Supporting VNF chains: an implementation using Segment Routing and PCEP

Portegies, C.; Kaat, M.; Grosso, P.

DOI
10.1109/ICIN51074.2021.9385527

Publication date
2021

Document Version
Final published version

Published in
2021 24th Conference on Innovation in Clouds, Internet and Networks and workshops

License
Article 25fa Dutch Copyright Act (https://www.openaccess.nl/en/in-the-netherlands/you-share-we-take-care)

Link to publication

Citation for published version (APA):
Portegies, C., Kaat, M., & Grosso, P. (2021). Supporting VNF chains: an implementation using Segment Routing and PCEP. In *2021 24th Conference on Innovation in Clouds, Internet and Networks and workshops* (pp. 1-5). (ICIN). IEEE. https://doi.org/10.1109/ICIN51074.2021.9385527

2021 24th Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN) | 978-1-7281-7705-2/20/$31.00 ©2021 IEEE | DOI: 10.1109/ICIN51074.2021.9385527

Supporting VNF chains: an implementation using Segment Routing and PCEP

1st Cees Portegies
Multiscale Networked Systems group
University of Amsterdam
Amsterdam, The Netherlands
c.portegies@uva.nl

2nd Marijke Kaat
SURFnet networking group
SURF Foundation
Utrecht, The Netherlands
marijke.kaat@surf.nl

3rd Paola Grosso
Multiscale Networked Systems group
University of Amsterdam
Amsterdam, The Netherlands
p.grosso@uva.nl

***Abstract*—The paradigm of Network Function Virtualization (NFV) requires the underlying networks to be able to route traffic through dynamically deployed nodes. The research presented here demonstrates the feasibility of using segment routing (SR), a modern incarnation of the source routing paradigm, and suitable SDN controllers to accomplish the traffic steering goal. Specifically, we focused on IPv4 SR and built a proof of concept SR-MPLS network using Juniper vQFX routers with custom VNFs. The Juniper NorthStar SDN controller was used to build SR-MPLS LSPs via the Path Computation Element Communication Protocol (PCEP). We validated the operation of our proof of concept in two scenarios: VNFs reinstantiation and service chain creation and support. In both cases traffic can be successfully steered through the functions, hence demonstrating for the first time the feasibility of adoption of the proposed technologies.**

***Index Terms*—Segment Routing, PCEP, SDN, NFV, VNF**

I. Introduction

Network Function Virtualization (NFV) allows to deploy network infrastructure in clouds, as traditionally hardware-bound functions, such as firewalling, routing and inspecting packets, are packaged into Virtual Network Functions (VNFs). A very interesting use of VNFs is when they are not used in isolation but instead chained as succession of multiple of these functions to form so called Service Function Chains [1]. Service Function chains are being actively investigated by service providers in a number of networking contexts, e.g. in the deployment of 5G radio access networks [2], [3] and in the flexible delivery of firewalling services in research and education networks like SURFnet, with whom we collaborated in this research.

For all interested adopters of VNFs the major hurdle to solve is to have the underlying network route traffic through each of the VNFs, wherever they may be hosted. Our research demonstrates how the source routing (SR) paradigm, implemented in the segment routing architecture, can support the dynamic use of VNFs and particularly their chains. Our work relies on the main feature of this paradigm, i.e. the fact that a network path is specified at ingress of the network and attached to each packet. This allows us to have each packet pass through the intended VNFs. To define and control the paths in an SR data plane we decided to rely on a Software Defined Networking controller (SDN) as this allows automated control of the network.

We focused our development on the SR-MPLS data plane, the IPv4 implementation for segment routing. We adopted the Path Computation Element Communication Protocol (PCEP) to control paths in the data plane, as PCEP is specifically aimed at path computation and path signaling [4]. The question our work answers is: ”How can we create SR-MPLS network paths to assist the network integration of VNFs?”, and more specifically: ”Can PCEP be used to this purpose?” The work we present here describes the operational proof of concept we built that shows how segment routed paths, specifically SR-MPLS paths, can be set up with PCEP to have traffic make use of VNFs and VNF chains. This work, to be best of our knowledge, is the first of its kind.

II. Background

To understand our work it is necessary to briefly introduce the technologies we rely on: segment routing, and its relation to NFV, as well as SDN controllers and PCEP.

A. Segment Routing

In source routing ”... the source of internet packets specifies the complete internet route.” [5], i.e. the source of the packets is the main factor determining the route. This is different from traditional IP routing where the final destination of the packet is the deciding factor. The *internet route*, path, specified at the source is attached to the packet and this path information is used for routing in the network. One of the main advantages of this mode of operation is that there is less complex routing needed in the network: each transit node makes use of the information attached to the packet for specific routing decisions; the source routers are responsible to attach the correct path information and no state information needs to be kept by the transit nodes. This reduces the computational and signaling overhead at these transit nodes, but does create additional overhead in packet size as the path information is attached.

The source routing concept has been implemented in the *segment routing* paradigm [6], [7]. The specific path to be

Part of this work has been supported by SURFnet under the RoN - Research on Networks program - 2020

978-1-7281-7705-2/21/$31.00 ©2021 IEEE

taken is specified as *segments* via segment identifiers (SIDs). There are four types of segments: node, adjacency, prefix and anycast. A node segment represents a specific network node and the adjacency segment represents the link between two nodes; the prefix segment type represents an IP prefix, while anycast segments can be used for anycast traffic.

Segment routing can be implemented as an extension of MPLS in SR-MPLS [8], [9]. In SR-MPLS, MPLS labels are used to specify segments: a 20-bit MPLS label can represent any of the SID types.

When an MPLS label is assigned as a specific SID, this information can be distributed via an intra-domain routing protocol (IGP). Currently both OSPF and IS-IS have support for the SR-MPLS extensions needed to carry this information [10]–[12]. There is an important difference between the use of labels in MPLS and in SR-MPLS: in MPLS the label has only local significance but acquires global significance within the scope of the network in SR-MPLS. A path through an SR-MPLS network is called a Segment Routed Label Switched Path (SR-LSP) and such a path is unidirectional. The segments associated with an SR-LSP are attached to a packet upon ingress in the network as a stack of MPLS labels. To combine this with NFV, a VNF might be represented as a specific segment, i.e. node, to be visited by the SR-LSP.

B. NFV and Segment Routing

Network Function Virtualization (NFV) was proposed to extend the trend of virtualizing workloads to include networking infrastructure like routers and firewalls [13]. Similar to the original virtualization paradigm, NFV allows better utilization of computing resources and more dynamic deployment. This dynamic deployment and instantiation of network devices is core to our research. With NFV, devices become deployable Virtual Network Functions (VNFs), usually in the form of VMs. The key point is then to redirect the flow of traffic to these devices based on their specific function.

Specifically when looking, as we did, to the integration of segment routing and VNF, we must point out that there are two types of virtual network functions: those that are segment routing compatible (*SR-Aware*) and those that are not (*SR-Unaware*). SR-Aware VNFs are capable of understanding the SR-encapsulation of the packets and can act accordingly, i.e. being directly compatible with the SR data plane. SR-Unaware VNFs can only be deployed in conjunction with *SR-proxies*, which serve as a bridge between the SR domain and the SR-Unaware VNF. Service Function Chaining is the coupling of multiple services in a series, in order to do a composite task [2], [14]. Within the context of VNFs, this means that multiple VNFs might need to be visited by traffic in order to correctly process it. The challenge is how to build such a chain and how to route the traffic such that it can visit each of the nodes and associated functions in the chain. To support this we need controllers.

C. SDN & PCEP

Software Defined Networking (SDN) controllers have emerged as elements to provide some form of automated control and abstraction over the network [15]. In general SDN controllers have shown to be useful for VNFs to assist in the deployment of service chains [16]–[18], beyond the scope of the segment routing data plane onto which we focus. SDN controller functionality is two-fold: first, the controller needs a view of the network and secondly, a way to control it.

In our research we used the Link State extension of the BGP protocol (BGP-LS) to fulfill the first task. This extension is aimed at sharing traffic engineering and link state information with an external component via BGP. A newly defined NLRI (Network Layer Reachability Information) encoding format describes node, link and prefix information to be shared. Thanks to this we can use BGP-LS to gather topological information at the SDN controller [19].

When addressing the second requirement, i.e. the control of the network elements, we focused on the Path Computation Element Communication Protocol (PCEP), which is aimed at communicating network paths [4]. For our purposes the goal is to push SR-LSPs to the routers in the network, and to this end, PCEP introduces a Path Computation Client (PCC) and a Path Computation Element (PCE). The PCC can request a path to be computed by the PCE and the PCE can compute a path and push this information back out to the PCC. The path is pushed out as an Explicit Route Object (ERO). In the PCEP protocol, support for segment routing was added as per RFC 8664 as segment routing EROs (SR-EROs) [20]. Previous work [21] and [22] have demonstrated the viability of using PCEP and BGP-LS in the MPLS date plane to construct paths, making them a good starting point for our implementation.

III. Proof of Concept Design

Having determined the base technologies our next step has been to design our prototype. Fig. 1 gives a full overview of our intended proof of concept.

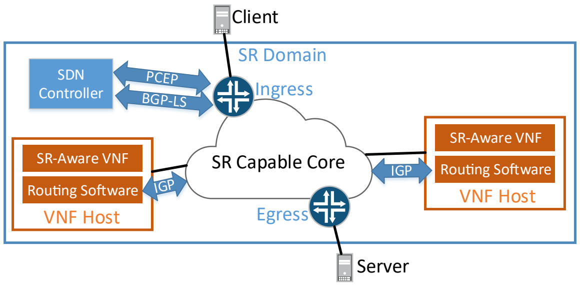

Fig. 1. Overview of the elements and protocol interactions.

The proof of concept network will have a *SR capable core* consisting of *VNF hosts* and SR capable routers. A client and server will be outside the SR capable core serving as the source and destination for traffic. We aim to steer their traffic through the VNFs in the network with SR-LSPs. The VNFs will be located on the VNF hosts and thus must either be SR-Aware or encapsulated by an SR-Proxy. The segment routed paths in the SR-MPLS data plane are controlled by an *SDN controller* via PCEP. We need to use an IGP which enables passing the segment routing information between each of these elements. Depending on the VNFs employed, the traffic might flow through one VNF or multiple VNFs when service chaining is deployed. Finally, all these components have to be coordinated: we need to be able to instantiate the VNFs on the hosts, and to add and change the SR-LSPs accordingly. This can be accomplished by developing a tool that can communicate with the hosts which run the VNFs as well as interact with the SDN controller to instruct it to change the paths as needed.

IV. Prototype

Based on the design, we created the implementation shown in Fig. 2. The core of the SR-MPLS network consists of Juniper vQFX 19.4R1.10 routers, R1 to R4. The VNFs are hosted at the edge of the network on Ubuntu 18.04 based machines, host12 and host13. Host11 and host14 serve as client and server respectively, with the SR-LSP steering traffic through the VNFs. The SDN controller used to control the SR-LSPs in the network is Juniper NorthStar 6.0.0. This SDN controller maintains a PCEP session with the ingress router in order to setup the SR-LSPs steering traffic to the VNFs.

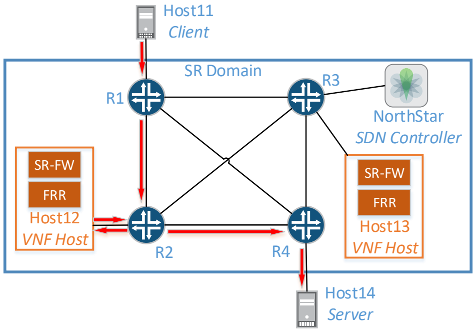

Fig. 2. Proof of concept implementation at the University of Amsterdam. We show also (in red) an example SR-LSP with traffic passing through the SR-Firewall on Host12.

The VNFs we use were custom developed at the University of Amsterdam to be SR-Aware [23]. There are currently three implementations available: SR-Firewall, SR-Mirror and SR-Proxy. The SR-Firewall is an SR-Aware firewall, used to filter traffic. The SR-Firewall does not keep state on flows and applies the rules on a per packet basis. The SR-Mirror mirrors traffic and the SR-Proxy allows for the integration of SR-Unaware VNFs. The VNFs are implemented as Berkeley Packet Filter (BPF) programs, specifics on their implementation can be found in [23]. For the development and evaluation of our prototype, the specific functionality of a VNF is not strictly of relevance. So, to make our work resonate more with production networks such as the SURFnet network in the Netherlands, we decided to focus in our evaluation onto the SR-MPLS capable firewalls exclusively.

To allow the VNF host machines to participate in the SR domain and its IGP, they run the Free Range Routing[1] (FRR) software. In our proof of concept, we used IS-IS as the IGP to carry the segment routing information and FRR version 7.4-dev supports the segment routing extensions for IS-IS. The main reason to use IS-IS was that the documentation as well as support for IS-IS segment routing appear to be more mature as compared to OSPF.

A crucial point of the implementation is the mechanism by which the VNFs are coordinated with the SR-LSPs. This involves the loading and unloading of the VNFs and correctly building the applicable SR-LSPs. We accomplished this via an external element, a tool specially developed as part of this research. This tool coordinates the SR-LSPs with the VNFs by using the available APIs. This tool uses the NorthStar Northbound API, which is used to instruct the SDN controller to build and modify the SR-LSPs. To control the VNFs, we use the BPF CLI on the VNF hosts. The tool is used from an external machine which has access to all nodes and to NorthStar. We rely on the NorthStar GUI to visualize the state of the network and the current SR-LSPs. Fig. 3 shows for example a path going through a VNF in host12.

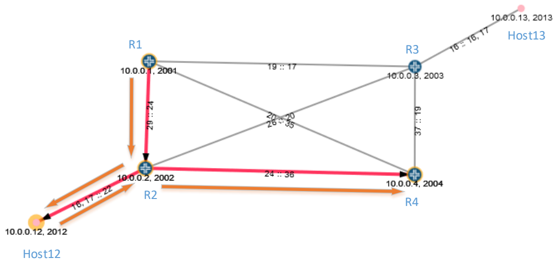

Fig. 3. Proof of concept network in NorthStar GUI with annotated SR-LSP, where traffic flows through the VNF on host12

We make all code developed for this research available via BitBucket[2] to support reproducability of our work as well as engage with other researchers interested in these efforts.

V. Experimental Results

To prove that our PoC is functional we focused on two scenarios: VNF reinstantiation and VNF chains.

A. Experiment 1: VNF reinstantiation

In this experiment we want to determine the impact of reinstantiating a VNF on the end-to-end traffic. We reinstantiate the SR-Firewall VNF originally in host12 on host13 and evaluate the impact on a traffic flow from host11 to host14. The SR-LSP is changed dynamically such that the traffic will flow via host13 after the reinstantiation. If the reinstantiation has succeeded, we should not observe any degradation or losses as this means the any change in the VNF location is transparent to the user.

[1] https://frrouting.org/

[2] https://bitbucket.org/uva-sne/ron20_sr

At the beginning of our experiment the SR-Firewall VNF is hosted on host12 (Fig. 3); later the VNF is reinstantiated on host13. From host11 pings are sent every second to host14. Our goal is to ensure that no packets are dropped. We will therefore instantiate the new VNF on host13 before changing the SR-LSP. We repeated the reinstantiation experiment 5 times to determine if any packet loss occurs. We never observed any packet loss in any of the iterations, confirming that our PoC can achieve transparent reinstantiation.

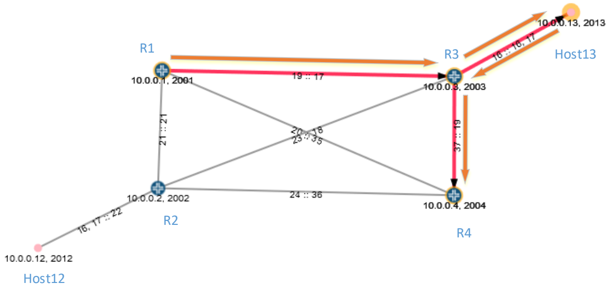

Fig. 4. NorthStar view of the SR-LSR path after reinstantiation

The migrated SR-LSP is visible from within NorthStar and is shown in Fig. 4. This correctly shows the new path, first heading to R3 before going to host13.

B. Experiment 2: VNF service chains

In the second experiment we want to prove we can setup a service chain of VNFs at ingress, requiring the SR-LSP to target several VNF hosts in the network. Specifically, we will deploy two SR-Firewalls in a chain, each with different rules, as this allows us to easily demonstrate the functionality of our PoC as we can observe how chaining influences traffic in serial fashion. Initially, the SR-LSP goes through host12, where the SR-Firewall is hosted and no VNF is deployed on host13. On host12, the SR-Firewall allows traffic to TCP port 80 on the server, host14; in contrast, the SR-Firewall on host13 will drop that traffic. Thus, once the services are chained in the SR-LSP, the traffic targeting port 80 on the server should be filtered. To test that our chain is operational, we start two simultaneous constant streams of TCP handshake attempts on host11 towards port 80 and port 443 on host14 at a rate of 1/s. As soon as the services are chained, and if the SR-LSP path works as intended, the TCP handshake attempts towards port 80 should time out as they cannot reach host14 and thus cannot get a reply. The handshake attempts towards port 443 should still receive a reply.

The results for this experiment are shown in Fig. 6 before and after the start of the VNF chain. We can observe that before the chaining (left part of the plot) the packets have an average latency of 0.269 seconds. This high latency is likely the result of the vQFX routers. Once the chain is made (right part of the plot) we observe that only the traffic targeting port 443 still receives replies, the traffic towards port 80 does not, indicated with a latency of 0. This confirms that the second firewall is dropping the traffic as intended. The resulting SR-LSP going through host12 and host13 is shown in Fig. 5.

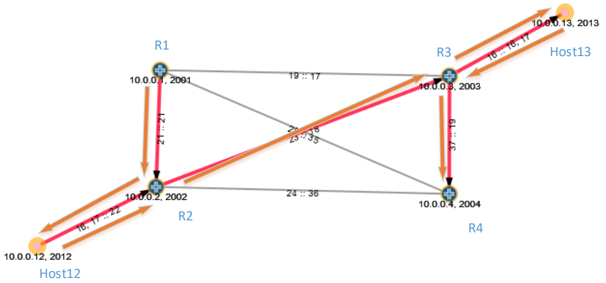

Fig. 5. Chained SR-LSP as shown in NorthStar where traffic passes via host12 and host13.

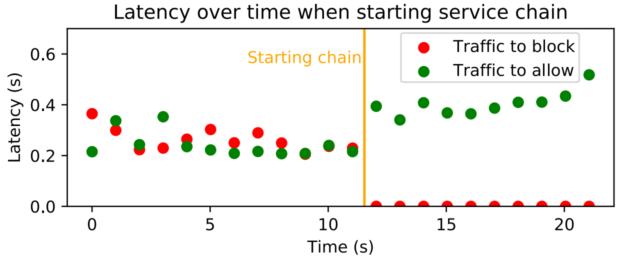

Fig. 6. Latency over the chaining of the SR-Firewall VNF, latency of 0 denotes a missed reply, the red line the moment when the chain was made.

Both experiments could be conducted successfully in our proof of concept, and the tool we developed worked well to coordinate the instantiation of VNFs with the SR-LSPs. However, the construction of the network exposed some shortcomings. The documentation as well as configuration possibilities for SR-MPLS are limited due to the relative immaturity of the technology as compared to IP routing. This was especially notable in the differences between the OSPF and IS-IS configuration for segment routing within Free Range Routing. These segment routing implementations are still regarded as experimental. In the used Juniper vQFX routers, support for segment routing is not equal between OSPF and IS-IS and is still changing with new releases.

For this research, the Juniper NorthStar SDN controller has been integrated in the SR network. This integration consisted of two parts, providing the controller with topological information and giving it control over the network. Providing the SDN controller with the topological information was done by using BGP-LS. However, for NorthStar a RESTCONF [24] connection was also needed to allow NorthStar to correlate interface information with the BGP-LS information. In the proof of concept network, the vQFX routers could support this RESTCONF connection, but the FRR software on the VNF hosts could not. Despite missing the RESTCONF connection to the VNF hosts, NorthStar could still function. This is presumably because the core of the network consisted of the vQFX routers, with the VNF hosts at the edges. In our view, the requirement from the SDN controller to have two protocols supported by the network makes integration with more heterogeneous networks more challenging.

VI. Conclusion

We started our work trying to answer the question: "How can we create SR-MPLS network paths to assist the network integration of VNFs?", and more specifically: "Can PCEP be used for this purpose?" The proof of concept we built successfully combines segment routing technology with PCEP supported by SDN. We demonstrated that network paths, targeting specific VNFs, in an SR-MPLS network can be built by PCEP and coordinated with the VNFs. Node SIDs can be used successfully to represent VNFs in network paths pushed out via the PCEP protocol. Our work opens up a significant number of further research opportunities in the area of VNFs and VNF chains. It serves as a stepping stone for production deployments of VNFs, for example in 5G networks and for granular and flexible use of firewalls. Our future research, expanding beyond this working prototype, will center on increased user control, and allow us to continue our efforts in the area of intelligent VNF deployment [25]. From a more technical perspective, we are also interested in conducting a similar study for SRv6, as this would provide a useful comparison with the SR-MPLS implementation described here.

Acknowledgment

This work was made possible thanks to the prior work and support from Łukasz Makowski. Juniper supported this research by providing the NorthStar controller software.

References

[1] J. M. Halpern and C. Pignataro, *Service Function Chaining (SFC) Architecture*, RFC 7665, Oct. 2015. DOI: 10.17487/RFC7665. [Online]. Available: https://rfc-editor.org/rfc/rfc7665.txt.

[2] P. Quinn and T. Nadeau, *Problem Statement for Service Function Chaining*, RFC 7498, Apr. 2015. DOI: 10.17487/RFC7498. [Online]. Available: https://rfc-editor.org/rfc/rfc7498.txt.

[3] D. Zhao, J. Ren, R. Lin, S. Xu, and V. Chang, "On orchestrating service function chains in 5g mobile network," *IEEE Access*, vol. 7, pp. 39 402–39 416, 2019.

[4] J. Vasseur and J.-L. L. Roux, *Path Computation Element (PCE) Communication Protocol (PCEP)*, RFC 5440, Mar. 2009. DOI: 10.17487/RFC5440. [Online]. Available: https://rfc-editor.org/rfc/rfc5440.txt.

[5] C. A. Sunshine, "Source routing in computer networks," *ACM SIGCOMM Computer Communication Review*, vol. 7, no. 1, pp. 29–33, 1977.

[6] C. Filsfils, N. K. Nainar, C. Pignataro, J. C. Cardona, and P. Francois, "The segment routing architecture," in *2015 IEEE Global Communications Conference (GLOBECOM)*, IEEE, 2015, pp. 1–6.

[7] C. Filsfils, S. Previdi, L. Ginsberg, B. Decraene, S. Litkowski, and R. Shakir, *Segment Routing Architecture*, RFC 8402, Jul. 2018. DOI: 10.17487/RFC8402. [Online]. Available: https://rfc-editor.org/rfc/rfc8402.txt.

[8] A. Bashandy, C. Filsfils, S. Previdi, B. Decraene, S. Litkowski, and R. Shakir, *Segment Routing with the MPLS Data Plane*, RFC 8660, Dec. 2019. DOI: 10.17487/RFC8660. [Online]. Available: https://rfc-editor.org/rfc/rfc8660.txt.

[9] X. Xu, S. Bryant, A. Farrel, S. Hassan, W. Henderickx, and Z. Li, *MPLS Segment Routing over IP*, RFC 8663, Dec. 2019. DOI: 10.17487/RFC8663. [Online]. Available: https://rfc-editor.org/rfc/rfc8663.txt.

[10] P. Psenak, S. Previdi, C. Filsfils, H. Gredler, R. Shakir, W. Henderickx, and J. Tantsura, *OSPF Extensions for Segment Routing*, RFC 8665, Dec. 2019. DOI: 10.17487/RFC8665. [Online]. Available: https://rfc-editor.org/rfc/rfc8665.txt.

[11] P. Psenak and S. Previdi, *OSPFv3 Extensions for Segment Routing*, RFC 8666, Dec. 2019. DOI: 10.17487/RFC8666. [Online]. Available: https://rfc-editor.org/rfc/rfc8666.txt.

[12] S. Previdi, L. Ginsberg, C. Filsfils, A. Bashandy, H. Gredler, and B. Decraene, *IS-IS Extensions for Segment Routing*, RFC 8667, Dec. 2019. DOI: 10.17487/RFC8667. [Online]. Available: https://rfc-editor.org/rfc/rfc8667.txt.

[13] M. Chiosi, D. Clarke, P. Willis, A. Reid, J. Feger, M. Bugenhagen, W. Khan, M. Fargano, C. Cui, H. Deng, *et al.*, "Network functions virtualisation: Introductory white paper," in *SDN and OpenFlow World Congress*, sn, vol. 48, 2012. [Online]. Available: https://portal.etsi.org/nfv/nfv_white_paper.pdf.

[14] P. Quinn and J. Guichard, "Service function chaining: Creating a service plane via network service headers," *Computer*, vol. 47, no. 11, pp. 38–44, 2014.

[15] D. Kreutz, F. M. Ramos, P. E. Verissimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," *Proceedings of the IEEE*, vol. 103, no. 1, pp. 14–76, 2014.

[16] A. Mohammed, M. Gharbaoui, B. Martini, F. Paganelli, and P. Castoldi, "Sdn controller for network-aware adaptive orchestration in dynamic service chaining," in *2016 IEEE NetSoft Conference and Workshops (NetSoft)*, IEEE, 2016, pp. 126–130.

[17] R. Munoz, R. Vilalta, R. Casellas, R. Martınez, T. Szyrkowiec, A. Autenrieth, V. López, and D. López, "Sdn/nfv orchestration for dynamic deployment of virtual sdn controllers as vnf for multi-tenant optical networks," in *Optical Fiber Communication Conference*, Optical Society of America, 2015, W4J–5.

[18] J. Matias, J. Garay, N. Toledo, J. Unzilla, and E. Jacob, "Toward an sdn-enabled nfv architecture," *IEEE Communications Magazine*, vol. 53, no. 4, pp. 187–193, 2015.

[19] H. Gredler, J. Medved, S. Previdi, A. Farrel, and S. Ray, *North-Bound Distribution of Link-State and Traffic Engineering (TE) Information Using BGP*, RFC 7752, Mar. 2016. DOI: 10.17487/RFC7752. [Online]. Available: https://rfc-editor.org/rfc/rfc7752.txt.

[20] S. Sivabalan, C. Filsfils, J. Tantsura, W. Henderickx, and J. Hardwick, *Path Computation Element Communication Protocol (PCEP) Extensions for Segment Routing*, RFC 8664, Dec. 2019. DOI: 10.17487/RFC8664. [Online]. Available: https://rfc-editor.org/rfc/rfc8664.txt.

[21] G. Rzym, K. Wajda, and P. Chołda, "Sdn-based wan optimization: Pce implementation in multi-domain mpls networks supported by bgp-ls," *Image Processing & Communications*, vol. 22, no. 1, pp. 35–48, 2017.

[22] O. Dugeon, R. Guedrez, S. Lahoud, and G. Texier, "Demonstration of segment routing with sdn based label stack optimization," in *2017 20th Conference on Innovations in Clouds, Internet and Networks (ICIN)*, IEEE, 2017, pp. 143–145.

[23] Ł. Makowski, *Ron19: Mpls-sr nfv*, 2020. [Online]. Available: https://wiki.surfnet.nl/display/SURFnetnetwerkWiki/Research+And+Development+Projects+2019.

[24] A. Bierman, M. Björklund, and K. Watsen, *RESTCONF Protocol*, RFC 8040, Jan. 2017. DOI: 10.17487/RFC8040. [Online]. Available: https://rfc-editor.org/rfc/rfc8040.txt.

[25] L. Ochoa-Aday, C. Cervelló-Pastor, A. Fernández-Fernández, and P. Grosso, "An online algorithm for dynamic nfv placement in cloud-based autonomous response networks," *Symmetry*, vol. 10, no. 5, p. 163, 2018.