may be composed of multiple VNFCs), compute, memory, and storage. For the links this includes bandwidth, latency, jitter, and loss.

o Orchestrator identifies the data centers based on proximity to the user site, and the availability of servers with the resources to meet the VNF (SF) resource requirements and any relevant policies.

  At this point, the choice of data centers for the VNF has been finalized.

- o Orchestrator requests the network controller to establish connectivity for each virtual link connecting VNFs in the network service description.

  A path must be found between each pair of VNF, one which can provide the minimum bandwidth associated with the virtual link between those VNF.  NOTE: If no suitable path is found, then the network service deployment FAILS.

```
   +-----------------+  +-------------+                +-------------+
   |Service Endpoint |  | Data Center |                | Data Center |
   |(Cust. Premises) +--+ (NFVI-PoP)  |         3 GB   | (NFVI-PoP)  |
   +-----------------+  | +---------+ |  -----  Needed | +---------+ |
                        | |  VNF A  +---( WAN )----------+  VNF B  | |
                        | +---------+ |  -----  2GB    | +---------+ |
                        +-------------+         Avail. +-------------+
```

>Inadequate bandwidth for service with VNF placement computed in advance.

Figure 7

## 10.4.  Connectivity Information in VNF Placement Planning

An obvious deficiency of this process is in choosing VNF sites without considering the ability of the network to provide the necessary connectivity and bandwidth.  The solution is to update the orchestrator to consider connectivity as a factor when computing where the VNF should be placed.  Whatever the placement algorithm, it must ensure that VNFs, connected by a virtual link in a network service definition, have supporting network connectivity that will meet the service requirements, e.g. bandwidth, or latency, associated with that virtual link.

Compared to the previous solution, here the network service deployment algorithm will place a VNF in a datacenter only if network connectivity is acceptable.  If an orchestrator chooses to place a VNF in a datacenter, then the next VNF in the service path must be placed in a site where connectivity between the two meets the requirements in the virtual link description.

A simple backtracking algorithm, such as depth first search (DFS), will start at the front of the service and check on possible next hop data centers (including the 'current' one).  It will choose one based on proximity, available compute, storage, etc. resources, policies, etc.  It will then request of the PCE or SDN controller to compute a path to the candidate data center, passing the virtual link requirements, e.g. bandwidth, latency etc.  The PCE/controller should consider the topology, currently allocated bandwidth, traffic volumes and profile, hop count, encapsulations, etc. in computing paths. This will continue until either all options have been searched unsuccessfully, or until a successful set of data centers and paths between them is found. Of course, DFS will not be used, as there are much more efficient heuristic algorithms available. But the concept is the same regardless of the algortihm. Connectivity information is utilized when choosing VNF sites.

One problem with this solution is the need to have the PCE or SDN controller compute a path each time connectivity between two data centers is to be checked. This will take a fair amount of time and CPU cycles. Efficiency can be improved through memoization of the path computation and its result, the first time the PCE or SDN controller returns a value. Before asking the PCE to compute a path, a check of the cache, for the result of any earlier computation, can be made. Keys would be the endpoint data centers, the network service ID and the virtual link ID.

## 10.5. Pre-computation of a Connectivity Matrix

One problem identified in the previous solution is the number of delays that accumulate due to repeated path requests to the PCE/ controller. Memoization is identified as a method for reducing the impact. Nevertheless, the response time to instantiate a service will be longer than it need be. For a network service with m VNF (SF), if each is in a separate site, then, at best, m-1 requests to the PCE/controller are required. If there are problems finding paths, the count of path computations rises. This has the potential to be computationally costly and require a lot of time before providing a satisfactory solution.

A better solution would be to pre-compute connectivity among the sites and make this available for the algorithm. This will likely increase the overall compute load, but it would shorten the time to respond to service requests. In this case a connection is computed for each pair of data centers and possible endpoints (user sites). It is maintained through continuous requests of the PCE, both triggered by a baseline frequency as well as reported events that could impact connection. Assuming a total of n data centers and other sites, this results in a connectivity matrix that is n x n. It may be sparse, as some sites may not connect to others, and of course, a site does not connect to itself. Each entry in the matrix represents a path between the sites, and each has relevant bandwidth, latency, etc. attributes.

Here now the computation, whatever the algorithm, is much more efficient. Connectivity information for any pair of data centers or possible endpoints is locally available.  There is no need to request information from another process, the PCE or other SDN controller.

## 10.6.  Multi-valued Connectivity Matrix

Unfortunately, as described above, this solution still has limitations.  Only one path, perhaps selected based on greatest available bandwidth, is computed for each entry in the n x n connectivity matrix.  This path is not necessarily engineered to maximize or weight the path metrics in any way.  If a path between two datacenters needs a latency of 40 msec and the path in the connectivity matrix has latency of 80 msec, that option will be rejected.  This is a problem because a perfectly acceptable path with the needed bandwidth might exist along a different route.

A possible solution here is to compute more than one path for each pair of sites, where each is based on a different set of criteria priorities.  These could include maximum bandwidth, minimum latency, fewest encapsulations (tunnels, tunnels in tunnels, packet layer to optical and back), fewest hops, fewest committed tunnels, least dynamic traffic patterns, etc.)  Others might include different balances of these values, and anti-affinity rules, e.g. do not share the same underlying optical paths, something important for identifying connectivity with backup paths.

The result, therefore, is an n x n x m array, or simply an n x n array with each cell containing a list of path options, where the list length may vary for different sets of endpoints.  Each list entry (or cell in the m dimension for a 3 dimensional array) will represent a unique path supporting the connectivity between the two endpoints, where each has been computed for a different set of service priorities.

There is still the concern that a network service might be requested for which a needed path's requirements (from a virtual link) are not satisfied by any entry in the connectivity matrix.  In other words, no connection with the relevant service requirements has been pre-computed.  The implication is that this could result in an unnecessary rejection of the requested network service.  To address this, it might be appropriate to include an option for the path or the orchestrator.  If the deplyment algorithm finds no satisfactory path in the matrix, the orchestrator could request directly to the PCE to see if it can find connectivity to meet its needs.

Note: In the matrix, there may be endpoint pairs for which no path has been found.  The value in the matrix would be FAIL, or perhaps include something with a bit more useful information.  If a request fails because the requirements do not match any of the cached paths, it is an option to ask the PCE to re-check, or not.

There are a number of ways the orchestrator could approach doing this.  The orchestrator might request the PCE to compute a point to point path that meets the service requirements, while itself, IN PARALLEL, continues to compute a different VNF placement and path solution.  If the PCE/controller finds a possible path to support the connection, or perhaps even if not, it would also make sense to add it to the connectivity matrix and change the path request options so that it is continuously computed and checked.

## 10.7.  Additional Optimization Options

To this point the connectivity matrix has been described as n x n, with each possible endpoint, e.g. a data center or customer network, being represented.  In other words, there are n possible endpoints. For any pair of endpoints, if there is some connectivity between them, even if across multiple physical networks and domains, one or more paths between them could be continuously computed and stored.

However, this is probably unnecessary.  Based on data center sites, physical and virtual connectivity, available functions, and customer interests, the sites that might connect with other sites are likely a significant subset of the full n x n possible connections. Therefore, it would make sense to weight the site-2-site connections.

A very simple method is to compute paths only for site-to-site pairs that commonly connect as part of network services.  Other site-to-site paths would remain empty.  There are likely a number of ways to improve this based upon other factors, including learned behavior. One option is to initially compute a full n x n connectivity matrix, with mutiple paths for different priorities.  Subsequent refreshing of any connection depends upon the frequency of the use of that connection.  The frequency of updates would depend upon the frequency of the connection use.  Lower frequency connections will be assigned a lower weight.  The weight is refreshed whenever the connection is used, with the weight raised and the 'time last assigned' updated. On the other hand, another agent will execute periodically, and it will reduce each weight by a certain amount (though no lower than zero.)

When a new service is to be placed, if the orchestrator finds a lower weight connection to be useful, it might choose to use it.  If the age is older than some configured threshold, it might ask the PCE to re-compute the path before it uses it.

## 10.8. Cached Paths Over Multiple Sites

In addition to the pre-computed site-2-site paths, another optimization step is to cache full or partial multi-site or multi data center paths.  Is is likely there will be a fixed number of network services and each with a fixed, or mostly fixed, order of network functions.  Even if some network services are dynamic in order, size and content (NFs), these will likely be a small minority, at the outset of NFV based network service delivery.  Therefore, it will be possible to identify a finite number of network services, including their VNF-FGs, i.e. the ordering of the NFs, and needed compute and network resources.  The frequencies and instantiation patterns of these services and their locations can be used to drive how often and how many cached instances should be computed, in order to ensure resources are available.  Computing for multiple links in a network service also provides the opportunity to sychronize the computations of available resources.  This should avoid the possibility of any connections in a network service unintentionally relying on the availability of bandwidth on the same physical link.

## 10.9. Summary

The purpose of this section has been to present options for computing optimized network service deployment that might drive requirements for a network SDN controller or PCE.  The organization of this section has been intended to build and make clear how the use of pre-computed and cached paths between the data center sites can benefit in the delivery of optimzed services.  It has described the creation and management of computing multiple paths for each pairwise set of data centers, based on different weights of factors, e.g. latency, bandwidth, etc.  It also has outlined how pre-computed full service paths might also work.  The result should be a greater likelihood of being able to deliver a service, as well as a significant reduction in the time to respond to service requests.

## 11. Connectivity based on Potential vs. Actual Topologies

Something missing here is how the orchestrator and PCE should consider 'potential connectivity' as part of the VNF placement and connectivity process.  For example, in the HPC use case, we know there will likely be a need for a high bandwidth low latency connection.  This is probably something that is not available in the connectivity matrix as described so far.  The matrix includes only information from the existing network topology configuration, including the configuration of existing tunnels, and performance metrics.

This calls out that there is a need for an agent providing connectivity information, e.g.  PCE or SDN controller, to also present potential connectivity, should it be needed.  This is not a simply issue.  For example, updating or introducing a new underlying optical network circuit will affect the topology of the packet network running on top of it.  This means that existing paths between sites would no longer be valid.

An orchestrator's deployment or placement algorithm should have the option to request this sort of 'potential' connectivity for any pair of sites, and the PCE or network controller that is computing these paths should be able to provide this.

This functionality could be used for near term needs, perhaps dedicating that circuit for a single network service, such as the HPC service.  In the long term, other advantages might be realized.  For example, if the system is creating a large number of multi-hop connections between two data centers, the orchestrator or PCE may recognize this and propose a possible direct circuit connection to reduce costs and latency.  This is an area for more analysis.  It is something that will drive new information exchange between functional blocks, e.g. the PCE or SDN controller, via its NBI and perhaps SBI, if hierarchical architectures are used.

## 12.  Virtual Networks Equivalency

To this point the description has been for a relatively simple network, with paths computed for different data center connections. Another perspective on this is to consider each data center and user endpoint as a node in the network, where these nodes are connected by one or more virtual links.  This is a virtual network then, where there is an abstraction layer between the physical and virtual networks.  Each entry in the connectivity matrix described above represents a link between any two data centers.  There may be more than one link, as these multiple links represent the multiple entries per matrix cell, with each created based on different priorities e.g. bandwidth, latency, etc.  In fact, the connectivity matrix described effectively describes this sort of virtual network.  As noted earlier, the connectivity matrix may be sparse.  Equivalently, the virtual network may not be show a link between every two nodes (data centers, etc.)

One thing to consider is the possibility of multiple virtual networks, where each supports a different customer or some type of independent domain.  Multiple virtual networks (connectivity matrices) created, for each per customer or domain.  There might even be layering.  For example, a customer may use virtual networks (plus compute resources) from multiple providers, weave these together, and then offer services to its own customers.  This is equivalent to having specific connectivity matrices from each separate network and then combining these to create per customer connectivity matrices that combine information from the separate network matrices.

## 13.  Implications for the Network Controllers (SDN / PCE) and Orchestrators

First, the network connectivity matrix (virtual network) must be continuously computed and the connection (link between sites) information passed to the NFV network service orchestrator as updates are generated.  The computations must consider topology, current resource allocations, measured utilization, analytics, e.g. trending, baselines, patterns, and policies.  Path computation must compute multiple paths concurrenty, for the same endpoints, but different priorities, policies, etc.  The computations are not for one set of service parameters, but to identify for a link the available resources (typically bandwidth) for meeting certain service requirements.

A PCE path request should configure path re-computation and updates at a constant rate, e.g. every 5 minutes, as well as compute updates whenever an event triggers it.  Such an event might be a change in bandwidth, or perhaps latency, due to allocation over a path currently in the connectivity matrix, or congestion due to traffic volumes.  Each time the PCE computes a path it must do so with the latest information regarding connectivity, link state, reservations, and utilization and performance metrics.

Note, unfortunatelypre-computation can create other problems.  For example, when computing a path, it is possible that the same physical (or virtual) link could be used in more than one NF-NF connection for a single network service.  A conflict might exist resulting in inadequate resources, and this would not be recognized.  For smaller services this might not be significant; however, for services that might require a lot of bandwidth, e.g. a large vCDN deployment, this might make a difference.

Finally, for the cached multi-site and connection solution, this will require support for requesting and then computing the hop by hop paths, synchronized, and caching them such that a full soluton can be delivered quickly.

## 14.  Relationship to TEAS TE Topology and ACTN

In the IETF there is a lot of relevant ongoing related to enhancing the PCE capabilities, the PCE Protocol, and the introduction of the Abstraction and Control of Transport Networks (ACTN) framework.  A number of concepts described here are really relevant to transport network services (including MPLS, segment routing, and technologies other than optical), whther supporting NFV network services or not. Some of these are described in the TEAS WG TE Topology, currently in development. These include the description and definitions of network nodes, TE virtual links, and multiple layers of abstraction. PCE enhancements are defined to support this, and the ACTN framework address management and control from the service layer.

## 15. Areas of Future Evaluation

This is an early draft of this Internet draft. Its intention has been to highlight, through use cases, the need for co-ordinated optimization of network and compute resources for NFV based network services delivery and to expand upon some solution options that lead to possible new requirements of the network controller (SDN / PCE.) There are a number of topics raised here, or which are related, that require further study. These include:

o Testbed to evaluate effectiveness of pre-computation of connectivity and full network service paths.

o Evaluate requirements to support forecast or pre-sechedule network services.

o Determine any controller / PCE (PCEP) NBI / SBI messaging updates that might be needed

o Evaluate implications of non-ACID computations vs. deployments and methods to reduce risk.

o Centralized vs. Distributed computation of optimization

o Review work in other IETF and standardswork - ONF Transport, MEF, FORCES, ALTO, NFVRG, SDNRG, SFC WG, etc.

## 16. IANA Considerations

This memo includes no request to IANA.

All drafts are required to have an IANA considerations section (see Guidelines for Writing an IANA Considerations Section in RFCs [RFC5226] for a guide). If the draft does not require IANA to do anything, the section contains an explicit statement that this is the case (as above). If there are no requirements for IANA, the section will be removed during conversion into an RFC by the RFC Editor.

## 17. Security Considerations

All drafts are required to have a security considerations section. See RFC 3552 [RFC3552] for a guide.

## 18. References

### 18.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

### 18.2. Informative References

[I-D.ietf-pce-hierarchy-extensions]
Zhang, F., Zhao, Q., Dios, O., Casellas, R., and D. King, "Extensions to Path Computation Element Communication Protocol (PCEP) for Hierarchical Path Computation Elements (PCE)", draft-ietf-pce-hierarchy-extensions-03 (work in progress), July 2016.

[I-D.ietf-pce-pcep-yang]
Dhody, D., Hardwick, J., Beeram, V., and j. jefftant@gmail.com, "A YANG Data Model for Path Computation Element Communications Protocol (PCEP)", draft-ietf-pce-pcep-yang-01 (work in progress), October 2016.

[I-D.ietf-pce-stateful-pce]
Crabbe, E., Minei, I., Medved, J., and R. Varga, "PCEP Extensions for Stateful PCE", draft-ietf-pce-stateful-pce-17 (work in progress), November 2016.

[I-D.ietf-teas-actn-framework]
Ceccarelli, D. and Y. Lee, "Framework for Abstraction and Control of Traffic Engineered Networks", draft-ietf-teas-actn-framework-01 (work in progress), October 2016.

[I-D.ietf-teas-pce-central-control]
Farrel, A., Zhao, Q., Li, Z., and C. Zhou, "An Architecture for Use of PCE and PCEP in a Network with Central Control", draft-ietf-teas-pce-central-control-00 (work in progress), September 2016.

[I-D.ietf-teas-yang-te]
Saad, T., Gandhi, R., Liu, X., Beeram, V., Shah, H., Bryskin, I., Chen, X., Jones, R., and B. Wen, "A YANG Data Model for Traffic Engineering Tunnels and Interfaces", draft-ietf-teas-yang-te-05 (work in progress), October 2016.

[I-D.ietf-teas-yang-te-topo]
Liu, X., Bryskin, I., Beeram, V., Saad, T., Shah, H., and O. Dios, "YANG Data Model for TE Topologies", draft-ietf-teas-yang-te-topo-06 (work in progress), October 2016.

[RFC3552] Rescorla, E. and B. Korver, "Guidelines for Writing RFC Text on Security Considerations", BCP 72, RFC 3552, DOI 10.17487/RFC3552, July 2003, <http://www.rfc-editor.org/info/rfc3552>.

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, DOI 10.17487/RFC5226, May 2008, <http://www.rfc-editor.org/info/rfc5226>.

[RFC5441] Vasseur, JP., Ed., Zhang, R., Bitar, N., and JL. Le Roux, "A Backward-Recursive PCE-Based Computation (BRPC) Procedure to Compute Shortest Constrained Inter-Domain Traffic Engineering Label Switched Paths", RFC 5441, DOI 10.17487/RFC5441, April 2009, <http://www.rfc-editor.org/info/rfc5441>.

## Author's Address

Andrew Veitch (editor)
Netcracker, Inc.
95 Sawyer Road
Waltham  02453
US

Email: andrew.veitch@netcracker.com