Experience with implementing VNF chains with Segment Routing and PCEP

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

Publication date
2021

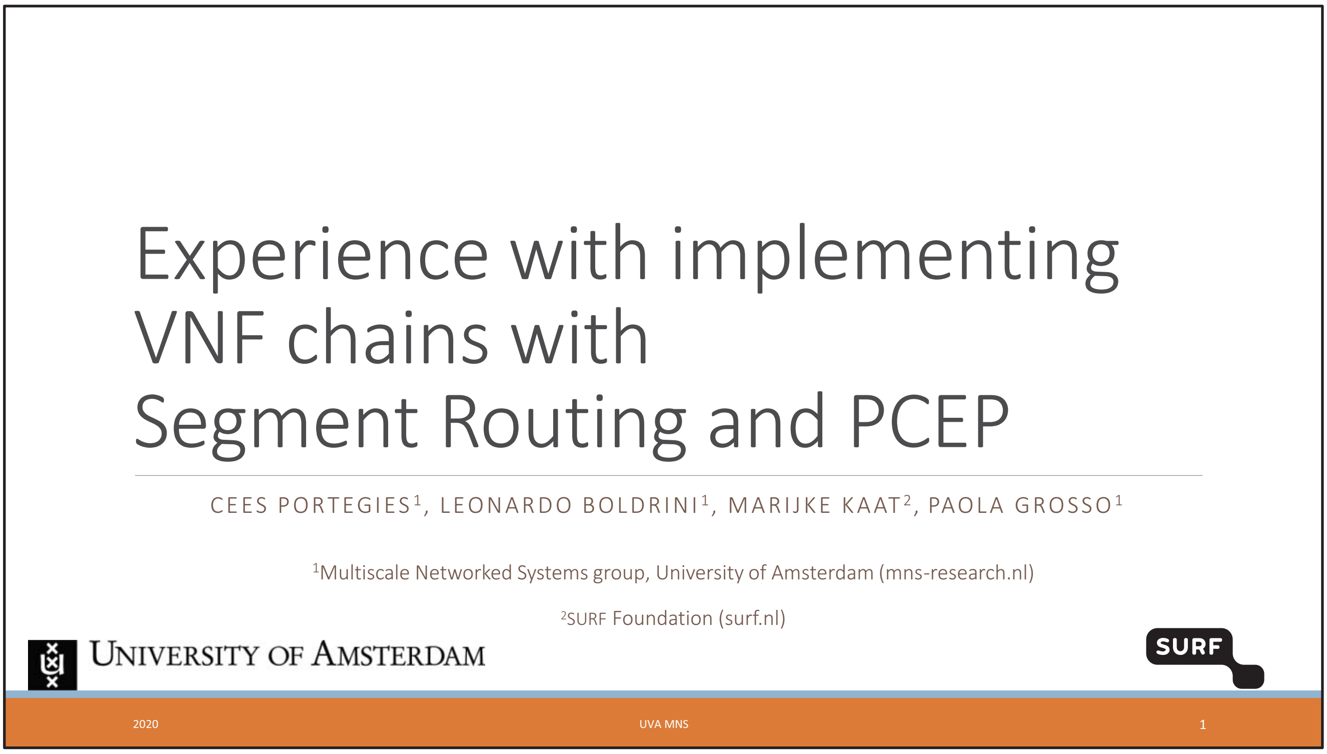
Document Version
Final published version

Published in
Proceedings of the IM 2021

License
Article 25fa Dutch Copyright Act

Link to publication

Citation for published version (APA):
Portegies, C., Boldrini, L., Kaat, M., & Grosso, P. (2021). Experience with implementing VNF chains with Segment Routing and PCEP. In T. Ahmed, O. Festor, Y. Ghamri-Doudane, J-M. Kang, A. E. Schaeffer-Filho, A. Lahmadi, & E. Madeira (Eds.), *Proceedings of the IM 2021: 2021 IFIP/IEEE International Symposium on Integrated Network Management : 17-21 May 2021, Bordeaux, France, virtual conference* (pp. 797-811). IEEE. https://ieeexplore.ieee.org/document/9463969



Experience with implementing VNF chains with Segment Routing and PCEP

CEES PORTEGIES[1], LEONARDO BOLDRINI[1], MARIJKE KAAT[2], PAOLA GROSSO[1]

[1]Multiscale Networked Systems group, University of Amsterdam (mns-research.nl)

[2]SURF Foundation (surf.nl)

UNIVERSITY OF AMSTERDAM

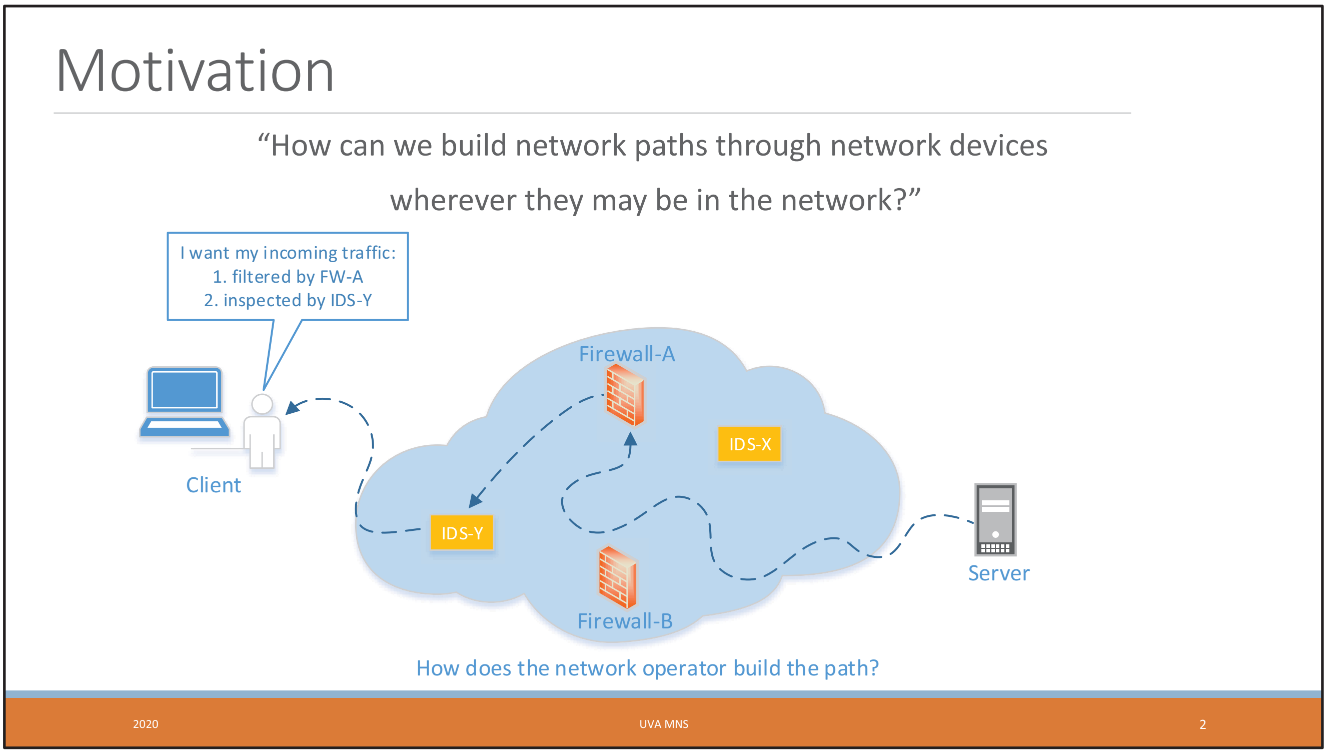
SURF

2020 UVA MNS 1

We present here an implementation to support function chaining and migration in the Network Function Virtualization (NFV) paradigm, using Segment Routing (SR) and Path Computation Element Communication Protocol (PCEP). The paradigm of NFV requires the underlying networks to be able to route traffic through dynamically deployed Virtual Network Functions (VNFs). Segment Routing is a modern incarnation of the source routing paradigm, which together with PCEP we can use to build network paths, steering the traffic through specific segments. The proof of concept we built uses SR-MPLS, focused on IPv4 SR, and consists of Juniper vQFX routers with custom VNFs. The VNFs are custom developed BPF programs that run on special VNF hosts, integrated into the SR-MPLS network. To build SR-MPLS paths through the network, we use the Juniper NorthStar Software Defined Networking (SDN) controller and the PCEP protocol. We developed two scenarios to validate the functionality of our proof of concept: the migration of VNFs and the creation of VNF service chains. We demonstrate that in both scenarios we can successfully steer the traffic through the VNFs, showing the feasibility of using SR with PCEP to assist the deployment of VNFs.

Part of this work has been supported by SURFnet under the RoN - Research on Network program – 2020
This research received funding from the Dutch Research Council (NWO) under the project UPIN

978-3-903176-32-4 ©2021 IFIP



Motivation

“How can we build network paths through network devices wherever they may be in the network?”

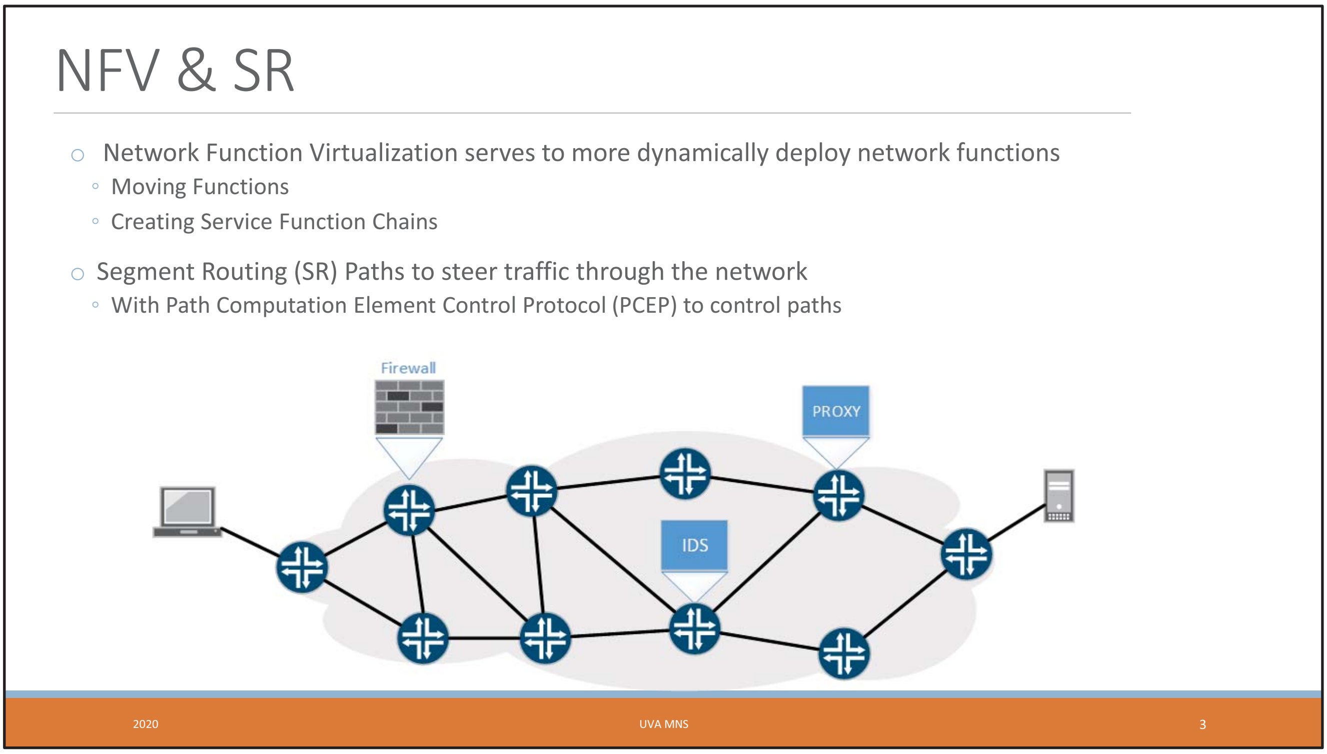
How does the network operator build the path?

2020 UVA MNS 2

Currently, the entire IT landscape is shifting to a more automated, more dynamic deployment model. Services are being deployed in the cloud in an ever increasing scale with even the infrastructure itself becoming virtualized and cloud deployable. The challenge becomes for network operators to make it possible to steer client traffic to specific networking infrastructure, wherever they are hosted in the network.
This trend of making more services cloud deployable is present in the upcoming paradigm of Network Function Virtualization (NFV) [1].

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 that the source routing 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.

[1] B. Han, V. Gopalakrishnan, L. Ji, and S. Lee, “Network function virtualization: Challenges and opportunities for innovations,” IEEE Communications Magazine, vol. 53, no. 2, pp. 90–97, 2015.



NFV & SR

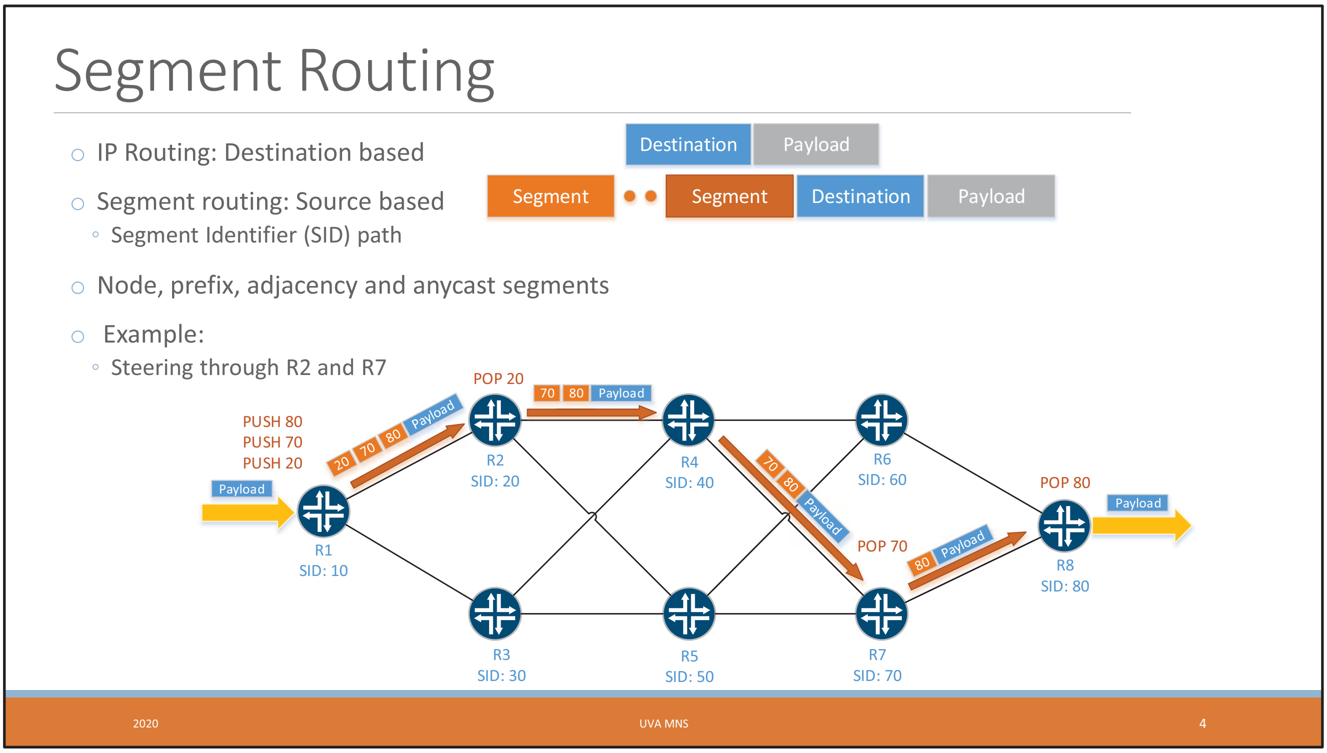
- Network Function Virtualization serves to more dynamically deploy network functions
  - Moving Functions
  - Creating Service Function Chains
- Segment Routing (SR) Paths to steer traffic through the network
  - With Path Computation Element Control Protocol (PCEP) to control paths

2020 UVA MNS 3

Network function virtualization (NFV) extends the trend of virtualizing workloads to include networking infrastructure like routers and firewalls [2]. Similar to the original virtualization paradigm, NFV allows better utilization of computing resources and more importantly within the context of this research, more dynamic deployment, instantiation, of network devices. Traditionally, putting in any sort of networking device would require the network operator to physically procure the device and install it in the network. With NFV, this appliance becomes a virtual network function (VNF) to be deployed, instantiated, on some sort of hosting infrastructure. The key point is then to redirect the flow of traffic to this new appliance based on the specific needs. As outlined in [2], this does pose challenges as it sets stringent requirements on the network in order to facilitate redirecting traffic without impacting the reliability of the network. Beyond merely steering the traffic in the network, such a VNF might be instructed to perform a specific action on the packet based on the headers in the packet, allowing a degree of programmability of the network.

Service Function Chaining is the concept of utilizing multiple services in a series in order to accomplish some sort of goal specific task [3]. Within the context of VNFs, this means that multiple VNFs might be needed to be visited by traffic in order to fulfil this task. For example, traffic might first need to be processed by a firewall and then by an IDS for it to be correctly handled. With this then comes the challenge of how to correctly build such a chain and how to route the traffic such that it can visit each of the services in the chain.

[2]: 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.
[3]: P. Quinn and T. Nadeau, Problem Statement for Service Function Chaining, RFC 7498, Apr. 2015. doi: 10.17487/RFC7498.



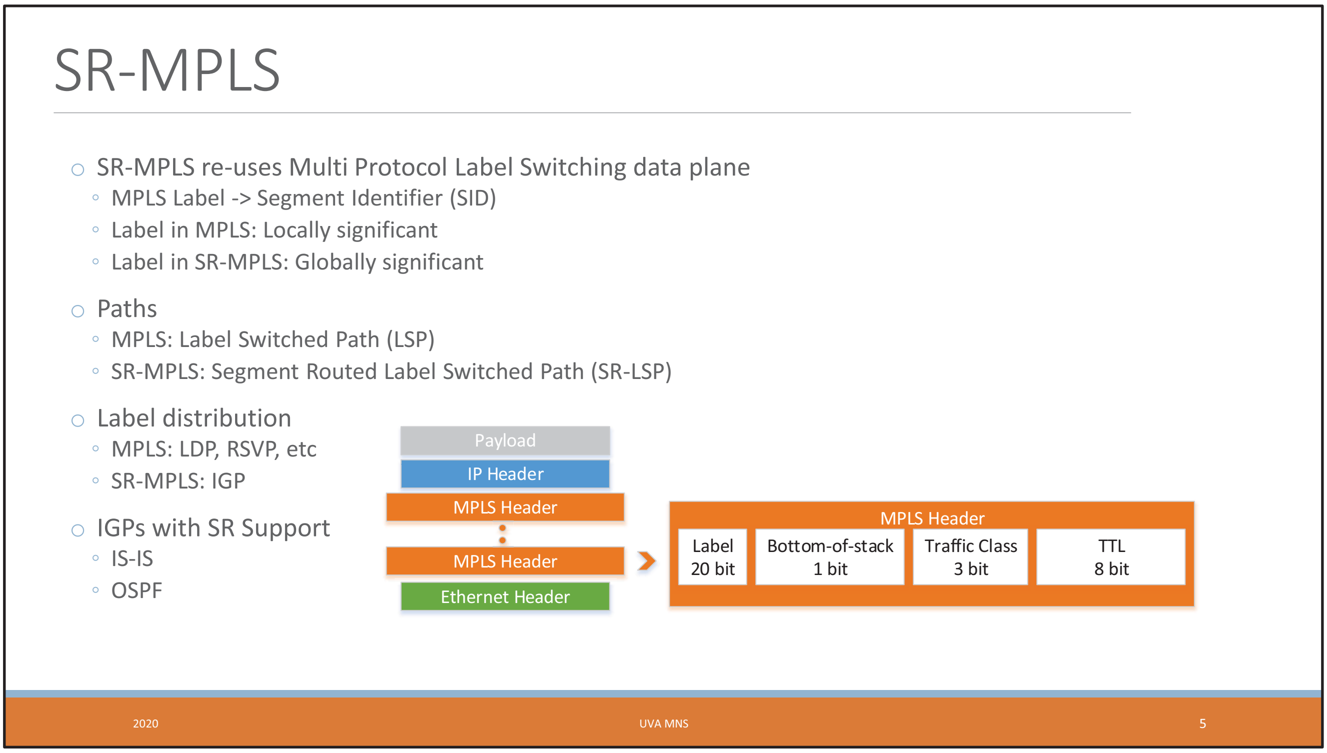
Segment Routing

- IP Routing: Destination based
- Segment routing: Source based
  - Segment Identifier (SID) path
- Node, prefix, adjacency and anycast segments
- Example:
  - Steering through R2 and R7

2020 UVA MNS 4

The paradigm of segment routing is based on the concept of source routing; ``... in which the source of internet packets specifies the complete internet route.'' [4]
This immediately explains the name, as the source of the packets is the principle factor deciding the route. This is different from traditional IPv4 and IPv6 routing in which the final destination of the packet is the deciding factor. This internet route, path, specified at the source is then attached to the packet and this attached path information is used by the next nodes in the network. This source routing paradigm is used in the standard for segment routing, where the path is defined in segments [5, 6]. These segments are represented as Segment Identifiers (SIDs) and the path specified in SIDs is attached to the packet. There are various sorts of these SIDs, namely node, prefix and adjacency and anycast segment identifiers. Each of these can be used to specifically steer the traffic, for example towards nodes with node SIDs.
In the given example network, some payload heads from R1 through the network to R8. We decide that the traffic must always explicitely visist R2 and R7. The rest of the intermediate nodes are decided by the IGP. The path we want goes from R1, through R2, then R7 and ends at R8. Once the traffic arrives at the ingress, R1, it will push the SIDs corresponding with the path we want. We first push the last destination in the path and the last item on the stack will be the first node we want to visist. Thus, we push 80, 70 then 20.

[4]: C. A. Sunshine, "Source routing in computer networks," ACM SIGCOMM Computer Communication Review, vol. 7, no. 1, pp. 29–33, 1977.
[5]: 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.
[6]: C. Filsfils, S. Previdi, L. Ginsberg, B. Decraene, S. Litkowski, and R. Shakir, Segment Routing Architecture, RFC 8402, Jul. 2018. doi: 10.17487/RFC8402.



SR-MPLS

- SR-MPLS re-uses Multi Protocol Label Switching data plane
  - MPLS Label -> Segment Identifier (SID)
  - Label in MPLS: Locally significant
  - Label in SR-MPLS: Globally significant
- Paths
  - MPLS: Label Switched Path (LSP)
  - SR-MPLS: Segment Routed Label Switched Path (SR-LSP)
- Label distribution
  - MPLS: LDP, RSVP, etc
  - SR-MPLS: IGP
- IGPs with SR Support
  - IS-IS
  - OSPF

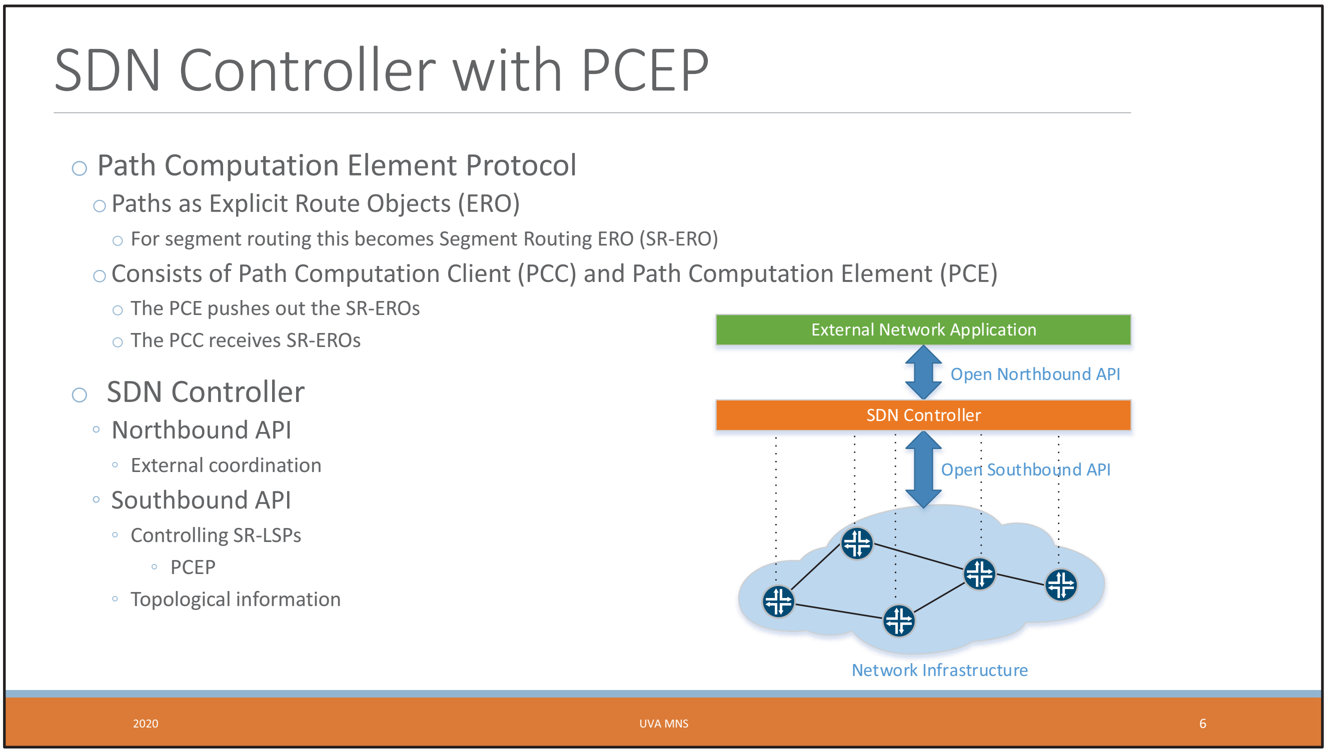
Payload
IP Header
MPLS Header
MPLS Header
Ethernet Header

MPLS Header

Label 20 bit	Bottom-of-stack 1 bit	Traffic Class 3 bit	TTL 8 bit

Segment routing itself is just a standard and concept, and needs an actual technological underpinning to serve as a data plane technology. It has been implemented as an extension of MPLS in SR-MPLS as per RFC 8660 and RFC 8663 [7, 8]. In SR-MPLS, MPLS labels are used to specify segments: a 20-bit MPLS label can represent any of the SID types. 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. 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 [9, 10, 11]. 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.

[7]: 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.
[8]: 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.
[9]: 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. [
[10]: P. Psenak and S. Previdi, OSPFv3 Extensions for Segment Routing, RFC 8666, Dec. 2019. doi: 10.17487/RFC8666
[11]: 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.



SDN Controller with PCEP

- Path Computation Element Protocol
  - Paths as Explicit Route Objects (ERO)
    - For segment routing this becomes Segment Routing ERO (SR-ERO)
  - Consists of Path Computation Client (PCC) and Path Computation Element (PCE)
    - The PCE pushes out the SR-EROs
    - The PCC receives SR-EROs
- SDN Controller
  - Northbound API
    - External coordination
  - Southbound API
    - Controlling SR-LSPs
      - PCEP
    - Topological information

External Network Application

Open Northbound API

SDN Controller

Open Southbound API

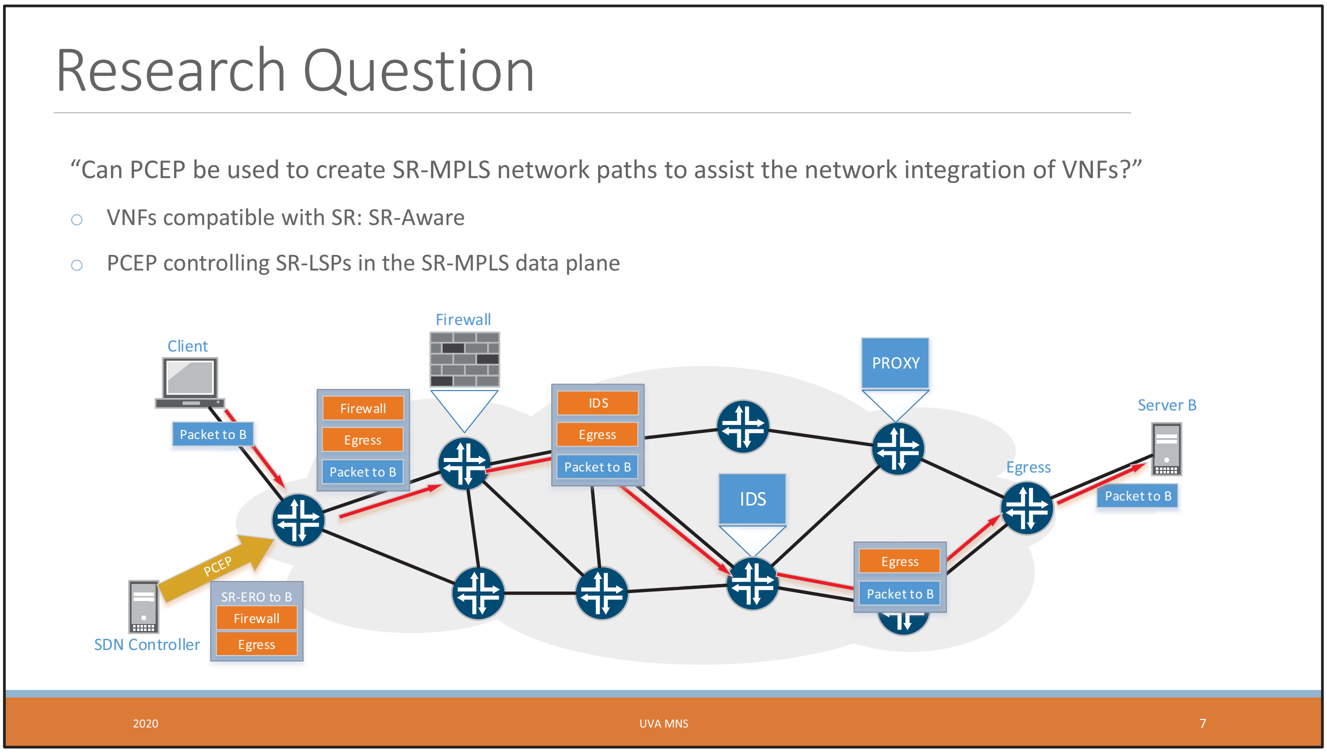
Network Infrastructure

2020 UVA MNS 6

In our research, we want to control paths, specifically the SR-LSPs, in the network, which is where the Path Computation Element Protocol (PCEP) comes in [12]. This protocol is specifically aimed at building paths in the network and it supports segment routing extensions as per RFC 8664 [13]. The protocol abstracts paths as Explicit Route Objects (EROs). For segment routing, the SR-ERO object can represent an SR-LSP. The protocol works with a client and server. The client, Path Computation Client, is the router, the element which builds or implements the actual path in the network. The server is the Path Computation Element which pushes out the EROs to the client.

To make use of PCEP, the concept of Software Defined Networking is relevant. Software Defined Networking (SDN) controllers have emerged as elements to provide automated control and abstraction over the network [14]. SDN controller functionality is two-fold: first, the controller needs a view of the network and secondly, a way to control it. To get a view of the network, there are multiple protocols available that provide the needed topological information. It can be done via an Interior Gateway Protocol (IGP) or via BGP via the BGP Link State extension. In the control part of the SDN controllers we want to use PCEP to have the SDN controller build the SR-LSPs in the network. In this fashion, the SDN controller serves as the Path Computation Element. For external coordination, SDN controllers present a Northbound API. This is useful for our research as it serves as a way to coordinate the SDN controller with the VNFs as there exist no protocols for that.

[12]: J. Vasseur and J.-L. L. Roux, Path Computation Element (PCE) Communication Protocol (PCEP), RFC 5440, Mar. 2009. doi: 10 . 17487 / RFC5440.
[13]: 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.
[14]: 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.



Research Question

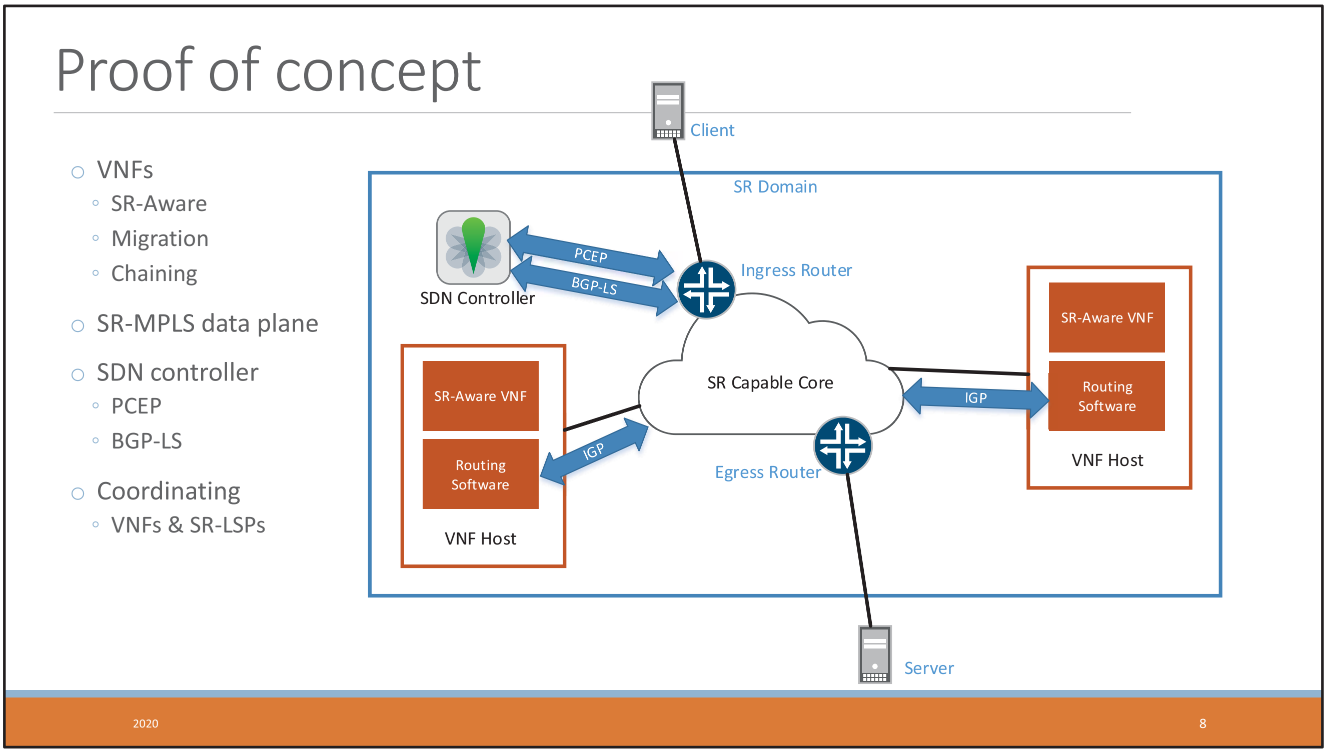
"Can PCEP be used to create SR-MPLS network paths to assist the network integration of VNFs?"

- ○ VNFs compatible with SR: SR-Aware
- ○ PCEP controlling SR-LSPs in the SR-MPLS data plane

2020 UVA MNS 7

With the technologies established the research question can be formulated, based on the goal of steering traffic through network functions in the network. The question then becomes "Can PCEP be used to create SR-MPLS network paths to assist the network integration of VNFs?"
To answer this research question with the posed technologies, some factors have to be considered. Firstly, the VNFs to be used have to be deployable in an SR domain and thus have to be compatible with SR-MPLS. This is called SR-Aware. Further more, the PCEP control of the SR-LSPs has to be compatible with the SR-MPLS data plane as well.

Combining the technologies in an example network on the slide, we can see that we have a network where we want to steer the traffic from the client to the server through the firewall and egress. We require a path through the network which will always visit the firewall. The question then is if we can use the SDN controller to push an SR-ERO object corresponding with this required route to the ingress router. The ingress router should attach the correct SIDs and the rest of the network should forward the packet based on the specified path.



Proof of concept

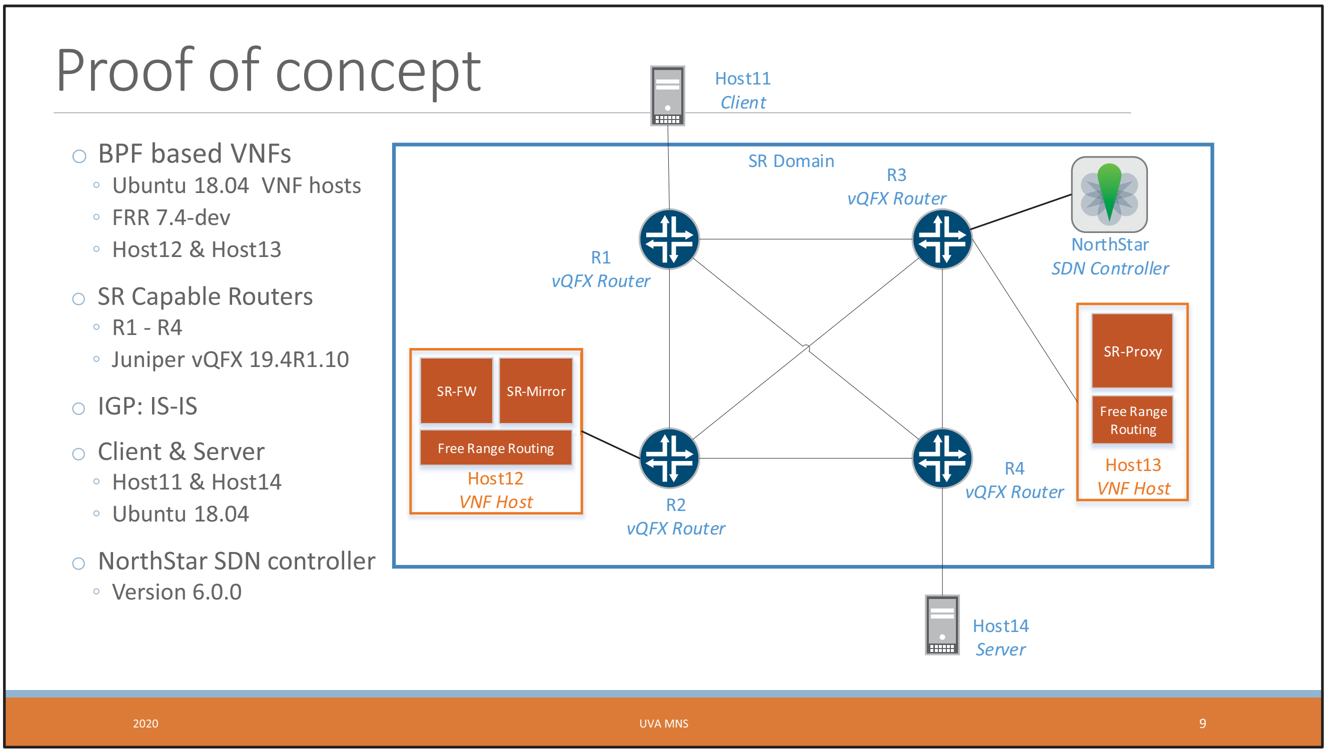
- VNFs
  - SR-Aware
  - Migration
  - Chaining
- SR-MPLS data plane
- SDN controller
  - PCEP
  - BGP-LS
- Coordinating
  - VNFs & SR-LSPs

2020

8

The proof of concept network will consist of the aforementioned components. Firstly it requires VNFs that can be used in an SR domain so these should be SR-Aware VNFs. These VNFs can then be migrated or chained in service chains. The core of the SR domain will need to consist of routers and the hosts that run the VNFs also need to participate in this network. To this end, the network needs an Interior Gateway Protocol to distribute the segment routing information with both the core of the network as well as the VNF hosts participating in the IGP. The VNF hosts will need routing software to run the IGP. Attached to the network we will have a client and server, to serve as a way to generate traffic flows through the network, which we can direct to the VNFs. To steer the traffic, SR-LSPs will have to be built in the network and to this end we will use PCEP. An SDN controller is added to utilize the PCEP protocol and this SDN controller will also need to acquire topological information via BGP-LS.

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. We accomplish this by developing a tool that 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.

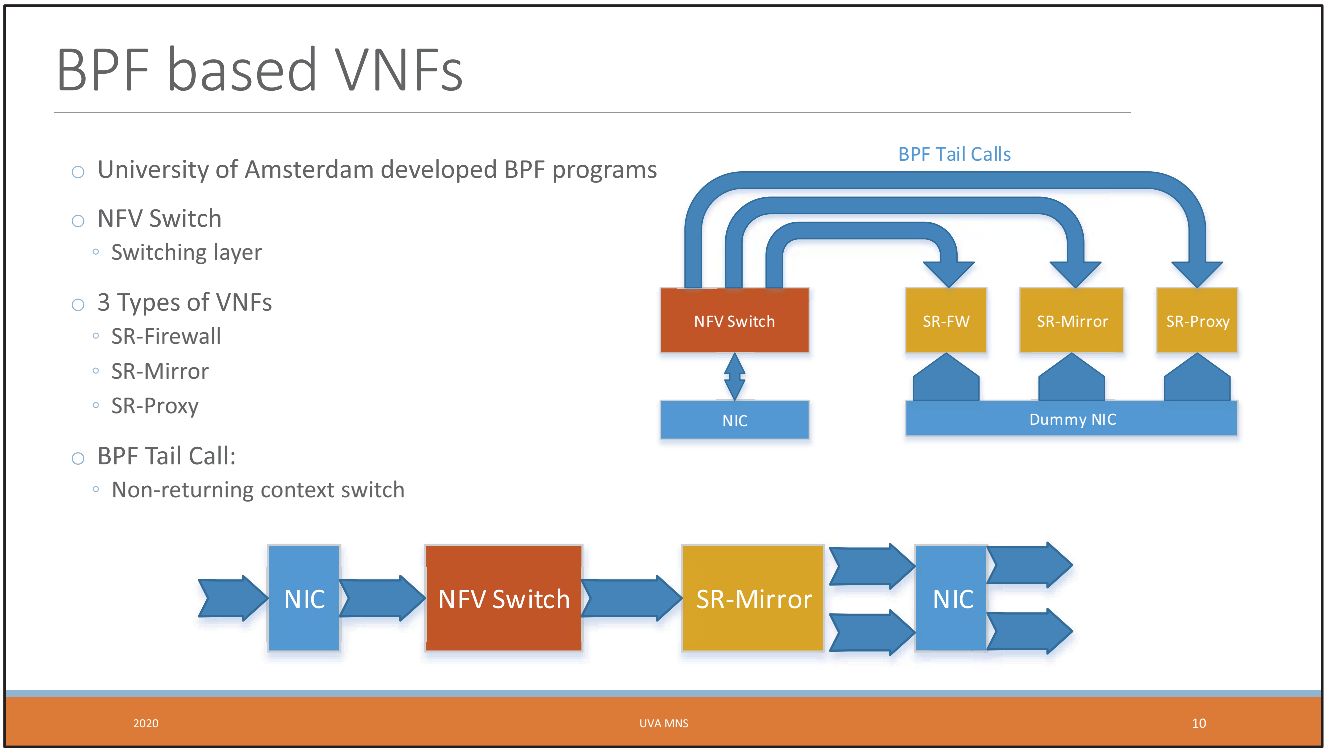


Proof of concept

- ○ BPF based VNFs
  - ◦ Ubuntu 18.04 VNF hosts
  - ◦ FRR 7.4-dev
  - ◦ Host12 & Host13
- ○ SR Capable Routers
  - ◦ R1 - R4
  - ◦ Juniper vQFX 19.4R1.10
- ○ IGP: IS-IS
- ○ Client & Server
  - ◦ Host11 & Host14
  - ◦ Ubuntu 18.04
- ○ NorthStar SDN controller
  - ◦ Version 6.0.0

2020 UVA MNS 9

With the design outlined, we built the actual proof of concept network. The VNFs used in the network are BPF based programs. These are hosted on Ubuntu 18.04 based machines. The core of the SR domain runs IS-IS as IGP to distribute the segment routing information. To allow the VNF hosts to participate in the IS-IS IGP, they run Free Range Routing, version 7.4-dev.
The core of this SR domain consists of Juniper vQFX routers, version 19.4R1.10 and these routers have the needed support for the segment routing extensions of IS-IS. The client is connected to R1 and the server to R4 and these serve to generate the traffic flows through the network. The SDN controller used to control the SR-LSP paths in the SR domain is Juniper NorthStar of which version 6.0.0 is used.



BPF based VNFs

- University of Amsterdam developed BPF programs
- NFV Switch
  - Switching layer
- 3 Types of VNFs
  - SR-Firewall
  - SR-Mirror
  - SR-Proxy
- BPF Tail Call:
  - Non-returning context switch

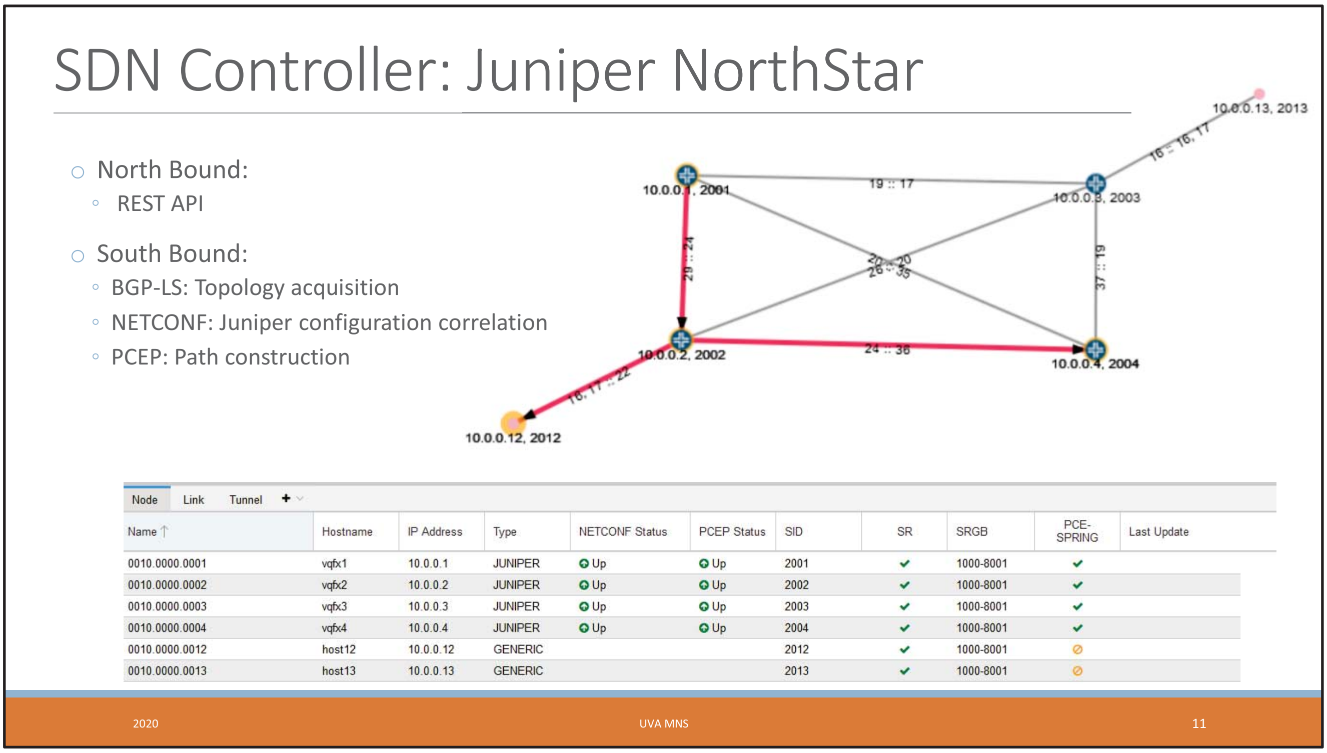
2020 UVA MNS 10

The VNFs used in the proof of concept are specially developed at the University of Amsterdam [15]. These are implemented as a special set of BPF programs to be compatible with the SR-MPLS data plane and deployed on Ubuntu based VNF hosts.
The functionality is roughly split into two parts. The NFV Switch program is the first part. This is a BPF program which is directly attached to the interface of the VNF host machine and serves to direct traffic to the VNFs when needed.
All incoming traffic is processed by this BPF program. If it is an SR-MPLS packet, the NFV Switch program will deliver the packet to the available VNF. The VNFs are the second part of this system. As they are BPF programs we attach them to an interface, however not one inline with the traffic; the dummy NIC. In this way, the NFV Switch is in control of pushing the traffic to the VNFs and the VNFs can be limited in the functionality they offer.

To deliver the traffic from the NFV Switch to the VNF, BPF tail calls are utilized. These are special BPF function calls that hand over the execution to the function and never return. Thus, once the packets are delivered to the VNF via the tail call, it is the VNFs responsibility to correctly return the packets back to the incoming interface.

There are three VNFs implemented. Firstly is the SR-Firewall, this serves as an SR-Aware firewall 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 functions as a proxy for SR-Unaware VNFs.

[15] Ł. Makowski, RON19: MPLS-SR NFV, 2020. Available at https://wiki.surfnet.nl/display/SURFnetnetwerkWiki/Research+And+Development+Projects+2019 under Segment routing and NFV



SDN Controller: Juniper NorthStar

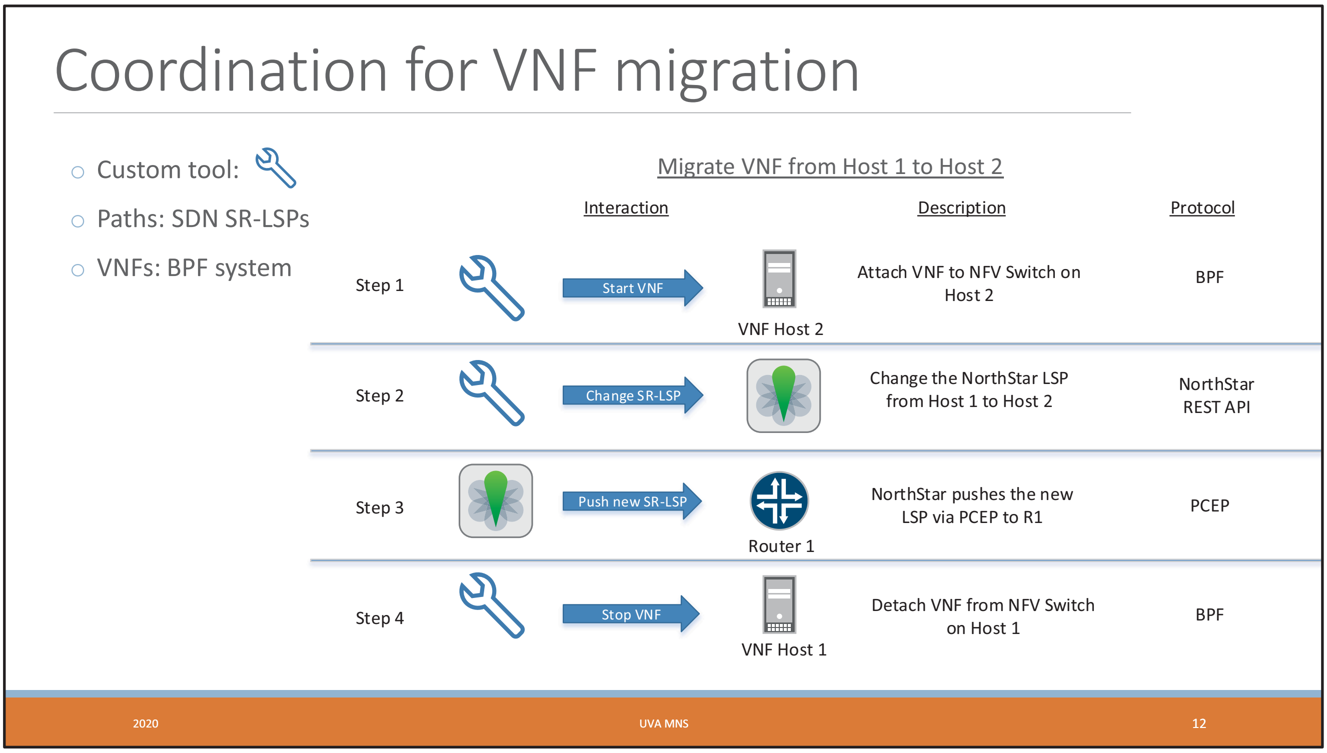
- ○ North Bound:
  - REST API
- ○ South Bound:
  - BGP-LS: Topology acquisition
  - NETCONF: Juniper configuration correlation
  - PCEP: Path construction

Name ↑	Hostname	IP Address	Type	NETCONF Status	PCEP Status	SID	SR	SRGB	PCE-SPRING	Last Update
0010.0000.0001	vqfx1	10.0.0.1	JUNIPER	Up	Up	2001	✓	1000-8001	✓	
0010.0000.0002	vqfx2	10.0.0.2	JUNIPER	Up	Up	2002	✓	1000-8001	✓	
0010.0000.0003	vqfx3	10.0.0.3	JUNIPER	Up	Up	2003	✓	1000-8001	✓	
0010.0000.0004	vqfx4	10.0.0.4	JUNIPER	Up	Up	2004	✓	1000-8001	✓	
0010.0000.0012	host12	10.0.0.12	GENERIC			2012	✓	1000-8001	⊘	
0010.0000.0013	host13	10.0.0.13	GENERIC			2013	✓	1000-8001	⊘	

2020 UVA MNS 11

The SDN controller used to control the SR-LSPs in the network is Juniper NorthStar. The two images shown are taken from the GUI of NorthStar, showing both its topological view of the network as well as more detailed node information. In the topological view it shows the loopback IP addresses, the node SIDs as well as the adjacency SIDs.
NorthStar adheres to the Northbound and Southbound split in APIs in order for its abstraction. Southbound, we configured NorthStar with a BPG-LS session with a router in the network such that it could acquire the topology. NorthStar combines this data with the NETCONF connection it needs with the routers as well. For the proof of concept network, the VNF hosts based on FRR do not support NETCONF, but the vQFX routers in the core do. Thus, NorthStar only has a connection with those routers, which fortunately is enough for NorthStar to correctly compute the topology of the network. To actually build the SR-LSPs in the network we then configure NorthStar to have a PCEP session with routers. For the Northbound connection, that is, to influence NorthStar externally, NorthStar offers the REST API which can be used to instruct NorthStar to push out SR-LSPs.

It is important to note here that NorthStar can only push out SR-LSPs via PCEP if it has a correct and complete view of the topology. Thus the BGP-LS and NETCONF connections are needed for this to function.



Coordination for VNF migration

- Custom tool: 🔧
- Paths: SDN SR-LSPs
- VNFs: BPF system

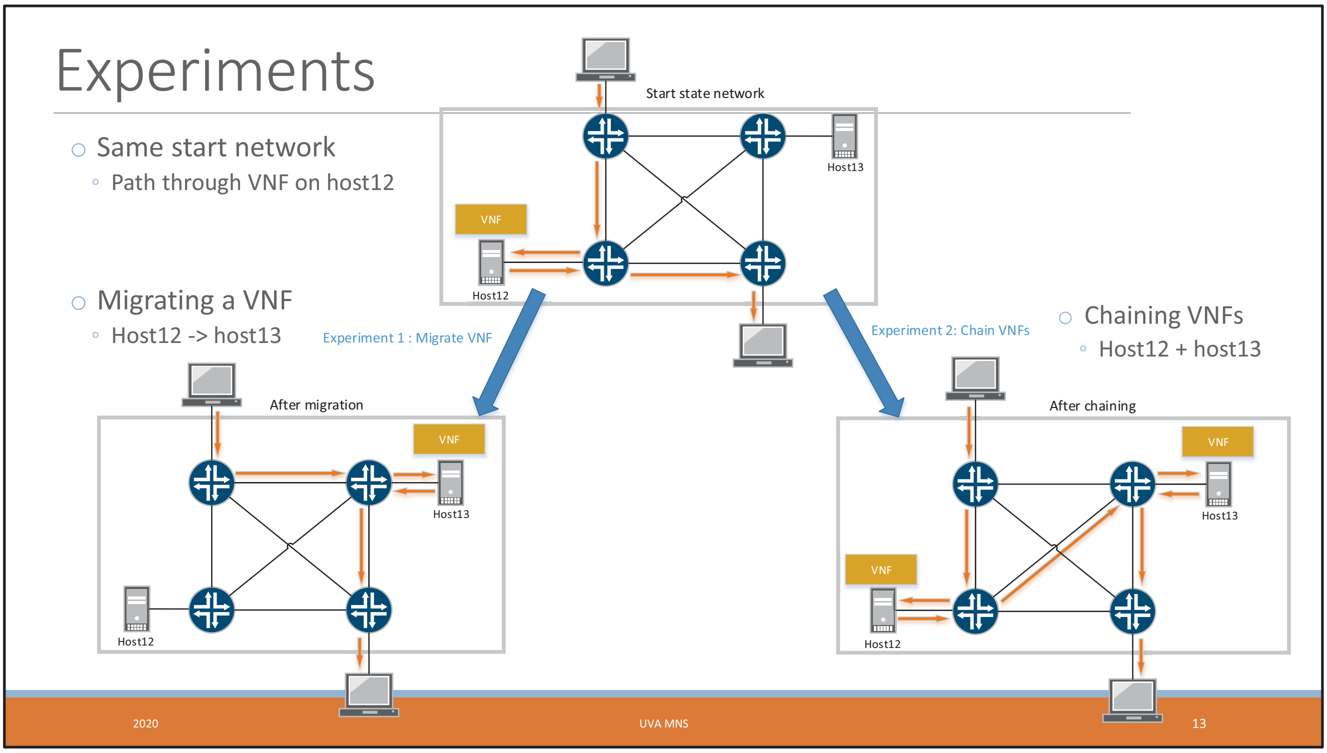
Migrate VNF from Host 1 to Host 2

	Interaction			Description	Protocol
Step 1	🔧	Start VNF	VNF Host 2	Attach VNF to NFV Switch on Host 2	BPF
Step 2	🔧	Change SR-LSP		Change the NorthStar LSP from Host 1 to Host 2	NorthStar REST API
Step 3		Push new SR-LSP	Router 1	NorthStar pushes the new LSP via PCEP to R1	PCEP
Step 4	🔧	Stop VNF	VNF Host 1	Detach VNF from NFV Switch on Host 1	BPF

2020 UVA MNS 12

Lastly, the components need to be coordinated to ensure it functions as a cohesive system. This involves the instantiation of the VNFs as well as building the SR-LSPs so they correspond with the VNFs. We accomplished this via an external element which took the form of a specially developed tool for this research. The tool uses the APIs available from both NorthStar as well as the BPF VNF system. This tool is used from an external machine which has access to the VNF hosts as well as NorthStar.

To walk through the interactions, we can take VNF migration for example. The idea here is that we have a VNF on a particular host, but we want to migrate it to another. Perhaps because it has more computing resources or the first one has to go down for maintenance. The first step is to instantiate a new VNF of the same type on the target host. Then, using the REST API exposed by NorthStar, the changed SR-LSP data is pushed. NorthStar itself then uses the PCEP protocol to instruct the ingress router R1 to change the SR-LSP segment stack. At that point, the traffic will have the updated label stack, now traversing host 2, with its VNF. Finally the old VNF on the first host can be spun down as it is no longer being used.



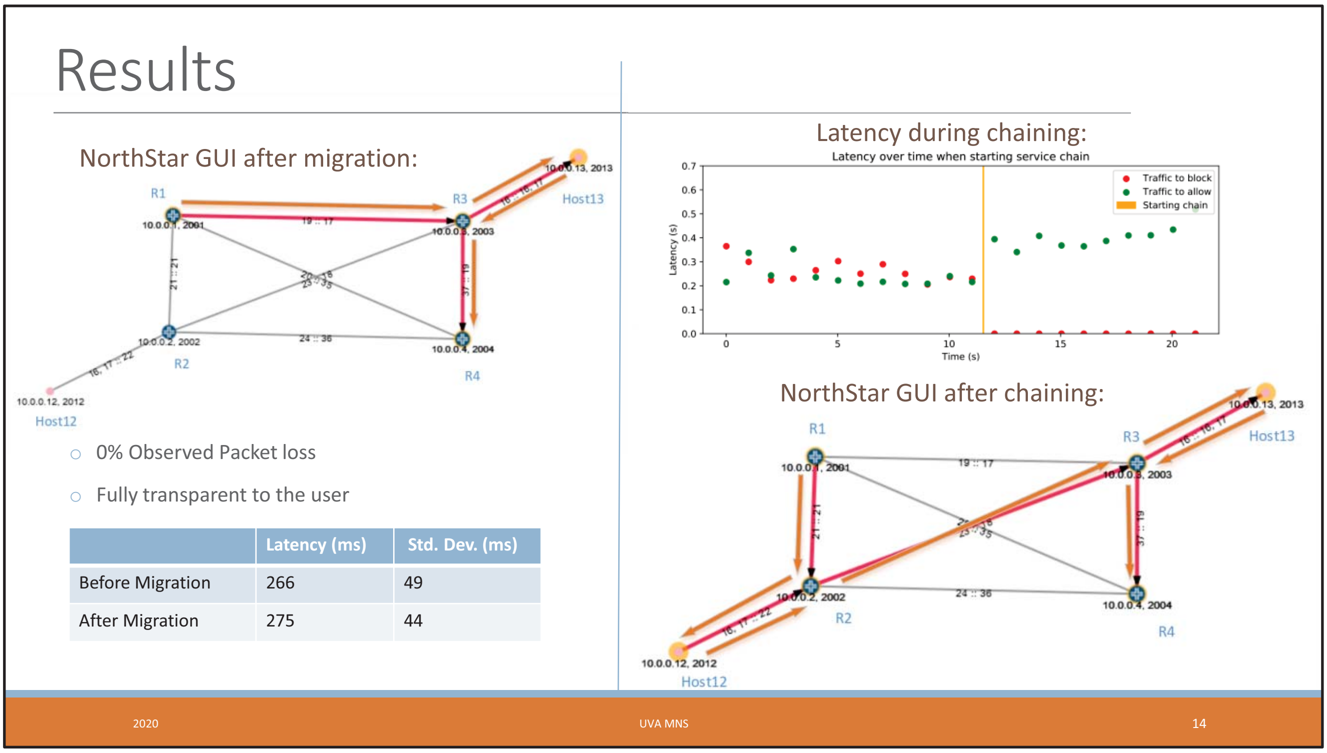
Experiments

- ○ Same start network
  - ◦ Path through VNF on host12
- ○ Migrating a VNF
  - ◦ Host12 -> host13
- ○ Chaining VNFs
  - ◦ Host12 + host13

2020 UVA MNS 13

On the built proof of concept, we can run various experiments to evaluate its functionality. In the starting situation, there is an SR-LSP configured such that the traffic from the client to the server is going through the VNF on host12.
In this starting situation we can observe the traffic in the VNF on host12, flowing correctly through the network from client to server.
The first experiment then is to migrate the VNF from host12 to host13. On host12 we start with a SR-Firewall VNF, which we want to move to host13. We expect this to be fully transparent to the user, with no packet loss observed.

The second experiment is the construction of a service chain of multiple VNFs. In this experiment, initial VNF on host12 is chained with another VNF on host13. On host12, the initial VNF is a SR-Firewall which allows TCP traffic targeting port 80. On host13, the chained VNF will be another SR-Firewall, that will drop only TCP traffic targeting port 80. This allows us to easily observe if the chain has been successfully instantiated, as that kind of particular traffic that could previously traverse the network will now be blocked. Furthermore, other types of traffic should be allowed, for example, traffic targeting port 433 should be uninterrupted.



Results

NorthStar GUI after migration:

- ○ 0% Observed Packet loss
- ○ Fully transparent to the user

	Latency (ms)	Std. Dev. (ms)
Before Migration	266	49
After Migration	275	44

Latency during chaining:

NorthStar GUI after chaining:

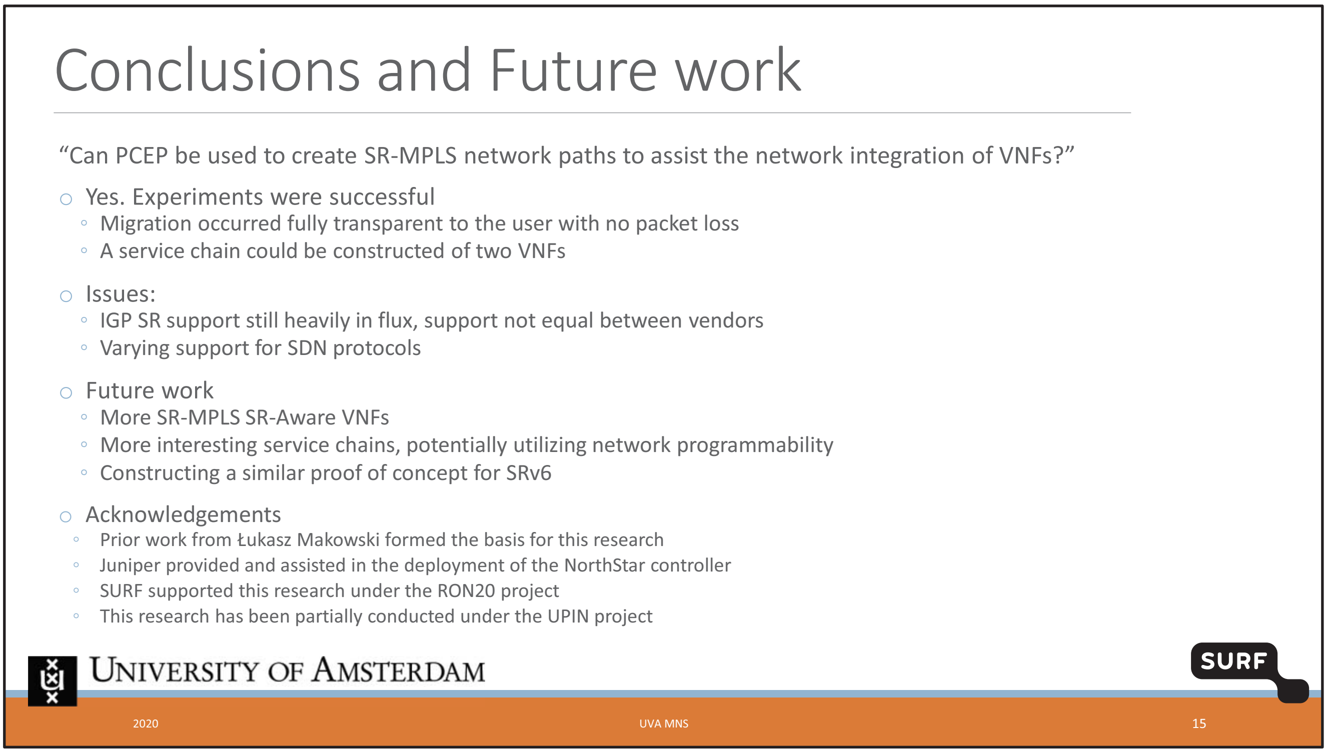
2020 UVA MNS 14

The experiments conducted have various dimensions in which they are evaluated.

Both the migration and chaining experiments change the SR-LSPs via NorthStar, thus in NorthStar we should be able to see the changed paths as we expect. Both the pictures show an annotated version of the NorthStar GUI.
For the migration experiment, we indeed see the path now goes through host13 and thus through the VNF on it. Similarly, for the chaining experiment, we now see the path first heading to host12 and then to host13 before heading to the egress router R4.

For the migration experiment we also had a continuous stream of traffic from the client to the server and did not observe any packet loss or significant change in latency during the migration or after its completion. Monitoring the VNF on the host13 itself we correctly observed it handling traffic once the migration was conducted. The migration has been successfully conducted.

For the chaining experiment, we do expect the traffic to be impacted as the second VNF blocks the traffic that the first one allows. The graph shows traffic that will be blocked by the chain as well as traffic that should be allowed. Initially both types of traffic are allowed through the network, through the first firewall. Once the service chain is built, the latency drops to 0 for the traffic that is no longer allowed, denoting missed replies for the outgoing traffic from the client. The traffic that is still allowed continues to flow correctly, but with a higher latency due to the longer network path. On the VNF on host13 we observe the traffic matching the rule and being dropped as intended, confirming the behavior. The service chain is functional as intended.



Conclusions and Future work

"Can PCEP be used to create SR-MPLS network paths to assist the network integration of VNFs?"

- Yes. Experiments were successful
  - Migration occurred fully transparent to the user with no packet loss
  - A service chain could be constructed of two VNFs
- Issues:
  - IGP SR support still heavily in flux, support not equal between vendors
  - Varying support for SDN protocols
- Future work
  - More SR-MPLS SR-Aware VNFs
  - More interesting service chains, potentially utilizing network programmability
  - Constructing a similar proof of concept for SRv6
- Acknowledgements
  - Prior work from Łukasz Makowski formed the basis for this research
  - Juniper provided and assisted in the deployment of the NorthStar controller
  - SURF supported this research under the RON20 project
  - This research has been partially conducted under the UPIN project

UNIVERSITY OF AMSTERDAM SURF

2020 UVA MNS 15

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 to this purpose?" The proof of concept built in order to answer the research question combines the segment routing technology with PCEP supported by SDN. The proof of concept demonstrates that network paths targeting specific VNFs can be built in an SR-MPLS network. Node SIDs can be used successfully to represent VNFs in network paths pushed out via the PCEP protocol.
The BPF based VNFs could be successfully integrated in network paths controlled by PCEP. In the proof of concept network, migration and chaining of VNFs, supported by PCEP has been demonstrated, providing a positive answer to the research question.

There are two points with regards to implementing the technologies in a network. Firstly, the Segment Routing support in the various IGPs is still heavily influx and support is far from equal or mature across vendors. Secondly, the integration of the SDN controller to utilize PCEP depends on more protocols to be supported. For the case of Juniper's NorthStar, the requirement of NETCONF makes integration of FRR more difficult.
Future work could be focused on creating more SR-Aware VNFs, specifically to create more varied service chains. Furthermore, a similar research question could be posed with regards to SRv6, to contrast it with this work. Lastly, the technologies from this PoC could be applied in a production network to expose any potential issues when applying it at a larger scale.

This research was made possible thanks to the prior work and support from Łukasz Makowski. Juniper supported this research by providing the NorthStar controller software and assisting in its deployment. This research has been partially conducted under the RON20 project with support from SURF and under the UPIN project.