Workgroup: RTG Working Group
Internet-Draft:
draft-bookham-rtgwg-nfix-arch-04
Published: 5 January 2022
Intended Status: Informational
Expires: 9 July 2022

Authors: C. Bookham, Ed. (Nokia), A. Stone (Nokia), J. Tantsura (Microsoft), M. Durrani (Equinix Inc), B. Decraene (Orange)

An Architecture for Network Function Interconnect

Abstract

The emergence of technologies such as 5G, the Internet of Things (IoT), and Industry 4.0, coupled with the move towards network function virtualization, means that the service requirements demanded from networks are changing. This document describes an architecture for a Network Function Interconnect (NFIX) that allows for interworking of physical and virtual network functions in a unified and scalable manner across wide-area network and data center domains while maintaining the ability to deliver against SLAs.

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 BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

With the introduction of technologies such as 5G, the Internet of Things (IoT), and Industry 4.0, service requirements are changing. In addition to the ever-increasing demand for more capacity, these services have other stringent service requirements that need to be met such as ultra-reliable and/or low-latency communication.

Parallel to this, there is a continued trend to move towards network function virtualization. Operators are building digitalized infrastructure capable of hosting numerous virtualized network functions (VNFs). Infrastructure that can scale in and scale out depending on the application demand and can deliver flexibility and service velocity. Much of this virtualization activity is driven by the afore-mentioned emerging technologies as new infrastructure is deployed in support of them. To try and meet the new service requirements some of these VNFs are becoming more dispersed, so it is common for networks to have a mix of centralized medium- or large-sized sized data centers together with more distributed smaller 'edge-clouds'. VNFs hosted within these data centers require seamless connectivity to each other, and to their existing physical network function (PNF) counterparts. This connectivity also needs to deliver against agreed SLAs.

Coupled with the deployment of virtualization is automation. Many of these VNFs are deployed within SDN-enabled data centers where automation is simply a must-have capability to improve service activation lead-times. The expectation is that services will be instantiated in an abstract point-and-click manner and be automatically created by the underlying network, dynamically adapting to service connectivity changes as virtual entities move between hosts.

This document describes an architecture for a Network Function Interconnect (NFIX) that allows for interworking of physical and virtual network functions in a unified and scalable manner. It describes a mechanism for establishing connectivity across multiple discreet domains in both the wide-area network (WAN) and the data center (DC) while maintaining the ability to deliver against SLAs. To achieve this NFIX works with the underlying topology to build a unified over-the-top topology.

The NFIX architecture described in this document does not define any new protocols but rather outlines an architecture utilizing a collaboration of existing standards-based protocols.

2. Terminology

- A physical network function (PNF) refers to a network device such as a Provider Edge (PE) router that connects physically to the wide-area network.

- A virtualized network function (VNF) refers to a network device such as a provider edge (PE) router that is hosted on an application server. The VNF may be bare-metal in that it consumes the entire resources of the server, or it may be one of numerous virtual functions instantiated as a VM or number of containers on a given server that is controlled by a hypervisor or container management platform.

- A Data Center Border (DCB) router refers to the network function that spans the border between the wide-area and the data center networks, typically interworking the different encapsulation techniques employed within each domain.

- An Interconnect controller is the controller responsible for managing the NFIX fabric and services.

- A DC controller is the term used for a controller that resides within an SDN-enabled data center and is responsible for the DC network(s)

3. Motivation

Industrial automation and business-critical environments use applications that are demanding on the network. These applications present different requirements from low-latency to high-throughput, to application-specific traffic conditioning, or a combination. The evolution to 5G equally presents challenges for mobile back-, front- and mid-haul networks. The requirement for ultra-reliable low-latency communication means that operators need to re-evaluate their network architecture to meet these requirements.

At the same time, the service edge is evolving. Where the service edge device was historically a PNF, the adoption of virtualization means VNFs are becoming more commonplace. Typically, these VNFs are hosted in some form of data center environment but require end-to-end connectivity to other VNFs and/or other PNFs. This represents a challenge because generally transport layer connectivity differs between the WAN and the data center environment. The WAN includes all levels of hierarchy (core, aggregation, access) that form the networks footprint, where transport layer connectivity using IP/MPLS is commonplace. In the data center native IP is commonplace, utilizing network virtualization overlay (NVO) technologies such as virtual extensible LAN (VXLAN) [RFC7348], network virtualization using generic routing encapsulation (NVGRE) [RFC7637], or generic network virtualization encapsulation (GENEVE) [I-D.ietf-nvo3-geneve]. There is a requirement to seamlessly integrate these islands and avoid heavy-lifting at interconnects as well as providing a means to provision end-to-end services with a single touch point at the edge.

The service edge boundary is also changing. Some functions that were previously reasonably centralized are now becoming more distributed. One reason for this is to attempt to deal with low latency requirements. Another reason is that operators seek to reduce costs by deploying low/medium-capacity VNFs closer to the edge. Equally, virtualization also sees some of the access network moving towards the core. Examples of this include cloud-RAN or Software-Defined Access Networks.

Historically service providers have architected data centers independently from the wide-area network, creating two independent domains or islands. As VNFs become part of the service landscape the service data-path must be extended across the WAN into the data center infrastructure, but in a manner that still allows operators to meet deterministic performance requirements. Methods for stitching WAN and DC infrastructures together with some form of service-interworking at the data center border have been implemented and deployed, but this service-interworking approach has several limitations:

- The data center environment typically uses encapsulation techniques such as VXLAN or NVGRE while the WAN typically uses encapsulation techniques such as MPLS [RFC3031]. Underlying optical infrastructure might also need to be programmed. These are incompatible and require interworking at the service layer.

- It typically requires heavy-touch service provisioning on the data center border. In an end-to-end service, midpoint provisioning is undesirable and should be avoided.

- Automation is difficult; largely due to the first two points but with additional contributing factors. In the virtualization world automation is a must-have capability.

- When a service is operating at Layer 3 in a data center with redundant interconnects the risk of routing loops exists. There is no inherent loop avoidance mechanism when redistributing routes between address families so extreme care must be taken. Proposals such as the Domain Path (D-PATH) attribute [I-D.ietf-bess-evpn-ipvpn-interworking] attempt to address this issue but as yet are not widely implemented or deployed.

- Some or all the above make the service-interworking gateway cumbersome with questionable scaling attributes.

Hence there is a requirement to create an open, scalable, and unified network architecture that brings together the wide-area network and data center domains. It is not an architecture e xclusively targeted at greenfield deployments, nor does it require a flag day upgrade to deploy in a brownfield network. It is an evolutionary step to a consolidated network that uses the constructs of seamless MPLS [I-D.ietf-mpls-seamless-mpls] as a baseline and extends upon that to include topologies that may not be link-state based and to provide end-to-end path control. Overall the NFIX architecture aims to deliver the following:

- Allows for an evolving service edge boundary without having to constantly restructure the architecture.

- Provides a mechanism for providing seamless connectivity between VNF to VNF, VNF to PNF, and PNF to PNF, with deterministic SLAs, and with the ability to provide differentiated SLAs to suit different service requirements.

- Delivers a unified transport fabric using Segment Routing (SR) [RFC8402] where service delivery mandates touching only the service edge without imposing additional encapsulation requirements in the DC.

- Embraces automation by providing an environment where any end-to-end connectivity can be instantiated in a single request manner while maintaining SLAs.

4. Requirements

The following section outlines the requirements that the proposed solution must meet. From an overall perspective, the proposed generic architecture must:

- Deliver end-to-end transport LSPs using traffic-engineering (TE) as required to meet appropriate SLAs for the service using(s) using those LSPs. End-to-end refers to VNF and/or PNF connectivity or a combination of both.

- Provide a solution that allows for optimal end-to-end path placement; where optimal not only meets the requirements of the path in question but also meets the global network objectives.

- Support varying types of VNF physical network attachment and logical (underlay/overlay) connectivity.

- Facilitate automation of service provision. As such the solution should avoid heavy-touch service provisioning and decapsulation/encapsulation at data center border routers.

- Provide a framework for delivering logical end-to-end networks using differentiated logical topologies and/or constraints.

- Provide a high level of stability; faults in one domain should not propagate to another domain.

- Provide a mechanism for homogeneous end-to-end OAM.

- Hide/localize instabilities in the different domains that participate in the end-to-end service.

- Provide a mechanism to minimize the label-stack depth required at path head-ends for SR-TE LSPs.

- Offer a high level of scalability.

- Although not considered in-scope of the current version of this document, the solution should not preclude the deployment of multicast. This subject may be covered in later versions of this document.

5. Theory of Operation

This section describes the NFIX architecture including the building blocks and protocol machinery that is used to form the fabric. Where considered appropriate rationale is given for selection of an architectural component where other seemingly applicable choices could have been made.

5.1. VNF Assumptions

For the sake of simplicity, references to VNF are made in a broad sense. Equally, the differences between VNF and Container Network Function (CNF) are largely immaterial for the purposes of this document, therefore VNF is used to represent both. The way in which a VNF is instantiated and provided network connectivity will differ based on environment and VNF capability, but for conciseness this is not explicitly detailed with every reference to a VNF. Common examples of VNF variants include but are not limited to:

- A VNF that functions as a routing device and has full IP routing and MPLS capabilities. It can be connected simultaneously to the data center fabric underlay and overlay and serves as the NVO tunnel endpoint [RFC8014]. Examples of this might be a virtualized PE router, or a virtualized Broadband Network Gateway (BNG).

*A VNF that functions as a device (host or router) with limited IP routing capability. It does not connect directly to the data center fabric underlay but rather connects to one or more external physical or virtual devices that serve as the NVO tunnel endpoint(s). It may however have single or multiple connections to the overlay. Examples of this might be a mobile network control or management plane function.

*A VNF that has no routing capability. It is a virtualized function hosted within an application server and is managed by a hypervisor or container host. The hypervisor/container host acts as the NVO endpoint and interfaces to some form of SDN controller responsible for programming the forwarding plane of the virtualization host using, for example, OpenFlow. Examples of this might be an Enterprise application server or a web server running as a virtual machine and front-ended by a virtual routing function such as OVS/xVRS/VTF.

Where considered necessary exceptions to the examples provided above or focus on a particular scenario will be highlighted.

5.2. Overview

The NFIX architecture makes no assumptions about how the network is physically composed, nor does it impose any dependencies upon it. It also makes no assumptions about IGP hierarchies and the use of areas/levels or discrete IGP instances within the WAN is fully endorsed to enhance scalability and constrain fault propagation. This could apply for instance to a hierarchical WAN from core to edge or from WAN to LAN connections. The overall architecture uses the constructs of seamless MPLS as a baseline and extends upon that. The concept of decomposing the network into multiple domains is one that has been widely deployed and has been proven to scale in networks with large numbers of nodes.

The proposed architecture uses segment routing (SR) as its preferred choice of transport. Segment routing is chosen for construction of end-to-end LSPs given its ability to traffic-engineer through source-routing while concurrently scaling exceptionally well due to its lack of network state other than the ingress node. This document uses SR instantiated on an MPLS forwarding plane(SR-MPLS), although it does not preclude the use of SRv6 either now or at some point in the future. The rationale for selecting SR-MPLS is simply maturity and more widespread applicability across a potentially broad range of network devices. This document may be updated in future versions to include more description of SRv6 applicability.

5.3. Use of a Centralized Controller

It is recognized that for most operators the move towards the use of a controller within the wide-area network is a significant change in operating model. In the NFIX architecture it is a necessary component. Its use is not simply to offload inter-domain path calculation from network elements; it provides many more benefits:

- It offers the ability to enforce constraints on paths that originate/terminate on different network elements, thereby providing path diversity, and/or bidirectionality/co-routing, and/or disjointness.

- It avoids collisions, re-tries, and packing problems that has been observed in networks using distributed TE path calculation, where head-ends make autonomous decisions.

- A controller can take a global view of path placement strategies, including the ability to make path placement decisions over a high number of LSPs concurrently as opposed to considering each LSP independently. In turn, this allows for 'global' optimization of network resources such as available capacity.

- A controller can make decisions based on near-real-time network state and optimize paths accordingly. For example, if a network link becomes congested it may recompute some of the paths transiting that link to other links that may not be quite as optimal but do have available capacity. Or if a link latency crosses a certain threshold, it may select to reoptimize some latency-sensitive paths away from that link.

- The logic of a controller can be extended beyond pure path computation and placement. If the controller is aware of services, service requirements, and available paths within the network it can cross-correlate between them and ensure that the appropriate paths are used for the appropriate services.

- The controller can provide assurance and verification of the underlying SLA provided to a given service.

As the main objective of the NFIX architecture is to unify the data center and wide-area network domains, using the term controller is not sufficiently succinct. The centralized controller may need to interface to other controllers that potentially reside within an SDN-enabled data center. Therefore, to avoid interchangeably using the term controller for both functions, we distinguish between them simply by using the terms 'DC controller' which as the name suggests is responsible for the DC, and 'Interconnect controller' responsible for managing the extended SR fabric and services.

The Interconnect controller learns wide-area network topology information and allocation of segment routing SIDs within that domain using BGP link-state [RFC7752] with appropriate SR extensions. Equally it learns data center topology information and Prefix-SID allocation using BGP labeled unicast [RFC8277] with appropriate SR extensions, or BGP link-state if a link-state IGP is used within the data center. If Route-Reflection is used for exchange of BGP link-state or labeled unicast NLRI within one or more domains, then the Interconnect controller need only peer as a client with those Route-Reflectors in order to learn topology information.

Where BGP link-state is used to learn the topology of a data center (or any IGP routing domain) the BGP-LS Instance Identifier (Instance-ID) is carried within Node/Link/Prefix NLRI and is used to identify a given IGP routing domain. Where labeled unicast BGP is used to discover the topology of one or more data center domains there is no equivalent way for the Interconnect controller to achieve a level of routing domain correlation. The controller may learn some splintered connectivity map consisting of 10 leaf switches, four spine switches, and four DCB's, but it needs some form of key to inform it that leaf switches 1-5, spine switches 1 and 2, and DCB's 1 and 2 belong to data center 1, while leaf switches 6-10, spine switches 3 and 4, and DCB's 3 and 4 belong to data center 2. What is needed is a form of 'data center membership identification' to provide this correlation. Optionally this could be achieved at BGP level using a standard community to represent each data center, or it could be done at a more abstract level where for example the DC controller provides the membership identification to the Interconnect controller through an application programming interface (API).

Understanding real-time network state is an important part of the Interconnect controllers role, and only with this information is the controller able to make informed decisions and take preventive or corrective actions as necessary. There are numerous methods implemented and deployed that allow for harvesting of network state, including (but not limited to) IPFIX [RFC7011], Netconf/YANG [RFC6241][RFC6020], streaming telemetry, BGP link-state [RFC7752] [I-D.ietf-idr-te-lsp-distribution], and the BGP Monitoring Protocol (BMP) [RFC7854].

5.4. Routing and LSP Underlay

This section describes the mechanisms and protocols that are used to establish end-to-end LSPs; where end-to-end refers to VNF-to-VNF, PNF-to-PNF, or VNF-to-PNF.

5.4.1. Intra-Domain Routing

In a seamless MPLS architecture domains are based on geographic dispersion (core, aggregation, access). Within this document a domain is considered as any entity with a captive topology; be it a link-state topology or otherwise. Where reference is made to the wide-area network domain, it refers to one or more domains that constitute the wide-area network domain.

This section discusses the basic building blocks required within the wide-area network and the data center, noting from above that the wide-area network may itself consist of multiple domains.

5.4.1.1. Wide-Area Network Domains

The wide-area network includes all levels of hierarchy (core, aggregation, access) that constitute the networks MPLS footprint as well as the data Center border routers. Each domain that constitutes part of the wide-area network runs a link-state interior gateway protocol (IGP) such as ISIS or OSPF, and each domain may use IGP-inherent hierarchy (OSPF areas, ISIS levels) with an assumption that visibility is domain-wide using, for example, L2 to L1 redistribution. Alternatively, or additionally, there may be multiple domains that are split by using separate and distinct instances of IGP. There is no requirement for IGP redistribution of any link or loopback addresses between domains.

Each IGP should be enabled with the relevant extensions for segment routing [RFC8667][RFC8665], and each SR-capable router should advertise a Node-SID for its loopback address, and an Adjacency-SID (Adj-SID) for every connected interface (unidirectional adjacency) belonging to the SR domain. SR Global Blocks (SRGB) can be allocated to each domain as deemed appropriate to specific network requirements. Border routers belonging to multiple domains have an SRGB for each domain.

The default forwarding path for intra-domain LSPs that do not require TE is simply an SR LSP containing a single label advertised by the destination as a Node-SID and representing the ECMP-aware shortest path to that destination. Intra-domain TE LSPs are constructed as required by the Interconnect controller. Once a path is calculated it is advertised as an explicit SR Policy [I-D.ietf-spring-segment-routing-policy] containing one or more paths expressed as one or more segment-lists, which may optionally contain binding SIDs if requirements dictate. An SR Policy is identified through the tuple [headend, color, endpoint] and this tuple is used extensively by the Interconnect controller to associate services with an underlying SR Policy that meets its objectives.

To provide support for ECMP the Entropy Label [RFC6790][RFC8662] should be utilized. Entropy Label Capability (ELC) should be advertised into the IGP using the IS-IS Prefix Attributes TLV [I-D.ietf-isis-mpls-elc] or the OSPF Extended Prefix TLV [I-D.ietf-ospf-mpls-elc] coupled with the Node MSD Capability sub-TLV to advertise Entropy Readable Label Depth (ERLD) [RFC8491][RFC8476] and the base MPLS Imposition (BMI). Equally, support for ELC together with the supported ERLD should be signaled in BGP using the BGP Next-Hop Capability [I-D.ietf-idr-next-hop-capability]. Ingress nodes and or DCBs should ensure sufficient entropy is applied to packets to exercise available ECMP links.

5.4.1.2. Data Center Domain

The data center domain includes all fabric switches, network virtualization edge (NVE), and the data center border routers. The data center routing design may align with the framework of [RFC7938] running eBGP single-hop sessions established over direct point-to-point links, or it may use an IGP for dissemination of topology information. This document focuses on the former, simply because the ue of an IGP largely makes the data centers behaviour analogous to that of a wide-area network domain.

The chosen method of transport or encapsulation within the data center for NFIX is SR-MPLS over IP/UDP [RFC8663] or, where possible, native SR-MPLS. The choice of SR-MPLS over IP/UDP or native SR-MPLS allows for good entropy to maximize the use of equal-cost Clos fabric links. Native SR-MPLS encapsulation provides entropy through use of the Entropy Label, and, like the wide-area network, support for ELC together with the support ERLD should be signaled using the BGP Next-Hop Capability attribute. As described in [RFC6790] the ELC is an indication from the egress node of an MPLS tunnel to the ingress node of the MPLS tunnel that is is capable of processing an Entropy Label. The BGP Next-Hop Capability is a non-transitive attribute which is modified or deleted when the next-hop is changed to reflect the capabilities of the new next-hop. If we assume that the path of a BGP-signaled LSP transits through multiple ASNs, and/or a single ASN with multiple next-hops, then it is not possible for the ingress node to determine the ELC of the egress node. Without this end-to-end signaling capability the entropy label must only be used when it is explicitly known, through configuration or other means, that the egress node has support for it. Entropy for SR-MPLS over IP/UDP encapsulation uses the source UDP port for IPv4 and the Flow Label for IPv6. Again, the ingress network function should ensure sufficient entropy is applied to exercise available ECMP links.

Another significant advantage of the use of native SR-MPLS or SR-MPLS over IP/UDP is that it allows for a lightweight interworking function at the DCB without the requirement for midpoint provisioning; interworking between the data center and the wide-area network domains becomes an MPLS label swap/continue action.

Loopback addresses of network elements within the data center are advertised using labeled unicast BGP with the addition of SR Prefix SID extensions [RFC8669] containing a globally unique and persistent Prefix-SID. The data-plane encapsulation of SR-MPLS over IP/UDP or native SR-MPLS allows network elements within the data center to consume BGP Prefix-SIDs and legitimately use those in the encapsulation.

5.4.2. Inter-Domain Routing

Inter-domain routing is responsible for establishing connectivity between any domains that form the wide-area network, and between the wide-area network and data center domains. It is considered unlikely that every end-to-end LSP will require a TE path, hence there is a requirement for a default end-to-end forwarding path. This default forwarding path may also become the path of last resort in the event of a non-recoverable failure of a TE path. Similar to the seamless MPLS architecture this inter-domain MPLS connectivity is realized using labeled unicast BGP [RFC8277] with the addition of SR Prefix SID extensions.

Within each wide-area network domain all service edge routers, DCBs, and ABRs/ASBRs form part of the labeled BGP mesh, which can be either full-mesh, or more likely based on the use of route-reflection. Each of these routers advertises its respective loopback addresses into labeled BGP together with an MPLS label and a globally unique Prefix-SID. Routes are advertised between wide-area network domains by ABRs/ASBRs that impose next-hop-self on advertised routes. The function of imposing next-hop-self for labeled routes means that the ABR/ASBR allocates a new label for advertised routes and programs a label-swap entry in the forwarding plane for received and advertised routes. In short it becomes part of the forwarding path.

DCB routers have labeled BGP sessions towards the wide-area network and labeled BGP sessions towards the data center. Routes are bidirectionally advertised between the domains subject to policy, with the DCB imposing itself as next-hop on advertised routes. As above, the function of imposing next-hop-self for labeled routes implies allocation of a new label for advertised routes and a label-swap entry being programmed in the forwarding plane for received and advertised labels. The DCB thereafter becomes the anchor point between the wide-area network domain and the data center domain.

Within the wide-area network next-hops for labeled unicast routes containing Prefix-SIDs are resolved to SR LSPs, and within the data center domain next-hops for labeled unicast routes containing Prefix-SIDs are resolved to SR LSPs or IP/UDP tunnels. This provides end-to-end connectivity without a traffic-engineering capability.

```
   +---------------+   +----------------+   +---------------+
   |  Data Center  |   |   Wide-Area    |   |   Wide-Area   |
   |              +-----+   Domain 1   +-----+  Domain ‘n’  |
   |              | DCB |              | ABR |              |
   |              +-----+              +-----+              |
   |               |   |                |   |               |
   +---------------+   +----------------+   +---------------+
   <-- SR/SRoUDP -->   <---- IGP/SR ---->   <--- IGP/SR ---->
   <--- BGP-LU ---> NHS <--- BGP-LU ---> NHS <--- BGP-LU --->
```

Figure 1

Default Inter-Domain Forwarding Path

5.4.3. Intra-Domain and Inter-Domain Traffic-Engineering

The capability to traffic-engineer intra- and inter-domain end-to-end paths is considered a key requirement in order to meet the service objectives previously outlined. To achieve optimal end-to-end path placement the key components to be considered are path calculation, path activation, and FEC-to-path binding procedures.

In the NFIX architecture end-to-end path calculation is performed by the Interconnect controller. The mechanics of how the objectives of each path is calculated is beyond the scope of this document. Once a path is calculated based upon its objectives and constraints, the path is advertised from the controller to the LSP headend as an explicit SR Policy containing one or more paths expressed as one or more segment-lists. An SR Policy is identified through the tuple [headend, color, endpoint] and this tuple is used extensively by the Interconnect controller to associate services with an underlying SR Policy that meets its objectives.

The segment-list of an SR Policy encodes a source-routed path towards the endpoint. When calculating the segment-list the Interconnect controller makes comprehensive use of the Binding-SID (BSID), instantiating BSID anchors as necessary at path midpoints when calculating and activating a path. The use of BSID is considered fundamental to segment routing as described in [I-D.filsfils-spring-sr-policy-considerations]. It provides opacity between domains, ensuring that any segment churn is constrained to a single domain. It also reduces the number of segments/labels that the headend needs to impose, which is particularly important given that network elements within a data center generally have limited label imposition capabilities. In the context of the NFIX architecture it is also the vehicle that allows for removal of heavy midpoint provisioning at the DCB.

For example, assume that VNF1 is situated in data center 1, which is interconnected to the wide-area network via DCB1. VNF1 requires connectivity to VNF2, situated in data center 2, which is interconnected to the wide-area network via DCB2. Assuming there is no existing TE path that meet VNF1's requirements, the Interconnect controller will:

*Instantiate an SR Policy on DCB1 with BSID n and a segment-list containing the relevant segments of a TE path to DCB2. DCB1 therefore becomes a BSID anchor.

*Instantiate an SR Policy on VNF1 with BSID m and a segment-list containing segments {DCB1, n, VNF2}.

```
    +---------------+  +----------------+  +---------------+
    | Data Center 1 |  |   Wide-Area    |  | Data Center 2 |
    | +----+       +----+      3       +----+       +----+ |
    | |VNF1|       |DCB1|-1   / \   5--|DCB2|       |VNF2| |
    | +----+       +----+  \ /   \ /   +----+       +----+ |
    |               |  |    2     4     |  |               |
    +---------------+  +----------------+  +---------------+
    SR Policy      SR Policy
    BSID m         BSID n
   {DCB1,n,VNF2} {1,2,3,4,5,DCB2}
```

Figure 2

Traffic-Engineered Path using BSID

In the above figure a single DCB is used to interconnect two domains. Similarly, in the case of two wide-area domains the DCB would be represented as an ABR or ASBR. In some single operator environments domains may be interconnected using adjacent ASBRs connected via a distinct physical link. In this scenario the procedures outlined above may be extended to incorporate the mechanisms used in Egress Peer Engineering (EPE) [I-D.ietf-spring-segment-routing-central-epe] to form a traffic-engineered path spanning distinct domains.

5.4.3.1. Traffic-Engineering and ECMP

Where the Interconnect controller is used to place SR policies, providing support for ECMP requires some consideration. An SR Policy is described with one or more segment-lists, end each of those segment-lists may or may not provide ECMP as a sum instruction and each SID itself may or may not support ECMP forwarding. When an individual SID is a BSID, an ECMP path may or may not also be nested within. The Interconnect controller may choose to place a path consisting entirely of non-ECMP-aware Adj-SIDs (each SID representing a single adjacency) such that the controller has explicit hop-by-hop knowledge of where that SR-TE LSP is routed. This is beneficial to allow the controller to take corrective action if the criteria that was used to initially select a particular link in a particular path subsequently changes. For example, if the latency of a link increases or a link becomes congested and a path should be rerouted. If ECMP-aware SIDs are used in the SR policy segment-list (including Node-SIDs, Adj-SIDs representing parallel links, and Anycast SIDs) SR routers are able to make autonomous decisions about where traffic is forwarded. As a result, it is not possible for the controller to fully understand the impact of a change in network state and react to it. With this in mind there are a number of approaches that could be adopted:

- If there is no requirement for the Interconnect controller to explicitly track path on a hop-by-hop basis, ECMP-aware SIDs may be used in the SR policy segment-list. This approach may require multiple [ELI, EL] pairs to be inserted at the ingress node; for example, above and below a BSID to provide entropy in multiple domains.

- If there is a requirement for the Interconnect controller to explicitly track paths on a hop-by-hop to provide the capability to reroute them based on changes in network state, SR policy segment-lists should be constructed of non-ECMP-aware Adj-SIDs.

- A hybrid approach that allows for a level of ECMP (at the headend) together with the ability for the Interconnect controller to explicitly track paths is to instantiate an SR policy consisting of a set of segment-lists, each containing non-ECMP-aware Adj-SIDs. Each segment-list will be assigned a weight to allow for ECMP or UCMP. This approach does however imply computation and programing of two paths instead of one.

- Another hybrid approach might work as follows. Redundant DCBs advertise an Anycast-SID 'A' into the data center, and also instantiate an SR policy with a segment-list consisting of non-ECMP-aware Adj-SIDs meeting the required connectivity and SLA. The BSID value of this SR policy 'B' must be common to both redundant DCBs, but the calculated paths are diverse. Indeed, multiple segment-lists could be used in this SR policy. A VNF could then instantiate an SR policy with a segment-list of {A, B} to achieve ECMP in the data center and TE in the wide-area network with the option of ECMP at the BSID anchor

5.5. Service Layer

The service layer is intended to deliver Layer 2 and/or Layer 3 VPN connectivity between network functions to create an overlay utilizing the routing and LSP underlay described in section 5.4. To do this the solution employs the EVPN and/or VPN-IPv4/IPv6 address families to exchange Layer 2 and Layer 3 Network Layer Reachability Information (NLRI). When these NLRI are exchanged between domains it is typical for the border router to set next-hop-self on advertised routes. With the proposed routing and LSP underlay however, this is not required and EVPN/VPN-IPv4/IPv6 routes should be passed end-to-end without transit routers modifying the next-hop attribute.

Section 5.4.2 describes the use of labeled unicast BGP to exchange inter-domain routes to establish a default forwarding path. Labeled-unicast BGP is used to exchange prefix reachability between service edge routers, with domain border routes imposing next-hop-self on routes advertised between domains. This provides a default inter-domain forwarding path and provides the required connectivity to establish inter-domain BGP sessions between service edges for the exchange of EVPN and/or VPN-IPv4/IPv6 NLRI. If route-reflection is used for the EVPN and/or VPN-IPv4/IPv6 address families within one or more domains, it may be desirable to create inter-domain BGP sessions between route-reflectors. In this case the peering addresses of the route-reflectors should also be exchanged between domains using labeled unicast BGP. This creates a connectivity model analogous to BGP/MPLS IP-VPN Inter-AS option C [RFC4364].

```
  +----------------+  +----------------+  +----------------+
  |     +-----+    |  |     +-----+    |  |     +-----+    |
+-----+ | RR |    +-----+   | RR |    +-----+   | RR |   +-----+
| NF |  +-----+   | DCI|    +-----+   | DCI|    +-----+  | NF |
+-----+           +-----+             +-----+            +-----+
  |     Domain    |  |     Domain    |  |     Domain    |
  +----------------+  +----------------+  +----------------+
  <-------> <-----> NHS <-- BGP-LU ---> NHS <-----> <------>
  <-------> <--------- EVPN/VPN-IPv4/v6 ----------> <------>
```

Figure 3

Inter-Domain Service Layer

EVPN and/or VPN-IPv4/v6 routes received from a peer in a different domain will contain a next-hop equivalent to the router that sourced the route. The next-hop of these routes can be resolved to labeled-unicast route (default forwarding path) or to an SR policy (traffic-engineered forwarding path) as appropriate to the service requirements. The exchange of EVPN and/or VPN-IPv4/IPv6 routes in this manner implies that Route-Distinguisher and Route-Target values remain intact end-to-end.

The use of end-to-end EVPN and/or VPN-IPv4/IPv6 address families without the imposition of next-hop-self at border routers complements the gateway-less transport layer architecture. It negates the requirement for midpoint service provisioning and as such provides the following benefits:

- Avoids the translation of MAC/IP EVPN routes to IP-VPN routes (and vice versa) that is typically associated with service interworking.

- Avoids instantiation of MAC-VRFs and IP-VPNs for each tenant resident in the DCB.

- Avoids provisioning of demarcation functions between the data center and wide-area network such as QoS, access-control, aggregation and isolation.

5.6. Service Differentiation

As discussed in section 5.4.3, the use of TE paths is a key capability of the NFIX solution framework described in this document. The Interconnect controller computes end-to-end TE paths between NFs and programs DC nodes, DCBs, ABR/ASBRs, via SR Policy, with the necessary label forwarding entries for each [headend, color, endpoint]. The collection of [headend, endpoint] pairs for the same color constitutes a logical network topology, where each topology satisfies a given SLA requirement.

The Interconnect controller discovers the endpoints associated to a given topology (color) upon the reception of EVPN or IPVPN routes advertised by the endpoint. The EVPN and IPVPN NLRIs are advertised by the endpoint nodes along with a color extended community which identifies the topology to which the owner of the NLRI belongs. At a coarse level all the EVPN/IPVPN routes of the same VPN can be advertised with the same color, and therefore a TE topology would be established on a per-VPN basis. At a more granular level IPVPN and especially EVPN provide a more granular way of coloring routes, that will allow the Interconnect controller to associate multiple topologies to the same VPN. For example:

- All the EVPN MAC/IP routes for a given VNF may be advertised with the same color. This would allow the Interconnect controller to associate topologies per VNF within the same VPN; that is, VNF1 could be blue (e.g., low-latency topology) and VNF2 could be green (e.g., high-throughput).

- The EVPN MAC/IP routes and Inclusive Multicast Ethernet Tag (IMET) route for VNF1 may be advertised with different colors, e.g., red and brown, respectively. This would allow the association of e.g., a low-latency topology for unicast traffic to VNF1 and best-effort topology for BUM traffic to VNF1.

- Each EVPN MAC/IP route or IP-Prefix route from a given VNF may be advertised with different color. This would allow the association of topologies at the host level or host route granularity.

5.7. Automated Service Activation

The automation of network and service connectivity for instantiation and mobility of virtual machines is a highly desirable attribute within data centers. Since this concerns service connectivity, it should be clear that this automation is relevant to virtual functions that belong to a service as opposed to a virtual network function that delivers services, such as a virtual PE router.

Within an SDN-enabled data center, a typical hierarchy from top to bottom would include a policy engine (or policy repository), one or more DC controllers, numerous hypervisors/container hosts that function as NVO endpoints, and finally the virtual machines(VMs)/ containers, which we'll refer to generically as virtualization hosts.

The mechanisms used to communicate between the policy engine and DC controller, and between the DC controller and hypervisor/container are not relevant here and as such they are not discussed further. What is important is the interface and information exchange between the Interconnect controller and the data center SDN functions:

- The Interconnect controller interfaces with the data center policy engine and publishes the available colors, where each color represents a topological service connectivity map that meets a set of constraints and SLA objectives. This interface is a straightforward API.

- The Interconnect controller interfaces with the DC controller to learn overlay routes. This interface is BGP and uses the EVPN Address Family.

With the above framework in place, automation of network and service connectivity can be implemented as follows:

- The virtualization host is turned-up. The NVO endpoint notifies the DC controller of the startup.

- The DC controller retrieves service information, IP addressing information, and service 'color' for the virtualization host from the policy engine. The DC controller subsequently programs the associated forwarding information on the virtualization host. Since the DC controller is now aware of MAC and IP address information for the virtualization host, it advertises that information as an EVPN MAC Advertisement Route into the overlay.

- The Interconnect controller receives the EVPN MAC Advertisement Route (potentially via a Route-Reflector) and correlates it with locally held service information and SLA requirements using Route Target and Color communities. If the relevant SR policies are not already in place to support the service requirements and logical connectivity, including any binding-SIDs, they are calculated and advertised to the relevant headends.

The same automated service activation principles can also be used to support the scenario where virtualization hosts are moved between hypervisors/container hosts for resourcing or other reasons. We refer to this simply as mobility. If a virtualization host is turned down the parent NVO endpoint notifies the DC controller, which in turn notifies the policy engine and withdraws any EVPN MAC Advertisement Routes. Thereafter all associated state is removed. When the virtualization host is turned up on a different hypervisor/container host, the automated service connectivity process outlined above is simply repeated.

5.8. Service Function Chaining

Service Function Chaining (SFC) defines an ordered set of abstract service functions and the subsequent steering of traffic through them. Packets are classified at ingress for processing by the required set of service functions (SFs) in an SFC-capable domain and are then forwarded through each SF in turn for processing. The ability to dynamically construct SFCs containing the relevant SFs in the right sequence is a key requirement for operators.

To enable flexible service function deployment models that support agile service insertion the NFIX architecture adopts the use of BGP as the control plane to distribute SFC information. The BGP control plane for Network Service Header (NSH) SFC [I-D.ietf-bess-nsh-bgp-control-plane] is used for this purpose and defines two route types;

the Service Function Instance Route (SFIR) and the Service Function Path Route (SFPR).

The SFIR is used to advertise the presence of a service function instance (SFI) as a function type (i.e. firewall, TCP optimizer) and is advertised by the node hosting that SFI. The SFIR is advertised together with a BGP Tunnel Encapsulation attribute containing details of how to reach that particular service function through the underlay network (i.e. IP address and encapsulation information).

The SFPRs contain service function path (SFP) information and one SFPR is originated for each SFP. Each SFPR contains the service path identifier (SPI) of the path, the sequence of service function types that make up the path (each of which has at least one instance advertised in an SFIR), and the service index (SI) for each listed service function to identify its position in the path.

Once a Classifier has determined which flows should be mapped to a given SFP, it imposes an NSH [RFC8300] on those packets, setting the SPI to that of the selected service path (advertised in an SFPR), and the SI to the first hop in the path. As NSH is encapsulation agnostic, the NSH encapsulated packet is then forwarded through the appropriate tunnel to reach the service function forwarder (SFF) supporting that service function instance (advertised in an SFIR). The SFF removes the tunnel encapsulation and forwards the packet with the NSH to the relevant SF based upon a lookup of the SPI/SI. When it is returned from the SF with a decremented SI value, the SFF forwards the packet to the next hop in the SFP using the tunnel information advertised by that SFI. This procedure is repeated until the last hop of the SFP is reached.

The use of the NSH in this manner allows for service chaining with topological and transport independence. It also allows for the deployment of SFIs in a condensed or dispersed fashion depending on operator preference or resource availability. Service function chains are built in their own overlay network and share a common underlay network, where that common underlay network is the NFIX fabric described in section 5.4. BGP updates containing an SFIR or SFPR are advertised in conjunction with one or more Route Targets (RTs), and each node in a service function overlay network is configured with one or more import RTs. As a result, nodes will only import routes that are applicable and that local policy dictates. This provides the ability to support multiple service function overlay networks or the construction of service function chains within L3VPN or EVPN services.

Although SFCs are constructed in a unidirectional manner, the BGP control plane for NSH SFC allows for the optional association of multiple paths (SFPRs). This provides the ability to construct a bidirectional service function chain in the presence of multiple equal-cost paths between source and destination to avoid problems that SFs may suffer with traffic asymmetry.

The proposed SFC model can be considered decoupled in that the use of SR as a transport between SFFs is completely independent of the use of NSH to define the SFC. That is, it uses an NSH-based SFC and SR is just one of many encapsulations that could be used between SFFs. A similar more integrated approach proposes encoding a service function as a segment so that an SFC can be constructed as a segment-list. In this case it can be considered an SR-based SFC with an NSH-based service plane since the SF is unaware of the presence of the SR. Functionally both approaches are very similar and as such both could be adopted and could work in parallel. Construction of SFCs based purely on SR (SF is SR-aware) are not considered at this time.

5.9. Stability and Availability

Any network architecture should have the capability to self-restore following the failure of a network element. The time to reconverge following the failure needs to be minimal to avoid evident disruptions in service. This section discusses protection mechanisms that are available for use and their applicability to the proposed architecture.

5.9.1. IGP Reconvergence

Within the construct of an IGP topology the Topology Independent Loop Free Alternate (TI-LFA) [I-D.ietf-rtgwg-segment-routing-ti-lfa] can be used to provide a local repair mechanism that offers both link and node protection.

TI-LFA is a repair mechanism, and as such it is reactive and initially needs to detect a given failure. To provide fast failure detection the Bidirectional Forwarding Mechanism (BFD) is used. Consideration needs to be given to the restoration capabilities of the underlying transmission when deciding values for message intervals and multipliers to avoid race conditions, but failure detection in the order of 50 milliseconds can reasonably be anticipated. Where Link Aggregation Groups (LAG) are used, micro-BFD [RFC7130] can be used to similar effect. Indeed, to allow for potential incremental growth in capacity it is not uncommon for operators to provision all network links as LAG and use micro-BFD from the outset.

5.9.2. Data Center Reconvergence

Clos fabrics are extremely common within data centers, and fundamental to a Clos fabric is the ability to load-balance using

Equal Cost Multipath (ECMP). The number of ECMP paths will vary dependent on the number of devices in the parent tier but will never be less than two for redundancy purposes with traffic hashed over the available paths. In this scenario the availability of a backup path in the event of failure is implicit. Commonly within the DC, rather than computing protect paths (like LFA), techniques such as 'fast rehash' are often utilized. In this particular case, the failed next-hop is removed from the multi-path forwarding data structure and traffic is then rehashed over the remaining active paths.

In BGP-only data centers this relies on the implementation of BGP multipath. As network elements in the lower tier of a Clos fabric will frequently belong to different ASNs, this includes the ability to load-balance to a prefix with different AS_PATH attribute values while having the same AS_PATH length; sometimes referred to as 'multipath relax' or 'multipath multiple-AS' [RFC7938].

Failure detection relies upon declaring a BGP session down and removing any prefixes learnt over that session as soon as the link is declared down. As links between network elements predominantly use direct point-to-point fiber, a link failure should be detected within milliseconds. BFD is also commonly used to detect IP layer failures.

5.9.3. Exchange of Inter-Domain Routes

Labeled unicast BGP together with SR Prefix-SID extensions are used to exchange PNF and/or VNF endpoints between domains to create end-to-end connectivity without TE. When advertising between domains we assume that a given BGP prefix is advertised by at least two border routers (DCBs, ABRs, ASBRs) making prefixes reachable via at least two next-hops.

BGP Prefix Independent Convergence (PIC) [I-D.ietf-rtgwg-bgp-pic] allows failover to a pre-computed and pre-installed secondary next-hop when the primary next-hop fails and is independent of the number of destination prefixes that are affected by the failure. When the primary BGP next-hop fails, it should be clear that BGP PIC depends on the availability o f a secondary next-hop in the Pathlist. To ensure that multiple paths to the same destination are visible the BGP ADD-PATH [RFC7911] can be used to allow for advertisement of multiple paths for the same address prefix. Dual-homed EVPN/IP-VPN prefixes also have the alternative option of allocating different Route-Distinguishers (RDs). To trigger the switch from primary to secondary next-hop PIC needs to detect the failure and many implementations support 'next-hop tracking' for this purpose. Next-hop tracking monitors the routing-table and if the next-hop prefix is removed will immediately invalidate all BGP prefixes learnt through that next-hop. In the absence of next-hop tracking, multihop BFD [RFC5883] could optionally be used as a fast failure detection mechanism.

5.9.4. Controller Redundancy

With the Interconnect controller providing an integral part of the networks' capabilities a redundant controller design is clearly prudent. To this end we can consider both availability and redundancy. Availability refers to the survivability of a single controller system in a failure scenario. A common strategy for increasing the availability of a single controller system is to build the system in a high-availability cluster such that it becomes a confederation of redundant constituent parts as opposed to a single monolithic system. Should a single part fail, the system can still survive without the requirement to failover to a standby controller system. Methods for detection of a failure of one or more member parts of the cluster are implementation specific.

To provide contingency for a complete system failure a geo-redundant standby controller system is required. When redundant controllers are deployed a coherent strategy is needed that provides a master/ standby election mechanism, the ability to propagate the outcome of that election to network elements as required, an inter-system failure detection mechanism, and the ability to synchronize state across both systems such that the standby controller is fully aware of current state should it need to transition to master controller.

Master/standby election, state synchronisation, and failure detection between geo-redundant sites can largely be considered a local implementation matter. The requirement to propagate the outcome of the master/standby election to network elements depends on a) the mechanism that is used to instantiate SR policies, and b) whether the SR policies are controller-initiated or headend-initiated, and these are discussed in the following sub-sections. In either scenario, state of SR policies should be advertised northbound to both master/standby controllers using either PCEP LSP State Report messages or SR policy extensions to BGP link-state [I-D.ietf-idr-te-lsp-distribution].

5.9.4.1. SR Policy Initiator

Controller-initiated SR policies are suited for auto-creation of tunnels based on service route discovery and policy-driven route/ flow programming and are ephemeral. Headend-initiated tunnels allow for permanent configuration state to be held on the headend and are suitable for static services that are not subject to dynamic changes. If all SR policies are controller-initiated, it negates the requirement to propagate the outcome of the master/standby election to network elements. This is because headends have no requirement for unsolicited requests to a controller, and therefore have no requirement to know which controller is master and which one is standby. A headend may respond to a message from a controller, but it is not unsolicited.

If some or all SR policies are headend-initiated, then the requirement to propagate the outcome of the master/standby election exists. This is further discussed in the following sub-section.

5.9.4.2. SR Policy Instantiation Mechanism

While candidate paths of SR policies may be provided using BGP, PCEP, Netconf, or local policy/configuration, this document primarily considers the use of PCEP or BGP.

When PCEP [RFC5440][RFC8231][RFC8281] is used for instantiation of candidate paths of SR policies [I-D.barth-pce-segment-routing-policy-cp] every headend/PCC should establish a PCEP session with the master and standby controllers. To signal standby state to the PCC the standby controller may use a PCEP Notification message to set the PCEP session into overload state. While in this overload state the standby controller will accept path computation LSP state report (PCRpt) messages without delegation but will reject path computation requests (PCReq) and any path computation reports (PCRpt) with the delegation bit set. Further, the standby controller will not path computation originate initiate messages (PCInit) or path computation update request messages (PCUpd). In the event of the failure of the master controller, the standby controller will transition to active and remove the PCEP overload state. Following expiration of the PCEP redelegation timeout at the PCC any LSPs will be redelegated to the newly transitioned active controller. LSP state is not impacted unless redelegation is not possible before the state timeout interval expires.

When BGP is used for instantiation of SR policies every headend should establish a BGP session with the master and standby controller capable of exchanging SR TE Policy SAFI. Candidate paths of SR policies are advertised only by the active controller. If the master controller should experience a failure, then SR policies learnt from that controller may be removed before they are re-advertised by the standby (or newly-active) controller. To minimize this possibility BGP speakers that advertise and instantiate SR policies can implement Long Lived Graceful Retart (LLGR) [I-D.ietf-idr-long-lived-gr], also known as BGP persistence, to retain existing routes treated as least-preferred until the new route arrives. In the absence of LLGR, two other alternatives are possible:

- *Provide a static backup SR policy.

- *Fallback to the default forwarding path.

5.9.5. Path and Segment Liveliness

When using traffic-engineered SR paths only the ingress router holds any state. The exception here is where BSIDs are used, which also implies some state is maintained at the BSID anchor. As there is no control plane set-up, it follows that there is no feedback loop from transit nodes of the path to notify the headend when a non-adjacent point of the SR path fails. The Interconnect controller however is aware of all paths that are impacted by a given network failure and should take the appropriate action. This action could include withdrawing an SR policy if a suitable candidate path is already in place, or simply sending a new SR policy with a different segment-list and a higher preference value assigned to it.

Verification of data plane liveliness is the responsibility of the path headend. A given SR policy may be associated with multiple candidate paths and for the sake of clarity, we'll assume two for redundancy purposes (which can be diversely routed). Verification of the liveliness of these paths can be achieved using seamless BFD (S-BFD)[RFC7880], which provides an in-band failure detection mechanism capable of detecting failure in the order of tens of milliseconds. Upon failure of the active path, failover to a secondary candidate path can be activated at the path headend. Details of the actual failover and revert mechanisms are a local implementation matter.

S-BFD provides a fast and scalable failure detection mechanism but is unlikely to be implemented in many VNFs given their inability to offload the process to purpose-built hardware. In the absence of an active failure detection mechanism such as S-BFD the failover from active path to secondary candidate path can be triggered using continuous path validity checks. One of the criteria that a candidate path uses to determine its validity is the ability to perform path resolution for the first SID to one or more outgoing interface(s) and next-hop(s). From the perspective of the VNF headend the first SID in the segment-list will very likely be the DCB (as BSID anchor) but could equally be another Prefix-SID hop within the data center. Should this segment experience a non-recoverable failure, the headend will be unable to resolve the first SID and the path will be considered invalid. This will trigger a failover action to a secondary candidate path.

Injection of S-BFD packets is not just constrained to the source of an end-to-end LSP. When an S-BFD packet is injected into an SR policy path it is encapsulated with the label stack of the associated segment-list. It is possible therefore to run S-BFD from a BSID anchor for just that section of the end-to-end path (for example, from DCB to DCB). This allows a BSID anchor to detect failure of a path and take corrective action, while maintaining opacity between domains.

5.10. Scalability

There are many aspects to consider regarding scalability of the NFIX architecture. The building blocks of NFIX are standards-based technologies individually designed to scale for internet provider networks. When combined they provide a flexible and scalable solution:

*BGP has been proven to scale and operate with millions of routes being exchanged. Specifically, BGP labeled unicast has been deployed and proven to scale in existing seamless-MPLS networks.

*By placing forwarding instructions in the header of a packet, segment routing reduces the amount of state required in the network allowing the scale of greater number of transport tunnels. This aids in the feasibility of the NFIX architecture to permit the automated aspects of SR policy creation without having an impact on the state in the core of the network.

*The choice of utilizing native SR-MPLS or SR over IP in the data center continues to permit horizontal scaling without introducing new state inside of the data center fabric while still permitting seamless end to end path forwarding integration.

*BSIDs play a key role in the NFIX architecture as their use provides the ability to traffic-engineer across large network topologies consisting of many hops regardless of hardware capability at the headend. From a scalability perspective the use of BSIDs facilitates better scale due to the fact that detailed information about the SR paths in a domain has been abstracted and localized to the BSID anchor point only. When BSIDs are re-used amongst one or many headends they reduce the amount of path calculation and updates required at network edges while still providing seamless end to end path forwarding.

*The architecture of NFIX continues to use an independent DC controller. This allows continued independent scaling of data center management in both policy and local forwarding functions, while off-loading the end-to-end optimal path placement and automation to the Interconnect controller. The optimal path placement is already a scalable function provided in a PCE architecture. The Interconnect controller must compute paths, but it is not burdened by the management of virtual entity lifecycle and associated forwarding policies.

It must be acknowledged that with the amalgamation of the technology building blocks and the automation required by NFIX, there is an additional burden on the Interconnect controller. The scaling considerations are dependent on many variables, but an implementation of a Interconnect controller shares many overlapping traits and scaling concerns as PCE, where the controller and PCE both must:

- Discover and listen to topological state changes of the IP/MPLS topology.

- Compute traffic-engineered intra and inter domain paths across large service provider topologies.

- Synchronize, track and update thousands of LSPs to network devices upon network state changes.

Both entail topologies that contain tens of thousands of nodes and links. The Interconnect controller in an NFIX architecture takes on the additional role of becoming end to end service aware and discovering data center entities that were traditionally excluded from a controllers scope. Although not exhaustive, an NFIX Interconnect controller is impacted by some of the following:

- The number of individual services, the number of endpoints that may exist in each service, the distribution of endpoints in a virtualized environment, and how many data centers may exist. Medium or large sized data centers may be capable to host more virtual endpoints per host, but with the move to smaller edge-clouds the number of headends that require inter-connectivity increases compared to the density of localized routing in a centralized data center model. The outcome has an impact on the number of headend devices which may require tunnel management by the Interconnect controller.

- Assuming a given BSID satisfies SLA, the ability to re-use BSIDs across multiple services reduces the number of paths to track and manage. However, the number of color or unique SLA definitions, and criteria such as bandwidth constraints impacts WAN traffic distribution requirements. As BSIDs play a key role for VNF connectivity, this potentially increases the number of BSID paths required to permit appropriate traffic distribution. This also impacts the number of tunnels which may be re-used on a given headend for different services.

- The frequency of virtualized hosts being created and destroyed and the general activity within a given service. The controller must analyze, track, and correlate the activity of relevant BGP routes to track addition and removal of service host or host subnets, and determine whether new SR policies should be instantiated, or stale unused SR policies should be removed from the network.

- The choice of SR instantiation mechanism impacts the number of communication sessions the controller may require. For example, the BGP based mechanism may only require a small number of sessions to route reflectors, whereas PCEP may require a connection to every possible leaf in the network and any BSID anchors.

- The number of hops within one or many WAN domains may affect the number of BSIDs required to provide transit for VNF/PNF, PNF/PNF, or VNF/VNF inter-connectivity.

- Relative to traditional WAN topologies, traditional data centers are generally topologically denser in node and link connectivity which is required to be discovered by the Interconnect controller, resulting in a much larger, dense link-state database on the Interconnect controller.

5.10.1. Asymmetric Model B for VPN Families

With the instantiation of multiple TE paths between any two VNFs in the NFIX network, the number of SR Policy (remote endpoint, color) routes, BSIDs and labels to support on VNFs becomes a choke point in the architecture. The fact that some VNFs are limited in terms of forwarding resources makes this aspect an important scale issue.

As an example, if VNF1 and VNF2 in Figure 1 are associated to multiple topologies 1..n, the Interconnect controller will instantiate n TE paths in VNF1 to reach VNF2:

[VNF1,color-1,VNF2] --> BSID 1

[VNF1,color-2,VNF2] --> BSID 2

...

[VNF1,color-n,VNF2] --> BSID n

Similarly, m TE paths may be instantiated on VNF1 to reach VNF3, another p TE paths to reach VNF4, and so on for all the VNFs that VNF1 needs to communicate with in DC2. As it can be observed, the number of forwarding resources to be instantiated on VNF1 may significantly grow with the number of remote [endpoint, color] pairs, compared with a best-effort architecture in which the number forwarding resources in VNF1 grows with the number of endpoints only.

This scale issue on the VNFs can be relieved by the use of an asymmetric model B service layer. The concept is illustrated in Figure 3.

```
                                             +------------+
       <-------------------------------------|    WAN     |
       |  SR Policy      +-------------------| Controller |
       |  BSID m         |   SR Policy       +------------+
       v  {DCI1,n,DCI2}  v   BSID n
                             {1,2,3,4,5,DCI2}
      +----------------+  +----------------+  +----------------+
      |     +----+     |  |                |  |     +----+     |
    +----+  | RR |    +----+             +----+    | RR |   +----+
    |VNF1|  +----+    |DCI1|             |DCI2|    +----+   |VNF2|
    +----+            +----+             +----+             +----+
      |       DC1      |  |       WAN      |  |       DC2      |
      +----------------+  +----------------+  +----------------+

      <-------- <-------------------------- NHS <------- <------
                           EVPN/VPN-IPv4/v6(colored)

      +------------------------------------>    +------------->
                TE path to DCI2                ECMP path to VNF2
            (BSID to segment-list
             expansion on DCI1)
```

Figure 4

Asymmetric Model B Service Layer

Consider the different n topologies needed between VNF1 and VNF2 are really only relevant to the different TE paths that exist in the WAN. The WAN is the domain in the network where there can be significant differences in latency, throughput or packet loss depending on the sequence of nodes and links the traffic goes through. Based on that assumption, for traffic from VNF1 to DCB2 in Figure 4, traffic from DCB2 to VNF2 can simply take an ECMP path. In this case an asymmetric model B Service layer can significantly relieve the scale pressure on VNF1.

From a service layer perspective, the NFIX architecture described up to now can be considered 'symmetric', meaning that the EVPN/IPVPN advertisements from e.g., VNF2 in Figure 2, are received on VNF1 with the next-hop of VNF2, and vice versa for VNF1's routes on VNF2. SR Policies to each VNF2 [endpoint, color] are then required on the VNF1.

In the 'asymmetric' service design illustrated in Figure 4, VNF2's EVPN/IPVPN routes are received on VNF1 with the next-hop of DCB2, and VNF1's routes are received on VNF2 with next-hop of DCB1. Now SR policies instantiated on VNFs can be reduced to only the number of TE paths required to reach the remote DCB. For example, considering n topologies, in a symmetric model VNF1 has to be instantiated with n SR policy paths per remote VNF in DC2, whereas in the asymmetric model of Figure 4, VNF1 only requires n SR policy paths per DC, i.e., to DCB2.

Asymmetric model B is a simple design choice that only requires the ability (on the DCB nodes) to set next-hop-self on the EVPN/IPVPN routes advertised to the WAN neighbors and not do next-hop-self for routes advertised to the DC neighbors. With this option, the Interconnect controller only needs to establish TE paths from VNFs to remote DCBs, as opposed to VNFs to remote VNFs.

6. Illustration of Use

For the purpose of illustration, this section provides some examples of how different end-to-end tunnels are instantiated (including the relevant protocols, SID values/label stacks etc.) and how services are then overlaid onto those LSPs.

6.1. Reference Topology

The following network diagram illustrates the reference network topology that is used for illustration purposes in this section. Within the data centers leaf and spine network elements may be present but are not shown for the purpose of clarity.

```
                 +----------+
                 |Controller|
                 +----------+
                   /  |  \
          +----+          +----+          +----+     +----+
  ~ ~ ~ ~ | R1 |----------| R2 |----------| R3 |-----|AGN1| ~ ~ ~ ~
  ~       +----+          +----+          +----+     +----+       ~
  ~   DC1    |                            /  |         |    DC2   ~
+----+       |      L=5   +----+   L=5   /   |       +----+    +----+
| Sn |       |    +-------| R4 |--------+    |       |AGN2|    | Dn |
+----+       |   /  M=20  +----+  M=20       |       +----+    +----+
  ~          |  /                            |         |          ~
  ~       +----+     +----+    +----+     +----+     +----+       ~
  ~ ~ ~ ~ | R5 |-----| R6 |----| R7 |-----| R8 |-----|AGN3| ~ ~ ~ ~
          +----+     +----+    +----+     +----+     +----+
```

Figure 5

Reference Topology

The following applies to the reference topology in figure 5:

- *Data center 1 and data center 2 both run BGP/SR. Both data centers run leaf/spine topologies, which are not shown for the purpose of clarity.
- *R1 and R5 function as data center border routers for DC 1. AGN1 and AGN3 function as data center border routers for DC 2.
- *Routers R1 through R8 form an independent ISIS-OSPF/SR instance.
- *Routers R3, R8, AGN1, AGN2, and AGN2 form an independent ISIS-OSPF/SR instance.
- *All IGP link metrics within the wide area network are metric 10 except for links R5-R4 and R4-R3 which are both metric 20.
- *All links have a unidirectional latency of 10 milliseconds except for links R5-R4 and R4-R3 which both have a unidirectional latency of 5 milliseconds.
- *Source 'Sn' and destination 'Dn' represent one or more network functions.

6.2. PNF to PNF Connectivity

The first example demonstrates the simplest form of connectivity; PNF to PNF. The example illustrates the instantiation of a unidirectional TE path from R1 to AGN2 and its consumption by an EVPN service. The service has a requirement for high-throughput with no strict latency requirements. These service requirements are catalogued and represented using the color blue.

- An EVPN service is provisioned at R1 and AGN2.

- The Interconnect controller computes the path from R1 to AGN2 and calculates that the optimal path based on the service requirements and overall network optimization is R1-R5-R6-R7-R8-AGN3-AGN2. The segment-list to represent the calculated path could be constructed in numerous ways. It could be strict hops represented by a series of Adj-SIDs. It could be loose hops using ECMP-aware Node-SIDs, for example {R7, AGN2}, or it could be a combination of both Node-SIDs and Adj-SIDs. Equally, BSIDs could be used to reduce the number of labels that need to be imposed at the headend. In this example, strict Adj-SID hops are used with a BSID at the area border router R8, but this should not be interpreted as the only way a path and segment-list can be represented.

- The Interconnect controller advertises a BGP SR Policy to R8 with BSID 1000, and a segment-list containing segments {AGN3, AGN2}.

- The Interconnect controller advertises a BGP SR Policy to R1 with BSID 1001, and a segment-list containing segments {R5, R6, R7, R8, 1000}. The policy is identified using the tuple [headed = R1, color = blue, endpoint = AGN2].

- AGN2 advertises an EVPN MAC Advertisement Route for MAC M1, which is learned by R1. The route has a next-hop of AGN2, an MPLS label of L1, and it carries a color extended community with the value blue.

- R1 has a valid SR policy [color = blue, next-hop = AGN2] with segment-list {R5, R6, R7, R8, 1000}. R1 therefore associates the MAC address M1 with that policy and programs the relevant information into the forwarding path.

- The Interconnect controller also learns the EVPN MAC Route advertised by AGN2. The purpose of this is two-fold. It allows the controller to correlate the service overlay with the underlying transport LSPs, thus creating a service connectivity map. It also allows the controller to dynamically create LSPs based upon service requirements if they do not already exist, or to optimize them if network conditions change.

6.3. VNF to PNF Connectivity

The next example demonstrates VNF to PNF connectivity and illustrates the instantiation of a unidirectional TE path from S1 to AGN2. The path is consumed by an IP-VPN service that has a basic set of service requirements and as such simply uses IGP metric as a path computation objective. These basic service requirements are cataloged and represented using the color red.

In this example S1 is a VNF with full IP routing and MPLS capability that interfaces to the data center underlay/overlay and serves as the NVO tunnel endpoint.

- An IP-VPN service is provisioned at S1 and AGN2.

- The Interconnect controller computes the path from S1 to AGN2 and calculates that the optimal path based on IGP metric is R1-R2-R3-AGN1-AGN2.

- The Interconnect controller advertises a BGP SR Policy to R1 with BSID 1002, and a segment-list containing segments {R2, R3, AGN1, AGN2}.

- The Interconnect controller advertises a BGP SR Policy to S1 with BSID 1003, and a segment-list containing segments {R1, 1002}. The policy is identified using the tuple [headend = S1, color = red, endpoint = AGN2].

- Source S1 learns an VPN-IPv4 route for prefix P1, next-hop AGN2. The route has an VPN label of L1, and it carries a color extended community with value red.

- S1 has a valid SR policy [color = red, endpoint = AGN2] with segment-list {R1, 1002} and BSID 1003. S1 therefore associates the VPN-IPv4 prefix P1 with that policy and programs the relevant information into the forwarding path.

- As in the previous example the Interconnect controller also learns the VPN-IPv4 route advertised by AGN2 in order to correlate the service overlay with the underlying transport LSPs, creating or optimizing them as required.

6.4. VNF to VNF Connectivity

The last example demonstrates VNF to VNF connectivity and illustrates the instantiation of a unidirectional TE path from S2 to D2. The path is consumed by an EVPN service that requires low latency as a service requirement and as such uses latency as a path computation objective. This service requirement is cataloged and represented using the color green.

In this example S2 is a VNF that has no routing capability. It is hosted by hypervisor H1 that in turn has an interface to a DC controller through which forwarding instructions are programmed. H1 serves as the NVO tunnel endpoint and overlay next-hop.

D2 is a VNF with partial routing capability that is connected to a leaf switch L1. L1 connects to underlay/overlay in data center 2 and serves as the NVO tunnel endpoint for D2. L1 advertises BGP Prefix-SID 9001 into the underlay.

- The relevant details of the EVPN service are entered in the data center policy engines within data center 1 and 2.

- Source S2 is turned-up. Hypervisor H1 notifies its parent DC controller, which in turn retrieves the service (EVPN) information, color, IP and MAC information from the policy engine and subsequently programs the associated forwarding entries onto S2. The DC controller also dynamically advertises an EVPN MAC Advertisement Route for S2's IP and MAC into the overlay with next-hop H1. (This would trigger the return path set-up between L1 and H2 not covered in this example.)

- The DC controller in data center 1 learns an EVPN MAC Advertisement Route for D2, MAC M, next-nop L1. The route has an MPLS label of L2, and it carries a color extended community with the value green.

- The Interconnect controller computes the path between H1 and L1 and calculates that the optimal path based on latency is R5-R4-R3-AGN1.

- The Interconnect controller advertises a BGP SR Policy to R5 with BSID 1004, and a segment-list containing segments {R4, R3, AGN1}.

- The Interconnect controller advertises a BGP SR Policy to the DC controller in data center 1 with BSID 1005 and a segment-list containing segments {R5, 1004, 9001}. The policy is identified using the tuple [headend = H1, color = green, endpoint = L1].

- The DC controller in data center 1 has a valid SR policy [color = green, endpoint = L1] with segment-list {R5, 1004, 9001} and BSID 1005. The controller therefore associates the MAC Advertisement Route with that policy, and programs the associated forwarding rules into S2.

- As in the previous example the Interconnect controller also learns the MAC Advertisement Route advertised by D2 in order to correlate the service overlay with the underlying transport LSPs, creating or optimizing them as required.

7. Conclusions

The NFIX architecture provides an evolutionary path to a unified network fabric. It uses the base constructs of seamless-MPLS and adds end-to-end LSPs capable of delivering against SLAs, seamless data center interconnect, service differentiation, service function chaining, and a Layer-2/Layer-3 infrastructure capable of interconnecting PNF-to-PNF, PNF-to-VNF, and VNF-to-VNF.

NFIX establishes a dynamic, seamless, and automated connectivity model that overcomes the operational barriers and interworking issues between data centers and the wide-area network and delivers the following using standards-based protocols:

- A unified routing control plane: Multiprotocol BGP (MP-BGP) to acquire inter-domain NLRI from the IP/MPLS underlay and the virtualized IP-VPN/EVPN service overlay.

- A unified forwarding control plane: SR provides dynamic service tunnels with fast restoration options to meet deterministic bandwidth, latency and path diversity constraints. SR utilizes the appropriate data path encapsulation for seamless, end-to-end connectivity between distributed edge and core data centers across the wide-area network.

- Service Function Chaining: Leverage SFC extensions for BGP and segment routing to interconnect network and service functions into SFPs, with support for various data path implementations.

- Service Differentiation: Provide a framework that allows for construction of logical end-to-end networks with differentiated logical topologies and/or constraints through use of SR policies and coloring.

- Automation: Facilitates automation of service provisioning and avoids heavy service interworking at DCBs.

NFIX is deployable on existing data center and wide-area network infrastructures and allows the underlying data forwarding plane to evolve with minimal impact on the services plane.

8. Security Considerations

The NFIX architecture based on SR-MPLS is subject to the same security concerns as any MPLS network. No new protocols are introduced, hence security issues of the protocols encompassed by this architecture are addressed within the relevant individual standards documents. It is recommended that the security framework for MPLS and GMPLS networks defined in [RFC5920] are adhered to. Although [RFC5920] focuses on the use of RSVP-TE and LDP control plane, the practices and procedures are extendable to an SR-MPLS domain.

The NFIX architecture makes extensive use of Multiprotocol BGP, and it is recommended that the TCP Authentication Option (TCP-AO) [RFC5925] is used to protect the integrity of long-lived BGP sessions and any other TCP-based protocols.

Where PCEP is used between controller and path headend the use of PCEPS [RFC8253] is recommended to provide confidentiality to PCEP communication using Transport Layer Security (TLS).

9. Acknowledgements

The authors would like to acknowledge Mustapha Aissaoui, Wim Henderickx, and Gunter Van de Velde.

10. Contributors

The following people contributed to the content of this document and should be considered co-authors.

Juan Rodriguez
Nokia
United States of America

Email: juan.rodriguez@nokia.com

Jorge Rabadan
Nokia
United States of America

Email: jorge.rabadan@nokia.com

Nick Morris
Verizon
United States of America

Email: nicklous.morris@verizonwireless.com

Eddie Leyton
Verizon
United States of America

Email: edward.leyton@verizonwireless.com

Figure 6

11. IANA Considerations

This memo does not include any requests to IANA for allocation.

12. References

12.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997, <http://xml.resource.org/public/rfc/html/rfc2119.html>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

12.2. Informative References

[I-D.ietf-nvo3-geneve] Gross, J., Ganga, I., and T. Sridhar, "Geneve: Generic Network Virtualization Encapsulation", Work in Progress, Internet-Draft, draft-ietf-nvo3-geneve-16, 7 March 2020, <https://www.ietf.org/archive/id/draft-ietf-nvo3-geneve-16.txt>.

[I-D.ietf-mpls-seamless-mpls] Leymann, N., Decraene, B., Filsfils, C., Konstantynowicz, M., and D. Steinberg, "Seamless MPLS Architecture", Work in Progress, Internet-Draft, draft-ietf-mpls-seamless-mpls-07, 28 June 2014, <https://www.ietf.org/archive/id/draft-ietf-mpls-seamless-mpls-07.txt>.

[I-D.ietf-bess-evpn-ipvpn-interworking] Rabadan, J., Sajassi, A., Rosen, E., Drake, J., Lin, W., Uttaro, J., and A. Simpson, "EVPN Interworking with IPVPN", Work in Progress, Internet-Draft, draft-ietf-bess-evpn-ipvpn-interworking-06, 22 September 2021, <https://www.ietf.org/archive/id/draft-ietf-bess-evpn-ipvpn-interworking-06.txt>.

[I-D.ietf-spring-segment-routing-policy] Filsfils, C., Talaulikar, K., Voyer, D., Bogdanov, A., and P. Mattes, "Segment Routing Policy Architecture", Work in Progress, Internet-Draft, draft-ietf-spring-segment-routing-policy-14, 25 October 2021, <https://www.ietf.org/archive/id/draft-ietf-spring-segment-routing-policy-14.txt>.

[I-D.ietf-rtgwg-segment-routing-ti-lfa] Litkowski, S., Bashandy, A., Filsfils, C., Francois, P., Decraene, B., and D. Voyer, "Topology Independent Fast

Reroute using Segment Routing", Work in Progress, Internet-Draft, draft-ietf-rtgwg-segment-routing-ti-lfa-07, 29 June 2021, <https://www.ietf.org/archive/id/draft-ietf-rtgwg-segment-routing-ti-lfa-07.txt>.

[I-D.ietf-bess-nsh-bgp-control-plane] Farrel, A., Drake, J., Rosen, E., Uttaro, J., and L. Jalil, "BGP Control Plane for the Network Service Header in Service Function Chaining", Work in Progress, Internet-Draft, draft-ietf-bess-nsh-bgp-control-plane-18, 21 August 2020, <https://www.ietf.org/archive/id/draft-ietf-bess-nsh-bgp-control-plane-18.txt>.

[I-D.ietf-idr-te-lsp-distribution] Previdi, S., Talaulikar, K., Dong, J., Chen, M., Gredler, H., and J. Tantsura, "Distribution of Traffic Engineering (TE) Policies and State using BGP-LS", Work in Progress, Internet-Draft, draft-ietf-idr-te-lsp-distribution-16, 22 October 2021, <https://www.ietf.org/archive/id/draft-ietf-idr-te-lsp-distribution-16.txt>.

[I-D.barth-pce-segment-routing-policy-cp] Koldychev, M., Sivabalan, S., Barth, C., Peng, S., and H. Bidgoli, "PCEP extension to support Segment Routing Policy Candidate Paths", Work in Progress, Internet-Draft, draft-barth-pce-segment-routing-policy-cp-06, 2 June 2020, <https://www.ietf.org/archive/id/draft-barth-pce-segment-routing-policy-cp-06.txt>.

[I-D.filsfils-spring-sr-policy-considerations] Filsfils, C., Talaulikar, K., Krol, P., Horneffer, M., and P. Mattes, "SR Policy Implementation and Deployment Considerations", Work in Progress, Internet-Draft, draft-filsfils-spring-sr-policy-considerations-08, 22 October 2021, <https://www.ietf.org/archive/id/draft-filsfils-spring-sr-policy-considerations-08.txt>.

[I-D.ietf-rtgwg-bgp-pic] Bashandy, A., Filsfils, C., and P. Mohapatra, "BGP Prefix Independent Convergence", Work in Progress, Internet-Draft, draft-ietf-rtgwg-bgp-pic-17, 12 October 2021, <https://www.ietf.org/archive/id/draft-ietf-rtgwg-bgp-pic-17.txt>.

[I-D.ietf-isis-mpls-elc] Xu, X., Kini, S., Psenak, P., Filsfils, C., Litkowski, S., and M. Bocci, "Signaling Entropy Label Capability and Entropy Readable Label Depth Using IS-IS", Work in Progress, Internet-Draft, draft-ietf-isis-mpls-elc-13, 28 May 2020, <https://www.ietf.org/archive/id/draft-ietf-isis-mpls-elc-13.txt>.

[I-D.ietf-ospf-mpls-elc] Xu, X., Kini, S., Psenak, P., Filsfils, C., Litkowski, S., and M. Bocci, "Signaling Entropy Label Capability and Entropy Readable Label Depth Using OSPF", Work in Progress, Internet-Draft, draft-ietf-ospf-mpls-elc-15, 1 June 2020, <https://www.ietf.org/archive/id/draft-ietf-ospf-mpls-elc-15.txt>.

[I-D.ietf-idr-next-hop-capability] Decraene, B., Kompella, K., and W. Henderickx, "BGP Next-Hop dependent capabilities", Work in Progress, Internet-Draft, draft-ietf-idr-next-hop-capability-07, 8 December 2021, <https://www.ietf.org/archive/id/draft-ietf-idr-next-hop-capability-07.txt>.

[I-D.ietf-spring-segment-routing-central-epe] Filsfils, C., Previdi, S., Dawra, G., Aries, E., and D. Afanasiev, "Segment Routing Centralized BGP Egress Peer Engineering", Work in Progress, Internet-Draft, draft-ietf-spring-segment-routing-central-epe-10, 21 December 2017, <https://www.ietf.org/archive/id/draft-ietf-spring-segment-routing-central-epe-10.txt>.

[I-D.ietf-idr-long-lived-gr] Uttaro, J., Chen, E., Decraene, B., and J. G. Scudder, "Support for Long-lived BGP Graceful Restart", Work in Progress, Internet-Draft, draft-ietf-idr-long-lived-gr-00, 5 September 2019, <https://www.ietf.org/archive/id/draft-ietf-idr-long-lived-gr-00.txt>.

[RFC7938] Lapukhov, P., Premji, A., and J. Mitchell, Ed., "Use of BGP for Routing in Large-Scale Data Centers", RFC 7938, DOI 10.17487/RFC7938, August 2016, <https://www.rfc-editor.org/info/rfc7938>.

[RFC7752] Gredler, H., Ed., Medved, J., Previdi, S., Farrel, A., and S. Ray, "North-Bound Distribution of Link-State and Traffic Engineering (TE) Information Using BGP", RFC 7752, DOI 10.17487/RFC7752, March 2016, <https://www.rfc-editor.org/info/rfc7752>.

[RFC8277] Rosen, E., "Using BGP to Bind MPLS Labels to Address Prefixes", RFC 8277, DOI 10.17487/RFC8277, October 2017, <https://www.rfc-editor.org/info/rfc8277>.

[RFC8667] Previdi, S., Ed., Ginsberg, L., Ed., Filsfils, C., Bashandy, A., Gredler, H., and B. Decraene, "IS-IS Extensions for Segment Routing", RFC 8667, DOI 10.17487/RFC8667, December 2019, <https://www.rfc-editor.org/info/rfc8667>.

[RFC8665]

Psenak, P., Ed., Previdi, S., Ed., Filsfils, C., Gredler, H., Shakir, R., Henderickx, W., and J. Tantsura, "OSPF Extensions for Segment Routing", RFC 8665, DOI 10.17487/RFC8665, December 2019, <https://www.rfc-editor.org/info/rfc8665>.

[RFC8669] Previdi, S., Filsfils, C., Lindem, A., Ed., Sreekantiah, A., and H. Gredler, "Segment Routing Prefix Segment Identifier Extensions for BGP", RFC 8669, DOI 10.17487/RFC8669, December 2019, <https://www.rfc-editor.org/info/rfc8669>.

[RFC8663] Xu, X., Bryant, S., Farrel, A., Hassan, S., Henderickx, W., and Z. Li, "MPLS Segment Routing over IP", RFC 8663, DOI 10.17487/RFC8663, December 2019, <https://www.rfc-editor.org/info/rfc8663>.

[RFC7911] Walton, D., Retana, A., Chen, E., and J. Scudder, "Advertisement of Multiple Paths in BGP", RFC 7911, DOI 10.17487/RFC7911, July 2016, <https://www.rfc-editor.org/info/rfc7911>.

[RFC7880] Pignataro, C., Ward, D., Akiya, N., Bhatia, M., and S. Pallagatti, "Seamless Bidirectional Forwarding Detection (S-BFD)", RFC 7880, DOI 10.17487/RFC7880, July 2016, <https://www.rfc-editor.org/info/rfc7880>.

[RFC4364] Rosen, E. and Y. Rekhter, "BGP/MPLS IP Virtual Private Networks (VPNs)", RFC 4364, DOI 10.17487/RFC4364, February 2006, <https://www.rfc-editor.org/info/rfc4364>.

[RFC5920] Fang, L., Ed., "Security Framework for MPLS and GMPLS Networks", RFC 5920, DOI 10.17487/RFC5920, July 2010, <https://www.rfc-editor.org/info/rfc5920>.

[RFC7011] Claise, B., Ed., Trammell, B., Ed., and P. Aitken, "Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information", STD 77, RFC 7011, DOI 10.17487/RFC7011, September 2013, <https://www.rfc-editor.org/info/rfc7011>.

[RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <https://www.rfc-editor.org/info/rfc6241>.

[RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <https://www.rfc-editor.org/info/rfc6020>.

[RFC7854] Scudder, J., Ed., Fernando, R., and S. Stuart, "BGP Monitoring Protocol (BMP)", RFC 7854, DOI 10.17487/RFC7854, June 2016, <https://www.rfc-editor.org/info/rfc7854>.

[RFC8300] Quinn, P., Ed., Elzur, U., Ed., and C. Pignataro, Ed., "Network Service Header (NSH)", RFC 8300, DOI 10.17487/RFC8300, January 2018, <https://www.rfc-editor.org/info/rfc8300>.

[RFC5440] Vasseur, JP., Ed. and JL. Le Roux, Ed., "Path Computation Element (PCE) Communication Protocol (PCEP)", RFC 5440, DOI 10.17487/RFC5440, March 2009, <https://www.rfc-editor.org/info/rfc5440>.

[RFC7348]

Mahalingam, M., Dutt, D., Duda, K., Agarwal, P., Kreeger, L., Sridhar, T., Bursell, M., and C. Wright, "Virtual eXtensible Local Area Network (VXLAN): A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks", RFC 7348, DOI 10.17487/RFC7348, August 2014, <https://www.rfc-editor.org/info/rfc7348>.

[RFC7637] Garg, P., Ed. and Y. Wang, Ed., "NVGRE: Network Virtualization Using Generic Routing Encapsulation", RFC 7637, DOI 10.17487/RFC7637, September 2015, <https://www.rfc-editor.org/info/rfc7637>.

[RFC3031] Rosen, E., Viswanathan, A., and R. Callon, "Multiprotocol Label Switching Architecture", RFC 3031, DOI 10.17487/RFC3031, January 2001, <https://www.rfc-editor.org/info/rfc3031>.

[RFC8014] Black, D., Hudson, J., Kreeger, L., Lasserre, M., and T. Narten, "An Architecture for Data-Center Network Virtualization over Layer 3 (NVO3)", RFC 8014, DOI 10.17487/RFC8014, December 2016, <https://www.rfc-editor.org/info/rfc8014>.

[RFC8402] Filsfils, C., Ed., Previdi, S., Ed., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", RFC 8402, DOI 10.17487/RFC8402, July 2018, <https://www.rfc-editor.org/info/rfc8402>.

[RFC5883] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD) for Multihop Paths", RFC 5883, DOI 10.17487/

RFC5883, June 2010, <https://www.rfc-editor.org/info/rfc5883>.

[RFC8231] Crabbe, E., Minei, I., Medved, J., and R. Varga, "Path Computation Element Communication Protocol (PCEP) Extensions for Stateful PCE", RFC 8231, DOI 10.17487/RFC8231, September 2017, <https://www.rfc-editor.org/info/rfc8231>.

[RFC8281] Crabbe, E., Minei, I., Sivabalan, S., and R. Varga, "Path Computation Element Communication Protocol (PCEP) Extensions for PCE-Initiated LSP Setup in a Stateful PCE Model", RFC 8281, DOI 10.17487/RFC8281, December 2017, <https://www.rfc-editor.org/info/rfc8281>.

[RFC5925] Touch, J., Mankin, A., and R. Bonica, "The TCP Authentication Option", RFC 5925, DOI 10.17487/RFC5925, June 2010, <https://www.rfc-editor.org/info/rfc5925>.

[RFC8253] Lopez, D., Gonzalez de Dios, O., Wu, Q., and D. Dhody, "PCEPS: Usage of TLS to Provide a Secure Transport for the Path Computation Element Communication Protocol (PCEP)", RFC 8253, DOI 10.17487/RFC8253, October 2017, <https://www.rfc-editor.org/info/rfc8253>.

[RFC6790] Kompella, K., Drake, J., Amante, S., Henderickx, W., and L. Yong, "The Use of Entropy Labels in MPLS Forwarding", RFC 6790, DOI 10.17487/RFC6790, November 2012, <https://www.rfc-editor.org/info/rfc6790>.

[RFC8662] Kini, S., Kompella, K., Sivabalan, S., Litkowski, S., Shakir, R., and J. Tantsura, "Entropy Label for Source Packet Routing in Networking (SPRING) Tunnels", RFC 8662, DOI 10.17487/RFC8662, December 2019, <https://www.rfc-editor.org/info/rfc8662>.

[RFC8491] Tantsura, J., Chunduri, U., Aldrin, S., and L. Ginsberg, "Signaling Maximum SID Depth (MSD) Using IS-IS", RFC 8491, DOI 10.17487/RFC8491, November 2018, <https://www.rfc-editor.org/info/rfc8491>.

[RFC8476] Tantsura, J., Chunduri, U., Aldrin, S., and P. Psenak, "Signaling Maximum SID Depth (MSD) Using OSPF", RFC 8476, DOI 10.17487/RFC8476, December 2018, <https://www.rfc-editor.org/info/rfc8476>.

Authors' Addresses

Colin Bookham (editor)
Nokia

740 Waterside Drive
Almondsbury, Bristol
United Kingdom

Email: colin.bookham@nokia.com

Andrew Stone
Nokia
600 March Road
Kanata, Ontario
Canada

Email: andrew.stone@nokia.com

Jeff Tantsura
Microsoft

Email: jefftant.ietf@gmail.com

Muhammad Durrani
Equinix Inc
1188 Arques Ave
Sunnyvale CA,
United States of America

Email: mdurrani@equinix.com

Bruno Decraene
Orange
38-40 Rue de General Leclerc
92794 Issey Moulineaux cedex 9
France

Email: bruno.decraene@orange.com