Demo: IPv6 Segment Routing to the End Host: A Linux Kernel Implementation

David Lebrun
ICTEAM, Université catholique de Louvain
Louvain-La-Neuve – Belgium
firstname.lastname@uclouvain.be

ABSTRACT

Segment Routing (SR) is already deployed in its MPLS variant. We focus on its IPv6 flavor (SRv6) and argue that it enables the hosts to directly participate in the management of their flows, through an SDN-like controller. To realise this, we implement SRv6 in the Linux kernel. Our implementation has been merged in the mainstream Linux tree and is available as of Linux 4.10. We argue that such a public, open-source implementation enables other researchers to explore SRv6. We propose a demonstration of our implementation.

1. SEGMENT ROUTING

Segment Routing is a new source routing architecture that is being developed within the Internet Engineering Task Force [6, 5]. It allows packets to follow non-shortest paths towards the destination by specifying a list of detours, i.e. waypoints. These waypoints are called *segments*. The packets are forwarded along the shortest path from the source to the first segment, then through the shortest path from the first segment to the second segment, and so on. Segment Routing currently exists in two flavors: MPLS and IPv6 [7, 10]. SR-MPLS is already implemented by network vendors and deployed by operators.

Multiple work on Segment Routing are already available in the literature, including architectural work [5], optimisation papers [9, 8, 1, 3], control-plane work [11] and monitoring research [2]. These work are usually agnostic to the SR flavor. We argue that the inherent features of the IPv6 flavor of Segment Routing (SRv6) open the doors to a large, yet mostly unexplored, research area involving the active participation of endhosts in their network traffic.

SOSR '17, April 03-04, 2017, Santa Clara, CA, USA

DOI: http://dx.doi.org/10.1145/3050220.3060605

SRv6 is realised through an IPv6 extension header, the routing header (RH). The RH is already defined in the IPv6 protocol [4]. The Segment Routing Header (SRH) is defined as an extension of the routing header [10]. The SRH contains a list of segments, encoded as IPv6 addresses (usually globally routable and announced by an underlying IGP such as OSPF or IS-IS). A segment can represent a topological instruction (node or link traversal) or any operator-defined instruction (*e.g.*, virtual function). An SRH can be used to steer packets through paths with given properties (*e.g.*, bandwidth or latency) and through various network functions (*e.g.*, firewalling). The list of segments present in the SRH thus specifies the network policy relevant to the packet. Additionally, an SRH may contain one or more TLV fields. Such TLVs include an optional HMAC TLV, used for authenticity and integrity checks.

When a router must impose an SRH on a forwarded packet, the packet is encapsulated in an outer IPv6 header containing the SRH. The original packet is left unmodified as the payload. The router is called the *SR ingress node*. The destination address of the outer IPv6 header is set to the first segment of the list and the packet is forwarded to the corresponding node following the shortest path. This node (called a *segment endpoint*) then processes the packet by updating the destination address to the next segment. The last segment of the list is the *SR egress node*. It decapsulates the inner packet and forwards it to its original destination.

A particular feature of SRv6 is that it can be used in the network up to the endhosts, as they already support (or are meant to support) IPv6, as opposed to MPLS which is only used in core networks. Such feature enables the endhosts to actively participate in the management and policing of their network traffic, with the help of an SDN-like controller. The first step in realising such an architecture would be to enable the support of SRv6 in the endhosts. Thus, we implemented SRv6 in the Linux kernel.

2. IMPLEMENTATION

We implemented the support for SRv6 in the Linux kernel, as defined in [10]. This implementation has been merged in the mainstream Linux tree [1] and is available as of Linux 4.10.

[1] https://lkml.org/lkml/2016/12/11/99

Our implementation consists of two classes of features: data plane support and control plane support. The data plane support is basically the implementation of [10], *i.e.*, the processing of the SRH. We also support the verification of the HMAC TLV optionally present in the SRH, with the SHA-1 and SHA-256 hashing algorithms.

The control plane part enables to map an SRH to an IPv6 route. Packets matching such a route are encapsulated in an outer IPv6 header with the corresponding SRH. Although only IPv6 packets can currently be encapsulated with an SRH, the implementation is generic enough to easily support other types of payload to be carried with SRv6 (*e.g.*, IPv4, MPLS, Ethernet, etc.). The encapsulation mechanism is called *lightweight tunnels* and enables to map a route to an alternative processing function. In our implementation, we map this function to the SRv6 encapsulation function. A userspace API is exposed through the `NETLINK` protocol, which is already used for most of the Linux networking configuration through the userspace tool `iproute2`. We modified this tool to support our SRv6 implementation.

Finally, we implemented a per-socket SRH insertion mechanism. This feature enables the applications to explicitly define an SRH for a particular socket through the `setsockopt()` system call. This SRH can be defined locally, or the application can receive it from an SDN-like controller. Such a mechanism enables to perform per-flow traffic engineering directly at the source of the traffic. It is the first step in more complex, host-centric features that would enable the endhosts to participate in the management of their traffic. For example, we could modify TCP to automatically define an SRH in a SYN/ACK packet, when an SRH is present in the related SYN packet.

The availability of SRv6 in the mainstream Linux kernel is a key achievement in enabling other researchers to explore these research opportunities with Segment Routing.

3. DEMONSTRATION

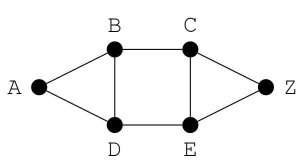

Figure 1: Demonstration network.

We propose to demonstrate the SRv6 Linux implementation through a dynamic network visualisation as shown in Figure 1. A traffic generator (`iperf3`) runs at node `A` and emits traffic towards `Z`, saturating the traversed links. The visualisation shows the link utilization both in text (percentage) and by changing the link's color, allowing to easily determine the flow's path. Initially, the network is setup to follow the shortest paths. Then, we steer the flow through various paths by inserting SR routes with the `iproute2` command. The visualisation shows the change in link utilization in real time, allowing to follow the path changes. We take care to distinguish both directions of a given link, to enable the visualisation of asymmetric traffic. This video [2] gives an idea of what the demonstration would look like. In this video, we first show the state of the network in shortest-path forwarding, with `iperf3` running. Then, we insert a route to steer the traffic through nodes `D,B,C,E` and visualize the resulting path. We then return in the shortest-path mode by removing the route and we insert another route to steer the traffic through node `C`.

Acknowledgements

This work was partially supported by a Cisco grant and by the ARC grant 13/18-054 (ARC-SDN) from Communauté française de Belgique.

4. REFERENCES

[1] F. Aubry, D. Lebrun, Y. Deville, and O. Bonaventure. Traffic duplication through segmentable disjoint paths. In *IFIP Networking Conference (IFIP Networking), 2015*, pages 1–9. IEEE, 2015.

[2] F. Aubry, D. Lebrun, et al. Scmon: Leveraging segment routing to improve network monitoring. In *35th Annual IEEE International Conference on Computer Communications, INFOCOM 2016, San Francisco, CA, USA, April 10-14, 2016*, pages 1–9, 2016.

[3] R. Bhatia, F. Hao, M. Kodialam, and T. Lakshman. Optimized network traffic engineering using segment routing. In *Computer Communications (INFOCOM), 2015 IEEE Conference on*, pages 657–665. IEEE, 2015.

[4] D. S. E. Deering. Internet Protocol, Version 6 (IPv6) Specification. RFC 2460, Dec. 1998.

[5] C. Filsfils et al. The segment routing architecture. In *2015 IEEE Global Communications Conference (GLOBECOM)*, pages 1–6. IEEE, 2015.

[6] C. Filsfils et al. Segment Routing Architecture. Internet-Draft draft-ietf-spring-segment-routing-11, Internet Engineering Task Force, Feb. 2017. Work in Progress.

[7] C. Filsfils et al. Segment Routing with MPLS data plane. Internet-Draft draft-ietf-spring-segment-routing-mpls-07, Internet Engineering Task Force, Feb. 2017. Work in Progress.

[8] F. Hao, M. Kodialam, and T. Lakshman. Optimizing restoration with segment routing. In *Computer Communications, IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on*, pages 1–9. IEEE, 2016.

[9] R. Hartert, S. Vissicchio, et al. A declarative and expressive approach to control forwarding paths in carrier-grade networks. In *ACM SIGCOMM Computer Communication Review*, volume 45, pages 15–28. ACM, 2015.

[10] S. Previdi et al. IPv6 Segment Routing Header (SRH). Internet-Draft draft-ietf-6man-segment-routing-header-04, Internet Engineering Task Force, Jan. 2017. Work in Progress.

[11] S. Salsano et al. Pmsr: Poor man's segment routing, a minimalistic approach to segment routing and a traffic engineering use case. In *Network Operations and Management Symposium (NOMS), 2016 IEEE/IFIP*, pages 598–604. IEEE, 2016.

[2] http://segment-routing.org/sosr-demo.mp4