Extended abstract: Evaluation of Path Computation Engines in Multi Domain scenarios

Boldrini, L.; Grosso, P.

DOI
10.1145/3538395.3545314

Publication date
2022

Document Version
Final published version

Published in
TAURIN+BGI '22

Extended abstract: Evaluation of Path Computation Engines in Multi Domain scenarios

Leonardo Boldrini
University of Amsterdam
Amsterdam, The Netherlands
l.boldrini@uva.nl

Paola Grosso
University of Amsterdam
Amsterdam, The Netherlands
p.grosso@uva.nl

ABSTRACT

The goal of the UPIN [1] project is providing knowledge and control to end users of a multi-domain network like the Internet to increase security of data in transit. The process of setting up a path across multiple domains according to end-to-end constraints set by the user necessitates a heavy exchange of control messages between the domains involved. In this paper, we investigate how Path Computation Engines (PCEs) that control single domains can help also in the setup of a multi-domain path. This research addresses the lack of transparency, accountability and controllability in today's Internet and narrows the gap towards a Responsible Internet.

1 INTRODUCTION

Modern society relies heavily on communication services and considers them as a strategic asset, as they form the foundations of for example safety-critical services such as smart energy grid and medical registries [2]. Everyone makes intensive use every day of smart devices that depend almost exclusively on the correct functionality of networked communications. Service providers need to deal with a multi-domain network that must carry data where it's supposed to and nowhere else, for safety and privacy reasons. Critical service providers also require new and ever higher levels of security from the Internet, as they increasingly depend on it as as a communication substrate to deliver their services.

The current landscape of networking is slowly moving to automated and software-based infrastructure. This allows the ability to quickly and dynamically steer or redirect traffic onto a different path. However, new and strong requirements on safety and privacy of data in transit, requires to have explicit control on how packets flow through the network; automation in this case also needs to follow the request set by a user, e.g. if a firewall is requested and suddenly becomes unavailable, traffic needs to be steered through a different firewall rather than to the destination directly.

In the current network architecture, each router makes forwarding decisions based on the destination of each packet. Source routing contrasts this traditional destination-based approach, and the full path that a packet needs to traverse to reach its destination, is decided and appended to the packet at the moment it enters the network. Following this paradigm, the packet is encapsulated with all the information that is needed to traverse the network to reach the destination. The network can then forward the packet based on these information.

Segment Routing (SR) is a form of source routing, where each node and link in the network is identified by a Segment IDentifier (SID). When a packet reaches the ingress router, that is, the first router of our network, it receives a set of segments that define the whole path to be taken by the packet itself. The packet then traverses each segment in sequence and gets forwarded over the shortest path to the network element defined by the next segment. This method offers fine-grain control in how traffic flows through the network. For example, packets can be redirected over a different path if a link in the initial path is suffering from congestion. To manage the routing decisions and the paths in an SR data plane, we rely on the concept of Software Defined Networking (SDN), as this allows automated control of the network. We focused our work on the SR-MPLS data plane, the implementation of Segment Routing in IPv4. We used the Path Computation Element Communication Protocol (PCEP) [5] to define paths and instruct ingress nodes to automatically and dynamically provision paths for specific ingress packets.

So far, deployments of these technologies have been limited to single administrative domains or multiple domains under the same administration. In these cases, a central controller has knowledge of the whole topology of the network to be traversed.

In this paper, we focus our research on the deployment of multiple domains, each with its own controller, that has the ability to control only on its domain, and communicates with other domains the segments needed to build the whole path. With this setup, we can ensure a specific set of end-to-end requirements to the path, across multiple different administrative domains.

Our goal is to chain multiple paths together across multiple domains. We achieve this by having multiple central controllers talk to each other using PCEP to provision an end to end path. We want to answer the following research question: "How can we achieve path construction across multiple domains using Path Computation Elements (PCEs)?". We present here how Path Computations Engines work in multi-domain and we provide a description of our setup.

2 PATH COMPUTATION ENGINES

A Path Computation Element (PCE) is an element in a network design that is responsible for computing paths based on a fixed set of constraints [4], according to the algorithm of Constraint-based Shortest Path First (CSPF). A PCE can be stateful or stateless; in a stateful setup, it keeps track of all the paths it has computed and

TAURIN+BGI '22, August 22, 2022, Amsterdam, Netherlands

ACM ISBN 978-1-4503-9394-2/22/08…$15.00
https://doi.org/10.1145/3538395.3545314

will remove this information if the path is no longer needed or is destroyed. In a stateless approach, the PCE will not keep state of the paths it has allocated and just uses the information available in the Traffic Engineering Database (TED) to compute a path.

The TED keeps track of the current links and reserved link utilisation in the network. With the information in the TED, the PCE can perform CSPF to compute paths based on a set of requested constraints.

The path that a PCE computes can be a set of MPLS labels, a specific wavelength in an Optical Transport Network (OTN) or a set of SIDs in an SRv6 network. The PCE will instruct the network layer to construct this path. The path construction can be done by, for example, pushing configuration changes to the network devices to provision the path, or, as in our experiments, using PCEP to program the network devices.

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 Segment IDentifiers (SIDs) and a path will consist of a set of these labels [3]. SR can be used with IPv6 as well and that remains for now a future direction of research.

The PCEP is a protocol originally designed with MPLS and Generalized Multi-Protocol Label Switching (GMPLS) networking in mind to facilitate the communication between PCEs, and specifies the operation between a PCE and a client, called Path Computation Client (PCC) [5]. The PCEP defines a specific set of requests. Each session consists of a Transmission Control Protocol (TCP) connection. This is optionally secured via SSL or TCP-MD5. This connection can be done via an out of band or in-band network. The setup of a Label Switched Path (LSP) using PCEP consists of a PCC sending a path computation request. After this, the PCE response message will contain the path or an error message, in case no path was found. If the PCE wants to update the path, it can send an update message containing the new label stack for this LSP. When the connection between the PCE and the PCC is lost, the PCC will try to minimize traffic disruptions. It will reconnect to the PCE, and only if this fails it will remove the installed paths.

In our setup, SR and PCEs allow users to request services for their traffic to go through. Our work does not focus only on the path creation, but also on the intermediate elements to be traversed, thus addressing the real goals of a Responsible Internet. This concept is applicable for all traffic, in all those situations where additional control is needed.

3 SETUP DESCRIPTION

In a multi-domain setup, PCEs need to talk to each other in order to create a path that crosses their domains. The following two RFCs provided two different approaches to this end.

RFC8685 [7] defines a set of extensions to the PCE Communication Protocol to make this inter PCE communication possible. This design splits the PCE into two parts. A domain-specific controller and a global "parent" controller. If a domain-specific controller does not know about the specific nodes in the path it will delegate this request to the parent controller. This parent controller can be used for inter-domain computation and delegation, or inter-layer computation and delegation. The exact setup described in this RFC depends on the kind of network that is deployed.

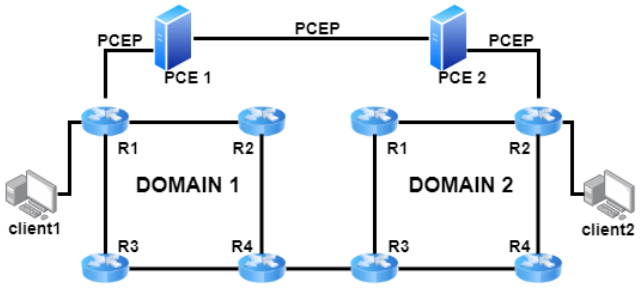

Figure 1: Setup topology. PCEP protocol running in all relevant links.

Another approach is defined in RFC5441 [6]. In this case, an extra flag is defined for the compute request. If a PCE receives a request with this flag, it will perform the algorithm defined in the RFC to compute the inter-domain path. Each PCE will forward the request towards the PCE that has the endpoint in its domain. This PCE will then append all possible paths from the previous domain to the response. Each PCE in the chain will perform the same action. The first PCE in the chain will then select the path based on the list of paths received. In our setup, we make use of this last technique. The chosen path is communicated back to the PCEs in the chain using a report message. This way each PCE in the chain knows which path is chosen and can provision the correct one.

We are currently working on this implementation of PCEP in the setup shown in Figure 1: two domains, each controlled by one PCE and running their own Interior Gateway Protocol (IGP). Two hosts act as clients to generate and receive traffic. Following the UPIN framework, a user is given a frontend where it can choose what constraints its traffic should have. These constraints are then translated into an SR path. The PCEs we use in this setup is a Netphony PCE. This decision comes from a review of different PCEs, namely NorthStar, OpenDayLight, ONOS, IOS XR and Netphony. We evaluated these PCEs based on which of the following features they have or support: open source, hybrid PCE, SR, stateful, extendability, full PCE, OSPF-TE, BGP-LS, ISIS-TE. We decided for Netphony as it checks most of the features we want. This setup is still under development. We are testing different messages to be exchanged in the inter-domain link. BGP-LS is used nowadays by controllers and PCEs also to gather information about the topology of their domain. Implementing BGP-LS between PCEs belonging to different domains brings the challenge of deciding what information each PCE wants to share with the others. Our contribution consists in changing the way a PCE works in order to exchange only a specific set of information to other domains. PCEPS, a version of PCEP that runs on TLS, is also under development. We will compare how different PCEs can achieve our goal of end-to-end, multi-domain path under the constraints set by the user. An interesting implementation consists in steering traffic towards specific Virtual Network Functions (VNFs), according to the request of the user. We already deployed a single domain implementation of this in [3].

This research is part of the UPIN (User-driven Path verification and control for Inter-domain Networks) project, that aims to increase the trust of end users in the Internet by providing them with

the necessary level of knowledge and control over how their traffic flows through the network. This research received funding from the Dutch Research Council (NWO).

REFERENCES

[1] Rodrigo Bazo, Leonardo Boldrini, Cristian Hesselman, and Paola Grosso. 2021. Increasing the Transparency, Accountability and Controllability of multi-domain networks with the UPIN framework. In *Proceedings of the ACM SIGCOMM 2021 Workshop on Technologies, Applications, and Uses of a Responsible Internet*. 8–13.

[2] Fabio Bisogni, Simona Cavallini, Luisa Franchina, and Giovanni Saja. 2012. The European perspective of telecommunications as a critical infrastructure. In *International Conference on Critical Infrastructure Protection*. Springer, 3–15.

[3] Cees Portegies, Leonardo Boldrini, Marijke Kaat, and Paola Grosso. 2021. Experience with implementing VNF chains with Segment Routing and PCEP. In *2021 IFIP/IEEE International Symposium on Integrated Network Management (IM)*. IEEE, 797–811.

[4] JP Vasseur, Adrian Farrel, and Gerald Ash. 2006. A Path Computation Element (PCE)-Based Architecture. RFC 4655. https://doi.org/10.17487/RFC4655

[5] JP Vasseur and Jean-Louis Le Roux. 2009. Path Computation Element (PCE) Communication Protocol (PCEP). RFC 5440. https://doi.org/10.17487/RFC5440

[6] JP Vasseur, Jean-Louis Le Roux, Raymond Zhang, and Dr. Nabil N. Bitar. 2009. A Backward-Recursive PCE-Based Computation (BRPC) Procedure to Compute Shortest Constrained Inter-Domain Traffic Engineering Label Switched Paths. RFC 5441. https://doi.org/10.17487/RFC5441

[7] Fatai Zhang, Quintin Zhao, Oscar Gonzalez de Dios, R. Casellas, and Daniel King. 2019. Path Computation Element Communication Protocol (PCEP) Extensions for the Hierarchical Path Computation Element (H-PCE) Architecture. RFC 8685. https://doi.org/10.17487/RFC8685