Exploring various use cases for IPv6 Segment Routing

Fabien Duchene UCLouvain, Belgium Mathieu Jadin UCLouvain, Belgium first.last@uclouvain.be Olivier Bonaventure UCLouvain, Belgium

ABSTRACT

IPv6 Segment Routing (SRv6) is a modern version of source routing that is being standardised within the IETF to address a variety of use cases in ISP, datacenter and entreprise networks. Its inclusion in recent versions of the Linux kernel enables researchers to explore and extend this new protocol.

We leverage and extend the SRv6 implementation in the Linux kernel to demonstrate two very different usages of this new protocol. We first show how entreprise networks can leverage SRv6 to better control the utilisation of their infrastructure and demonstrate how DNS resolvers can act as SDN controllers. We then demonstrate how SRv6Pipes can be used to efficiently implement network functions that need to process bytestreams on top of a packet-based SRv6 network.

CCS CONCEPTS

• Networks → Network architectures; Routing protocols; Middle boxes / network appliances; Network management; Routers; End nodes; Network manageability; Naming and addressing;

ACM Reference Format:

Fabien Duchene, Mathieu Jadin, and Olivier Bonaventure. 2018. Exploring various use cases for IPv6 Segment Routing. In SIGCOMM Posters and Demos '18: ACM SIGCOMM 2018 Conference Posters and Demos, August 20–25, 2018, Budapest, Hungary. ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3234200.3234213

1 INTRODUCTION

Network operators often need to control how packets flow through their networks either for performance or security reasons. A wide range of techniques have been proposed over the last decades to solve this problem (e.g. ATM, MPLS, Openflow, ...). Segment Routing (SR) is a modern variant of

SIGCOMM Posters and Demos '18, August 20–25, 2018, Budapest, Hungary © 2018 Association for Computing Machinery.

ACM ISBN 978-1-4503-5915-3/18/08...\$15.00

https://doi.org/10.1145/3234200.3234213

the source routing paradigm. Two variants of SR are being developed within the IETF. The MPLS variant is targeted at ISP networks. The IPv6 version of Segment Routing (SRv6) has much broader use cases. It uses a new IPv6 routing extension header, the Segment Routing Header (SRH) [5]. This header contains the IPv6 adresses of intermediate nodes that need to be traversed by the packet before reaching the final destination and several additional fields including a HMAC to solve the security problems that affected the IPv4 variant of source routing. Compared with IPv4 source routing, SRv6 brings several key benefits. First, SRv6 leverages routing protocols such as OSPF or IS-IS to distribute the network topology and the nodes/links addresses that are used in the SRH. Second, thanks to the large size of the IPv6 addresses, it becomes possible to program an SRv6 network as a series of functions triggered by the SRH [1].

We show how the recent implementation of SRv6 in the Linux kernel [2] can be leveraged by researchers to develop new use cases or extend this new protocol. We first demonstrate how entreprise networks can benefit from SRv6 by exposing SRHs to endhosts. We then show how to combine TCP proxies with SRv6 to provide in-network functions that operate on bytestreams.

2 IPV6 SEGMENT ROUTING IN ENTERPRISE NETWORKS

Many entreprises are seduced by Software Defined Networks (SDN) which promise to simplify the management of their networks. Software Resolved Networks (SRN) [3] instantiate this SDN vision by using SRv6 in enterprise networks. Like SDNs, SRNs use a controller that manages the network resources. As in SDNs, the presence of the controller simplifies the management of the network and allows the operator to better control the available resources. However, there are several differences between SRNs and SDNs. First, SRNs leverage SRv6 in the dataplane and the SRH to control the flow of packets through the network. This reduces the amount of state required on the routers in contrast with Openflow-based SDNs. Second, applications can interact explicitely with the controller to indicate the requirements for their flows. The controller responds to these requirements by returning an SRH for a path that meets them.

In SRNs, the controller is co-located with the entreprise DNS resolver and hosts use the DNS protocol to interact with the controller/resolver. When an application initiates

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.



Figure 1: Illustration of path selection in SRN

a conversation, it performs the following operations. First, it issues a DNS request to resolve the DNS server name and adds its requirements. Then, the controller chooses a network path that meets those requirements. The controller can use any optimization algorithm to select this path. Once the path is chosen, it is transformed into a list of SRv6 segments. Different path selection algorithms can be included in the SRN controller. The controller then sends back to the endhost a DNS response containing the server IPv6 address and the SRH corresponding to the selected path. Finally, the endhost attaches the SRH to each packet of the connection.

Figure 1 shows a Software Resolved Network. All its links have an IGP weight of 1. In a traditional IPv6 network, the application flows have to follow the shortest network path. In this example, the application wants an RTT of maximum 8ms and the shortest path has an RTT of 12ms. The controller selects the upper path and returns its SRH to the endhost. Additional details about SRNs and the realisation of its controller may be found in [3]. During this demonstration, we will setup the small topology shown in Figure 1 with 3 home routers running OpenWRT and two laptops. We will conduct the following experiment: a client sends a DNS request, receives an SRH matching its requirements (as shown in Figure 1) and a link failure happens in the network to which the controller reacts to minimize the delay. The network path and its delay will be displayed at each step.

3 MIDDLEBOXES AND IPV6 SEGMENT ROUTING

IPv6 Segment Routing can also be used to support middleboxes. Middleboxes can perform two different types of network functions: per-packet (Network Address Translation, simple firewall) and per-bytestream (Intrusion Detection, Transocoding). Per-packet functions can be easily supported by using SRv6 [1] since they operate on the network or transport layer header. Per-bystream functions are more complex and often need to reconstruct the TCP bytestream and cope with reordering, losses, Such functions need to reorder the received TCP packets which is similar to including a complete TCP implementation in each function. Instead of forcing each function to include a TCP stack, our recently proposed SRv6Pipes [4] enable in-network bytestream functions with two main components: a transparent TCP proxy and a scheme to encode functions and parameters inside the SRH. Each middlebox uses a transparent TCP proxy to terminate the TCP connections before passing its bytestream to the network function. A middlebox can support different functions (e.g. transparent compression/decompression, IDS, \ldots). Each of the supported functions is implemented as a dynamically loadable function that processes the bytestream exposed by the TCP proxy. The network uses the SRH to enforce the utilisation of specific network functions. This SRH can be inserted by the client (that receives it from the controller described in section 2), or by an access router. Since a given middlebox may support different functions, we need a way to specify the function that needs to process each packet on each middlebox. For this, we encode the functions and the parameters in the IPv6 address, as described in [1, 4]. The SRH is thus used for two different purposes: (*i*) enforcing a specific path through selected middleboxes and (*ii*) indicating the network functions to be applied to each packet/flow.



Figure 2: IPv6 network with 2 middleboxes.

Consider the network described in Figure 2 where the client requires to encrypt the traffic between P1 and P2. The client will use the function bits of the address of P1 to specify the identifier of the encrypt function (AAAA), and the parameters bits to specify the identifier (BCDE:FFFF) of an encryption key. The same will be done in P2's address with the decrypt function. While receiving the packets, the proxies will look at the function and parameters bits to pass them to the selected function. In [4] we demonstrated that our implementation¹ incurs almost no performance degradation at 10Gbps. During this demonstration, we will setup a small testbed with home routers running OpenWRT and 2 laptops and conduct the following experiment: a client sends traffic that will be processed and modified by the different proxies. The modified payload will be displayed at each step.

ACKNOWLEDGEMENTS

This work was partially supported by a Cisco grant and by the ARC grant 13/18-054 (ARC-SDN) funded by Communaute francaise de Belgique. M. Jadin is supported by a grant from F.R.S.-FNRS FRIA.

¹Available at http://segment-routing.org/index.php/SRv6Pipes

Exploring various use cases for IPv6 SR

REFERENCES

- Clarence Filsfils et al.. 2018. SRv6 Network Programming. (March 2018). Internet draft, draft-filsfils-spring-srv6-network-programming-04. Internet Engineering Task Force. Work in progress.
- [2] David Lebrun et al.. 2017. Implementing IPv6 Segment Routing in the Linux Kernel. In ANRW. ACM.
- [3] David Lebrun et al.. 2018. Software resolved networks: Rethinking enterprise networks with ipv6 segment routing. In *Proceedings of the Symposium on SDN Research*. ACM, 6.
- [4] Fabien Duchêne et al.. 2018. SRv6Pipes: enabling in-network bytestream functions. In *IFIP Networking 2018.*
- [5] Stefano Previdi et al.. 2017. IPv6 Segment Routing Header (SRH). Internet-Draft draft-ietf-6man-segment-routing-header-07. Internet Engineering Task Force. Work in Progress.