



UvA-DARE (Digital Academic Repository)

User Controlled Routing Exploiting PCEPS and Inter-domain Label Switched Paths

Boldrini, L.; Bachiddu, M.; Koning, R.; Grosso, P.

DOI

[10.1007/978-3-031-56950-0_39](https://doi.org/10.1007/978-3-031-56950-0_39)

Publication date

2024

Document Version

Final published version

Published in

Proceedings of the Second International Conference on Advances in Computing Research (ACR'24)

License

Article 25fa Dutch Copyright Act (<https://www.openaccess.nl/en/policies/open-access-in-dutch-copyright-law-taverne-amendment>)

[Link to publication](#)

Citation for published version (APA):

Boldrini, L., Bachiddu, M., Koning, R., & Grosso, P. (2024). User Controlled Routing Exploiting PCEPS and Inter-domain Label Switched Paths. In K. Daimi, & A. Al Sadoon (Eds.), *Proceedings of the Second International Conference on Advances in Computing Research (ACR'24)* (pp. 465-478). (Lecture Notes in Networks and Systems; Vol. 956). Springer. https://doi.org/10.1007/978-3-031-56950-0_39

General rights

It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations

If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: <https://uba.uva.nl/en/contact>, or a letter to: Library of the University of Amsterdam, Secretariat, P.O. Box 19185, 1000 GD Amsterdam, The Netherlands. You will be contacted as soon as possible. University of Amsterdam (<https://dare.uva.nl>)



User Controlled Routing Exploiting PCEPS and Inter-domain Label Switched Paths

Leonardo Boldrini¹(✉), Matteo Bachiddu², Ralph Koning³, and Paola Grosso¹

¹ MNS Group, University of Amsterdam, Amsterdam, The Netherlands
`{l.boldrini,p.grosso}@uva.nl`

² NETGroup, Politecnico di Torino, Turin, Italy

³ SIDN Labs, Arnhem, The Netherlands
`ralph.koning@sidn.nl`

Abstract. The UPIN project tackles the security issue of the Internet at its root, by providing more transparency and control over the network to the end user. Traffic Engineering is required to create paths across multiple domains following constraints set by the user. We present here the feasibility of using the Path Computation Element Communication Protocol Secure (PCEPS) to accomplish the goal of traffic steering - hence control - across multiple domains. Specifically, we leverage on IPv4 Segment Routing (SR) and a modified version of the Netphony Path Computation Element (PCE) to build a multi domain Label Switched Path (LSP) according to a user's request. We present a proof of concept where we verify the correctness of the operation of the modified PCE code.

Keywords: PCEPS · PCEP · UPIN · Segment Routing · Path Control

1 Introduction

The Internet is today a global infrastructure that supports a wide range of services and products on which all companies and governments depend on [1]. This dependence is often based on systems produced and managed by entities in other countries, which generates in several circumstances political and economical concerns. As a result, we are observing the emergence of more concerns regarding digital autonomy and greater attention over control of data. The Responsible Internet was proposed to address these problems and provide a higher degree of trust and sovereignty for users of the Internet [2]. Ideally users should understand what the network is capable of, what problems may arise with their data in transit, and ultimately, make responsible decisions on how should the network deal with their data.

The User-driven Path verification and control in Inter-domain Networks (UPIN) project [3] addresses these needs. Its main goal is to provide more transparency and control over the network to end users.

The research we present here focuses on the control aspects of the communication, and in particular, on the problem of inter-domain control. In a multi domain setup, different operators are in charge of setting up different portions of a path. If the user requires from the network specific services, all network operators need to decide together how to steer traffic in order to satisfy this request.

To accomplish its goal the UPIN framework that we present here adopts and integrates three technologies that have seen significant attention in the networking research community, as attested by the numerous publication on this topic: Segment Routing (SR), Path Computation Elements (PCEs) and the associated Path Computation Element Communication Protocol Secure (PCEPS) [4], that exploits Transport Layer Security (TLS) across domains. We will describe these in more detail in Sect. 2.

With this work we intend to answer the following questions: “Can we accomplish SR-MPLS path computation across multiple domains using Path Computation Elements (PCEs)?”, and more specifically: “Can PCEPS be used for this purpose?”.

Our contributions are as follows:

- we evaluated a number of PCEs implementation to identify the best suited; we concluded that Netphony is the implementation to adopt;
- we modified the communication between PCEs to support the creation of paths across domains by defining and exchanging a set of global labels in a secure fashion by means of PCEPS;
- we evaluated our changes in a prototype testbed by creating SR paths across two domains.

Our implementation and subsequent experiment prove that it is possible to adopt SR-MPLS and PCEPS to enable user control over network services.

2 Background

In order to understand our work, we need to briefly introduce the technologies we rely on, namely Segment Routing, Path Computation Elements and PCEPS.

2.1 Segment Routing

In the current Internet architecture, every router makes forwarding decisions based on each packet’s destination. In contrasts source routing is a paradigm where “... the source of internet packets specifies the complete internet route.” [5].

The source routing concept has been implemented in the Segment Routing paradigm [6]. The actual path to be taken is specified as a list of segments, via segment identifiers (SIDs). Segments can be of four types: 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 represents an IP prefix, anycast segments are used for anycast traffic. In this

paper we use node segments that represent routers. Segment Routing helps network operators with distributing the traffic more efficiently and makes rerouting possible in case of failures. We rely on SR to provide users with a mechanism to identify services running on certain network nodes that they want and steer traffic through them, e.g. users can specify that their traffic needs to be evaluated by a firewall or an intrusion detection system.

Segment Routing can be implemented as an extension of MPLS in SR-MPLS [7,8]. In SR-MPLS, the segments are specified by the MPLS labels. An MPLS label is 20-bit long and it can represent any of the SID types. When an MPLS label represents a specific SID, this information can be distributed within a domain by an Interior Gateway Protocol (IGP). A Segment Routed Label Switched Path (SR-LSP) is a path through an SR-MPLS network. This path is unidirectional. We had presented a single domain implementation of SR-MPLS to steer traffic according to specific constraints in [9].

Because we use an IGP to distribute SID information within a domain, none of this information is available for other domains, hence it is not possible to create a multi domain SR-LSP following this method. Furthermore, the pool of MPLS labels available within a domain can be the same for each domain, so the simple solution of distribution of SIDs across domains is not feasible. This is also a security feature for network operators, as they don't want to expose to other domains their SIDs and what they use them for. For all of these reasons, we set out to identify ways to construct a multi domain SR-LSP by signaling between Path Computation Elements that belong to different network domains.

2.2 Path Computation Elements

A Path Computation Element (PCE) is a network element that is capable of computing a network path or route based on a network graph, and of applying computational constraints during the computation [10]. This follows the paradigm of Constraint-based Shortest Path First (CSPF). The information based on which a PCE computes a path is stored in the Traffic Engineering Database (TED), which contains the topology and resource information of the domain. The TED may be fed by Interior Gateway Protocol (IGP) extensions or potentially by other means.

A PCE can be either stateful or stateless; in a stateful setup, the PCE keeps memory not only of topology and resource information contained in the TED, but also of all the paths it has already computed, and will remove this information if the path is no longer needed or available. The drawback of this approach is that all information available to the PCE needs to be up to date, resulting in heavy control plane overhead. On the other hand, a stateless PCE does not keep state of the paths it has already computed, it instead uses the information available in the TED to compute a new path. Every new path is then processed independently of the others. We use stateful PCEs because we need to keep track of already computed local paths in order to signal to other domains which portion of a multi domain path to use. PCEs traditionally communicate between each other using Path Computation Element Communication Protocol (PCEP). More recently, PCEPS has been introduced as a more secure version of PCEP [4].

2.3 PCEP and PCEPS

The Path Computation Element Communication Protocol (PCEP) allows to communicate network paths [11] among PCEs. It was originally designed for MPLS and Generalized Multi-Protocol Label Switching (GMPLS) networks to facilitate communications between PCEs. It identifies the operation between a PCE and a client, called Path Computation Client (PCC). PCEP defines a specific set of requests. Each session consists of a Transmission Control Protocol (TCP) connection that can be secured via SSL or TCP-MD5. Table 1 summarizes the types of PCEP messages available and that we use in our Proof of Concept.

Table 1. Message types available in PCEP.

Message type	Description
PCReq	The PCC requests a path to the PCE
PCRep	The PCE responds to the PCC with a path
PCUpd	The PCE updates a path on the PCC
PCNtf	The PCE sends a notification to the PCC
PCErr	The PCE or the PCC sends an error

In PCEP, a Label Switched Path (LSP) is set up by a PCC sending a path computation request. The PCE response message contains the path or an error message, in case no path was found. The PCE can send an update message containing a new label stack for an LSP in case it needs to update the corresponding path. If the connection between the PCE and the PCC is lost, the PCC will minimize traffic disruptions. If it can no longer reach the PCE, it will remove the installed paths.

Our goal is to push SR-LSPs to the routers in the network. The path is pushed out by the PCE to the PCC as an Explicit Route Object (ERO). In PCEP, support for segment routing was added as per RFC 8664 as segment routing EROs (SR-EROs) [12]. Previous work [13, 14] have demonstrated the viability of using PCEP and BGP-LS in the MPLS data plane to construct paths, making them a good starting point for our implementation.

The Path Computation Element Communication Protocol Secure (PCEPS) was presented in RFC 8253 as an addition of Transport Layer Security (TLS) to provide a secure transport for PCEP [4]. TLS provides a secure transport channel above layer 4 bringing many security features: peer authentication, message confidentiality, message authentication and integrity, protection against replay and filtering attacks [15]. Only one new message type, other than the ones presented in Table 1, needs to be added in order to implement PCEPS, which is the StartTLS message. The other regular PCEP messages are sent over the new established TLS channel. New errors are encapsulated inside the PCErr message in order to signal problems occurring in the TLS connection.

The initiation of a PCEPS session between a PCE and PCC happens in 4 phases:

- initialization and establishment of a TCP connection; this phase is no different from what happens in regular PCEP;
- both peers (PCE/PCC) send a StartTLS message to initiate TLS;
- negotiation of TLS parameters and establishment according to the TLS procedure;
- PCEP messages can now be sent in a secure fashion over TLS, starting with PCEP Open messages.

By modifying all the PCEs in our implementation to run PCEPS, we ensure that every message that is exchanged across different network domains benefits from the security features that come with TLS. This way, we can provide a secure transmission of the SR-EROs that cross inter-domain links and carry information about what path their data will go through.

Although PCEPS was introduced in RFC 8253 as a more secure version of PCEP, it has not been standardized neither widely used as PCEP is not usually deployed across different domains. However, our implementation requires all the security improvements that PCEPS brings in order to send sensitive information for users across domains.

3 The UPIN Framework

The technologies we described briefly in the previous section support the creation of the UPIN [3] framework.

The UPIN project provides the concrete implementation for the Responsible Internet, namely transparency, controllability and accountability for the users of the network infrastructure [1, 3, 16].

The UPIN framework consists of the following components: a Domain Explorer, Path Controller, Path Tracer, Path Verifier, and Frontend [3, 16]. Every component plays a specific role in managing a domain. The Domain Explorer obtains metadata about properties of the network, such as security and environmental details. It stores information on the nodes in the network. The Path Controller sets forwarding rules based on what the user has asked. The Path Tracer gathers measurements on the traffic in the UPIN domain. Its goal is to store important details for the possible verification. The Path Verifier analyzes measurements from the Path Tracer to verify whether the desires of the user are satisfied. If the path traverses a non-UPIN enabled domain, there is no way for the Path Verifier to be certain whether the intent is satisfied over the full path. The Frontend allows for the communication between the user and the domain. A useful implementation of the Frontend within the scope of UPIN can be found in [2]. Through the Frontend, a user can see what services or Virtual Network Functions (VNFs) are available in the network, and which paths can be taken to reach the desired destination. A user can then request a specific path to be

taken, following a set of constraints that can include for example which VNFs to traverse (e.g. a firewall), or which jurisdictions the whole path can traverse.

In this paper we use PCEs to perform the functions of the Path Controller of the UPIN framework.

We leverage on SR and PCEs so that we can steer traffic of every user through different paths, following their requests. We focus on the path creation, including the intermediate elements to be traversed, therefore addressing the real goals of a Responsible Internet.

4 PCE Evaluation

There are many PCE implementations available and we wanted to evaluate the one most suitable for adoption in UPIN. We considered the following:

- *OpenDaylight(ODL)*. The PCE developed in this automation platform supports Segment Routing, stateful paths, binding labels, objective functions and include route objects;
- *NorthStar*. The SDN controller developed by Juniper Networks contains a PCE that supports stateful paths, binding labels and segment routing extensions;
- *Netphony-PCE*. The PCE developed by Telefonica supports segment routing extensions, stateful paths and is open source.
- *ONOS*. This is an open-source network operating system that can provide the control plane in an SDN. It supports the PCEP southbound interface to communicate to network devices.
- *IOS XR*. In this network operating system developed by Cisco there is support for version 2 of PCEP and the relevant SR extensions.

We evaluated all of the above with respect to a number of essential features for inclusion in UPIN.

- **Open source**: we want access to the code for extension possibilities. The main extension we want to implement is the use of PCEPS for path creation across domains, as well as the security that comes with TLS. Open source is the ideal option but we evaluated also easily obtainable closed source PCEs;
- **Hierarchical-PCE (H-PCE)**: for inter-domain path creation we want to avoid to have a central PCE that controls all domains. H-PCEs introduce a child-PCE and a parent-PCE, where the child-PCE is responsible for intra-domain control, while the parent-PCE takes care of inter-domain PCE communication;
- **Stateful**: a stateful PCE will maintain knowledge of the paths that were computed in the past; this allows us to propagate back path requests across the domain to create the full path;
- **SR support**: our implementation of UPIN relies at the moment on VNFs that are reachable by their label number; hence we need support for SR to build a SR-MPLS multi domain path;

- **Protocol support:** we want the PCE to support traffic engineering protocols such as OSPF-TE or ISIS-TE, as well as BGP-LS. This is needed to communicate information and instructions to the routers.

Table 2 shows the features supported by the PCE implementations we considered. Based on this evaluation, we decided to use Netphony as base PCE implementation. Netphony has most features we need and, being open source, it allows us to implement the RFCs and extensions that we require, namely the implementation of PCEPS. OpenDaylight and ONOS could also be an option; they both miss support for H-PCE but this could be potentially implemented; however the code base for both is larger than Netphony-PCE and more complex to manage.

Table 2. PCE evaluation based on the required features.

	ODL	Northstar	Netphony	ONOS	IOS XR
Open source	Yes	No	Yes	Yes	No
H-PCE	No	No	Yes	No	No
SR support	Yes	Yes	Yes	Yes	Yes
Stateful	Yes	Yes	Yes	Yes	Yes
OSPF-TE	No	Yes	Yes	Yes	Yes
BGP-LS	Yes	Yes	Yes	Yes	Yes
ISIS-TE	No	Yes	No	Yes	Yes

Initially, Netphony implemented older drafts of the multiple RFCs about PCEs. However, the backend library that this PCE uses for implementing various protocols, called `netphony-network-protocols`, has a more recent development version. We implemented the non-draft RFCs that we needed to be able to communicate with FRR, as that is the software of our routers. This process led us to change certain objects that were either moved or renamed from drafts to RFCs. As an example, the Type Length Value (TLV) encoding of sub-objects in the PCEP messages got overhauled in the standardized RFCs [11, 17].

5 Inter-domain Paths with Global Labels

SR-MPLS requires the definition of a block range for the SID selection within a single domain. To set up a multi domain path, we need also to define a range of global labels which will be exchanged only across the inter-domain links. This range needs to be disjoint of the single domain range, to ensure no ambiguity in path selection.

Global labels sent across domains carry information on paths used by users and this might have privacy implications. Therefore, securing the exchange of

these labels was a primary requirement of our setup. This is the main reason that led us to implement PCEPS for inter-domain communication.

Each domain configures a static export for the whole global range towards the other domain. A PCE can then build a path using one of these global labels. The global labels will be replaced by the locally significant labels once the packets reach the next domain.

We implement an SR-MPLS algorithm in Netphony based on RFC5441 [18] and an Internet-draft on PCEP extensions for stateful Inter-Domain tunnels [19]. RFC5441 discusses how to perform the Backwards Recursive Path Computation (BRPC) procedure. It discusses how paths are computed through multiple domains, as well as the relevant extensions needed to the PCEP. [19] extends stateful paths to inter-domain deployments. In all these scenarios, the domain discovery and resolving is left to the implementation. This led us to delve into related works and propose our idea.

RFC8685 [20] proposes that PCEs communicate via a Hierarchical Path Computation Element (H-PCE). They propose the use of one central parent PCE that connects to each child PCE. This method creates a single point of failure and assumes one entity in control of the whole network. We propose instead to use a distributed setup, where each PCE only has to maintain a connection to the PCEs of the neighbouring domains.

Segment Routing Traffic Engineering (SR-TE) policies are stateful. This means that we need to keep a list of policies that are active. With our approach, all this needs to be done in the PCE.

In our implementation, when a PCE receives a computation request from a PCC, the PCE resolves the source and the destination. If one of them is not present in the TED, a *no path possible* error is sent back to the PCC. If both endpoints are present in its domain, the PCE computes the local path.

If the destination is not in the local domain but exists in the TED, and the TED has an entry to reach the PCE of the corresponding domain, a request will be forwarded to the external PCE via PCEPS. When a domain forwards a request, it waits for a response. If it gets a *no path possible* response, this is sent back to the requestor. If it receives a partial path, it computes a path from the start to the router that borders the other domain and appends this to the received path. This total path gets sent back to the requestor. This happens in compliance with the Backwards Recursive Path Computation as described in [18].

If the PCE has no stored information on either the source nor the destination, but both are present in the reachability manager, the PCE forwards the request to the PCE that contains the destination node, and then sends back the partial path to the requestor.

Our algorithm takes the shortest route possible between two nodes. It performs the shortest path algorithm on the graph to get a node list. For each node in this node list, we perform a lookup in the TED to find the corresponding SID.

6 Proof of Concept

We built a proof of concept to verify that our implementation worked correctly and was able to build multi-domain paths while exchanging only secure PCEPS messages across domains.

Our setup topology is shown in Fig. 1. This includes two domains, controlled by one PCE each. A domain is composed of four SR capable routers, a client that acts as a host to generate and receive traffic, and the modified Netphony PCE. We use the latest version of Netphony PCE: 1.3.3. For the routers, we use Free Range Routing version 8.3.1, which at the time of testing was the latest available. Because we haven't modified FRR, routers in our setup use regular PCEP and not PCEPS. The clients are hosted on Ubuntu 22.04 based machines.

We focused our work on the SR-MPLS data plane, the implementation of Segment Routing in IPv4. Therefore, in our setup, MPLS labels will correspond to SIDs and a path will consist of a set of these labels [9]. In [9] we also investigated how we can map these SIDs to VNFs present in one domain, loaded in specific hosts. This allowed us to compute a path where traffic was going through VNFs present in one domain. The setup that we present now is compatible with this feature, but for the sake of this multi-domain proof of concept, we haven't loaded any VNF in our system. SR can be used with IPv6 as well and that remains for now a future direction of research.

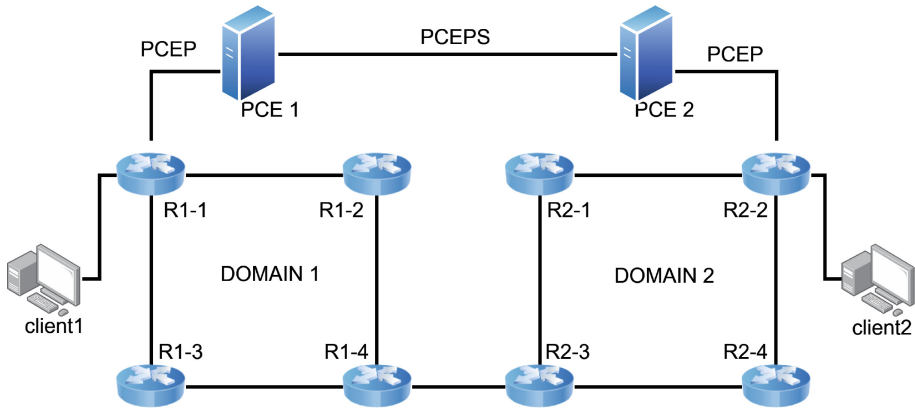


Fig. 1. Setup topology with two domains, each containing four routers and its own PCE. The two PCEs connect directly and exchange PCEPS messages on this link.

In our proof of concept, once the TLS communication has been established, all PCEs need to support two types of requests:

- local destination requests, i.e. the PCE receives a path request for an endpoint that is located in its domain;
- remote destination requests, i.e. the request received by the PCE is for a destination in a different domain.

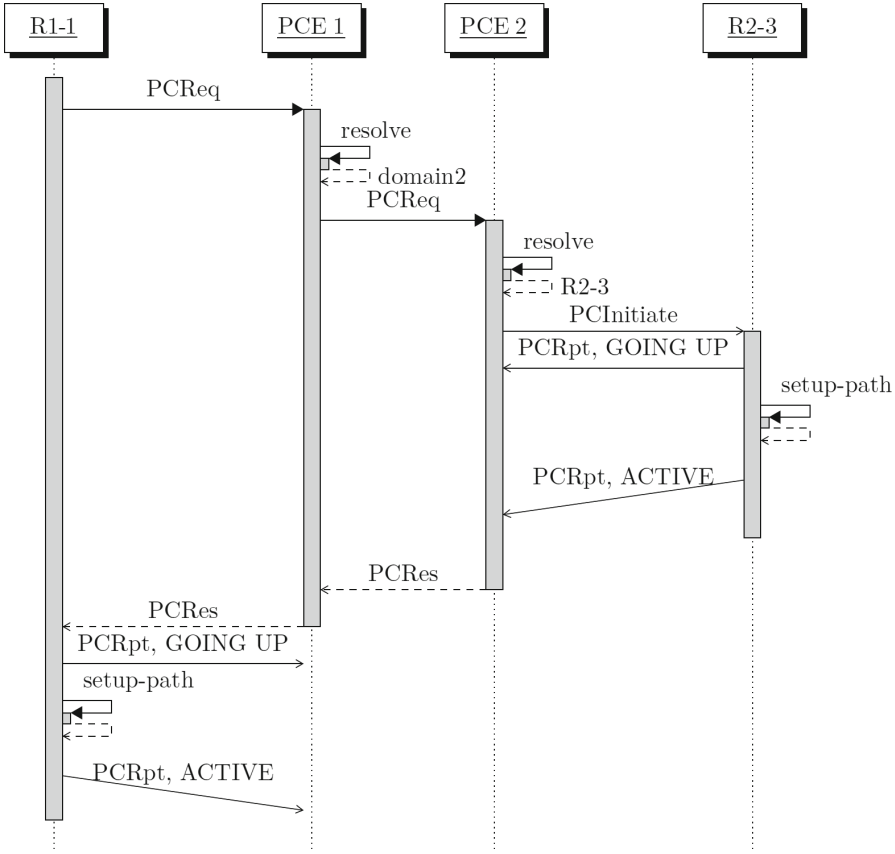


Fig. 2. PCEP and PCEPS messages exchange for multi domain path provisioning. Routers only support PCEP, while PCEs support both and use PCEPS between each other.

All our software, configuration of routers and the code of the PCEs is available and can be found in the following repository [21].

The first exchange of messages across domains happens between the PCEs, as described in 2.3. Once the TLS establishment between the PCEs is completed, PCEPS messages can be exchanged. Because FRR does not support PCEPS, we implemented in our modified Netphony PCE an option for allowing the use of regular PCEP with legacy devices that require it. Our PCEs then establish a TLS channel between them and use regular PCEP with routers in their own domains. We analyze now the PCEP and PCEPS messages that are exchanged to create the multi domain path.

When a router needs to reach a destination in a remote domain, it sends a *remote destination request* to its own PCE and this one forwards the request to the corresponding remote PCE on top of the TLS communication. The remote

PCE receives it as a *local destination request* as this destination is present in its own domain. The receiving PCE then provisions a global label at its border router and sends this back to the requesting domain. The global label acts as a stitching label: it is appended at the egress router of the first domain and removed and modified at the ingress router of the following domain.

Our goal is to steer traffic from client1 in Domain 1 to client2 in Domain 2 through inter-domain SR-LSPs that are created by the exchanging secure PCEPS messages between the PCEs and regular PCEP between a PCE and its routers. Within a domain, we need to use an IGP which enables passing the segment routing information between local network elements. The two clients are not in the same domain, so when client1 sends the first packet to R1-1, the only router it is directly connected to, R1-1 has no information on how to reach client2. In order to set up this path, there needs to be communication between R1-1 and PCE 1, between R2-3 and PCE 2, as well as between the two PCEs. This is shown in Fig. 2 where we illustrate the whole chain of messages needed to set up such a path across domains. The chain of messages is illustrative also for other cases where source and destination clients attach to different routers than the ones in our test setup.

The first PCEP message (*PCReq*) is sent from R1-1 to the PCE of its domain, PCE 1. Note that here R1-1 acts as a PCC. PCE 1 resolves the endpoint location and determines that the endpoint is in Domain 2. This corresponds to the *remote destination request*. In this case PCE 1 initiates a PCEPS *PCReq* to the PCE of the destination domain, PCE 2, over a TLS channel. This will be a *local destination request* from the perspective of PCE 2.

PCE 2 resolves the endpoint location, it computes the path that goes from the Area Border Router (ABR) R2-3 to R2-2, where the prefix that corresponds to client2 is injected into the routing domain. PCE 2 then sends a *PCInitiate* R2-3. R2-3 informs PCE 2 on the state of the path with a *PCRpt* message, and once R2-3 has calculated the path to client2, it informs PCE 2 with another *PCRpt*. PCE 2 can now respond to PCE 1 over the TLS channel with the global label sent to the ABR. PCE 1 computes the path towards the ABR that got assigned this global label and sends this as a *PCRes* back to R1-1. R1-1 computes the path to R2-3 through R1-4 and informs PCE 1 with a *PCRpt*.

We verified the correct operations of this path setup across domains by looking at the Label Information Base (LIB) in all relevant routers: R1-1, R1-4, R2-3 and R2-2. All the correct SIDs are added to incoming packets. The path between R1-1 and R1-4 is chosen by the IGP, either through R1-2 or R1-3, and a similar situation happens in Domain 2. We could confirm that the correct forwarding entries were installed in all routers, hence that our software implementation supports creation across domains, while using only PCEPS across domains, the ultimate goal of this work.

7 Discussion

We encountered a few limitations in our implementation that need to be addressed in future work.

At the time of writing there are no devices that officially support PCEPS, hence we started our work with determining which PCE software was extendable to implement this protocol. However, also FRR doesn't support PCEPS. Therefore we had to implement our modified Netphony PCE to be able to set up a PCEPS session with other PCEs as well as regular PCEP with routers.

Furthermore, we only used a pool of 10 global labels per direction, and only between two domains. This poses challenges in the amount of requests from users that can be served within an ISP with this setup and how many domains we can connect to each other. In all these cases, we need to use one unique label to avoid collisions. The scalability remains a challenge due to the limited number of 20-bit MPLS labels available. Ideally, we would like to have the possibility for two PCEs to negotiate a union of free labels and use them on the inter-domain link in an on-demand fashion. The limited amount of labels may be overcome by using SRv6 instead of SR-MPLS; this should be further investigated.

The PCEP protocol still leaves many details about inter-PCE communication open, especially with regard to the PCUpdate and PCReport messages. This means that signaling that one path needs to be deployed is possible, but informing the other party that the path is no longer needed requires more coordination.

The use of stateful PCEs increases the overall control plane overhead, so it is necessary to investigate how the performance of our implementation scales in more complex networks.

The PCE requires reachability information. Ideally, the PCE receives a full Border Gateway Protocol (BGP) feed of each border, but each locally configured route-policy also needs to be replicated towards the PCE to aid the route selection. RFC8821 [22] discusses possible extensions to the PCEP protocol for this. One option proposed is transmitting the next-hop information within the PCEP response.

Furthermore, we chose Free Range Routing (FRR) as the router stack on top of Linux. Yet this has had some limitations: the PCEP implementation of FRR is still experimental and not completely upstreamed at time of testing.

Finally, at the time of writing, a new draft adds support to deploy bidirectional paths. To prevent asymmetric routing it would be beneficial if the return traffic uses the same path. The implementation and design of this are out of scope for this paper.

8 Conclusion

The aim of this research was to investigate possibilities and limitations of PCEPS to set up a multi-domain path to allow for user-driven path control. We explored several aspects of this protocol, specifically focusing on what devices support it, how we can implement it in devices that don't support it natively, and finally analyzing its feasibility in a virtualized multi-domain environment. Our software consists of a modified version of the Netphony PCE that allows for the instantiation of PCEPS sessions, that we use to maintain safe the data about what

multi-domain path is requested by a user. This is an important milestone in our investigation on the consequences that shifting control from operators to end users of a network has on its security.

Acknowledgment. This research received funding from the Dutch Research Council (NWO) under the project UPIN.

References

1. Hesselman, C., et al.: A responsible internet to increase trust in the digital world. *J. Netw. Syst. Manage.* **28**(4), 882–922 (2020)
2. Meijer, A.R., Boldrini, L., Koning, R., Grosso, P.: In: 2022 IEEE/ACM International Workshop on Innovating the Network for Data-Intensive Science (INDIS). IEEE (2022)
3. Bazo, R., Boldrini, L., Hesselman, C., Grosso, P.: In: Proceedings of the ACM SIGCOMM 2021 Workshop on Technologies, Applications, and Uses of a Responsible Internet, pp. 8–13 (2021)
4. Lopez, D., de Dios, O.G., Wu, Q., Dhody, D.: PCEPS: Usage of TLS to Provide a Secure Transport for the Path Computation Element Communication Protocol (PCEP). RFC 8253 (2017). <https://doi.org/10.17487/RFC8253>
5. Sunshine, C.A.: Source routing in computer networks. *ACM SIGCOMM Comput. Commun. Rev.* **7**(1), 29–33 (1977)
6. Filsfils, C., Previdi, S., Ginsberg, L., Decraene, B., Litkowski, S., Shakir, R.: Segment Routing Architecture. RFC 8402 (2018). <https://doi.org/10.17487/RFC8402>
7. Bashandy, A., Filsfils, C., Previdi, S., Decraene, B., Litkowski, S., Shakir, R.: Segment Routing with the MPLS Data Plane. RFC 8660 (2019). <https://doi.org/10.17487/RFC8660>
8. Xu, X., Bryant, S., Farrel, A., Hassan, S., Henderickx, W., Li, Z.: MPLS Segment Routing over IP. RFC 8663 (2019). <https://doi.org/10.17487/RFC8663>
9. Portegies, C., Kaat, M., Grosso, P.: In: 2021 24th Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN), pp. 1–5 IEEE (2021)
10. Farrel, A., Vasseur, J.P., Ash, J.: A path computation element (PCE)-based architecture. Tech. rep. (2006)
11. Vasseur, J., Roux, J.L.L.: Path Computation Element (PCE) Communication Protocol (PCEP). RFC 5440 (2009). <https://doi.org/10.17487/RFC5440>
12. Sivabalan, S., Filsfils, C., Tantsura, J., Henderickx, W., Hardwick, J.: Path Computation Element Communication Protocol (PCEP) Extensions for Segment Routing. RFC 8664 (2019). <https://doi.org/10.17487/RFC8664>
13. Rzym, G., Wajda, K., Cholda, P.: SDN-based WAN optimization: PCE implementation in multi-domain MPLS networks supported by BGP-LS. *Image Process. Commun.* **22**(1), 35–48 (2017)
14. Dugeon, O., Guedrez, R., Lahoud, S., Texier, G.: In: 2017 20th Conference on Innovations in Clouds, Internet and Networks (ICIN), pp. 143–145. IEEE (2017)
15. Dowling, B., Fischlin, M., Günther, F., Stebila, D.: A cryptographic analysis of the TLS 1.3 handshake protocol. *J. Cryptol.* **34**(4), 37 (2021)
16. Boldrini, L., Bazo, R., Hesselman, C., Grosso, P.: In: ICT Open 2021 (2021)
17. Minei, I., Crabbe, E., Sivabalan, S., Ananthakrishnan, H., Dhody, D., Tanaka, Y.: RFC 8697 Path Computation Element Communication Protocol (PCEP) Extensions for Establishing Relationships between Sets of Label Switched Paths (LSPs) (2020)

18. Vasseur, J., Zhang, R., Bitar, N., Le Roux, J.: A backward-recursive PCE-based computation (BRPC) procedure to compute shortest constrained inter-domain traffic engineering label switched paths. Tech. rep. (2009)
19. Dugeon, O., Meuric, J., Lee, Y., Ceccarelli, D.: PCEP Extension for Stateful Inter-Domain Tunnels. Internet-Draft draft-ietf-pce-stateful-interdomain-03, Internet Engineering Task Force (2022). <https://datatracker.ietf.org/doc/draft-ietf-pce-stateful-interdomain/03/>. Work in Progress
20. Zhang, F., Zhao, Q., de Dios, O.G., Casellas, R., King, D.: Path Computation Element Communication Protocol (PCEP) Extensions for the Hierarchical Path Computation Element (H-PCE) Architecture (2019)
21. Boldrini, L.: PCEPS Proof of concept (2023). <https://bitbucket.org/leoboldrini/workspace/projects/PCEPS>
22. Wang, A., Khasanov, B., Zhao, Q., Chen, H.: PCE-Based Traffic Engineering (TE) in Native IP Networks. RFC 8821 (2021). <https://doi.org/10.17487/RFC8821>