Spring
Internet-Draft
Intended status: Informational
Expires: April 30, 2015

J. Brzozowski
J. Leddy
Comcast
I. Leung
Rogers Communications
S. Previdi
M. Townsley
C. Martin
C. Filsfils
R. Maglione, Ed.
Cisco Systems

October 27, 2014

IPv6 SPRING Use Cases
draft-ietf-spring-ipv6-use-cases-02

Abstract

Source Packet Routing in Networking (SPRING) architecture leverages the source routing paradigm. A node steers a packet through a controlled set of instructions, called segments, by prepending the packet with SPRING header. A segment can represent any instruction, topological or service-based. A segment can have a local semantic to the SPRING node or global within the SPRING domain. SPRING allows to enforce a flow through any topological path and service chain while maintaining per-flow state only at the ingress node to the SPRING domain.

The objective of this document is to illustrate some use cases that need to be taken into account by the Source Packet Routing in Networking (SPRING) architecture.

This Internet-Draft will expire on April 30, 2015.

Table of Contents

1. Introduction

Source Packet Routing in Networking (SPRING) architecture leverages the source routing paradigm. An ingress node steers a packet through a controlled set of instructions, called segments, by prepending the packet with SPRING header. A segment can represent any instruction, topological or service-based. A segment can represent a local semantic on the SPRING node, or a global semantic within the SPRING domain. SPRING allows one to enforce a flow through any topological path and service chain while maintaining per-flow state only at the ingress node to the SPRING domain.

The SPRING architecture is described in [I-D.filsfils-spring-segment-routing]. The SPRING control plane is agnostic to the dataplane, thus it can be applied to both MPLS and IPv6. In case of MPLS the (list of) segment identifiers are carried

in the MPLS label stack, while for the IPv6 dataplane, a new type of routing extension header is required.

The details of the new routing extension header are described in [I-D.previdi-6man-segment-routing-header] which also covers the security considerations and the aspects related to the deprecation of the IPv6 Type 0 Routing Header described in [RFC5095].

2. IPv6 SPRING use cases

In today's networks, source routing is typically accomplished by encapsulating IP packets in MPLS LSPs that are signaled via RSVP-TE. Therefore, there are scenarios where it may be possible to run IPv6 on top of MPLS, and as such, the MPLS Segment Routing architecture described in [I-D.filsfils-spring-segment-routing-mpls] could be leveraged to provide SPRING capabilities in an IPv6/MPLS environment.

However, there are other cases and/or specific network segments (such as for example the Home Network, the Data Center, etc.) where MPLS may not be available or deployable for lack of support on network elements or for an operator's design choice. In such scenarios a non-MPLS based solution would be preferred by the network operators of such infrastructures.

In addition there are cases where the operators could have made the design choice to disable IPv4, for ease of management and scale (return to single-stack) or due to an address constraint, for example because they do not possess enough IPv4 addresses resources to number all the endpoints and other network elements on which they desire to run MPLS.

In such scenario the support for MPLS operations on an IPv6-only network would be required. However today's IPv6-only networks are not fully capable of supporting MPLS. There is ongoing work in the MPLS Working Group, described in [I-D.ietf-mpls-ipv6-only-gap] to identify gaps that must be addressed in order to allow MPLS-related protocols and applications to be used with IPv6-only networks. This is an another example of scenario where an IPv6-only solution could represent a valid option to solve the problem and meet operators' requirements.

It is important to clarify that today, it is possible to run IPv6 on top of an IPv4 MPLS network by using the mechanism called 6PE, described in [RFC4798]. However this approach does not fulfill the requirement of removing the need of IPv4 addresses in the network, as requested in the above use case.

In addition it is worth to note that in today's MPLS dual-stack networks IPv4 traffic is labeled while IPv6 traffic is usually natively routed, not label-switched. Therefore in order to be able to provide Traffic Engineering "like" capabilities for IPv6 traffic additional/alternative encapsulation mechanisms would be required.

In summary there is a class of use cases that motivate an IPv6 data plane. The authors identify some fundamental scenarios that, when recognized in conjunction, strongly indicate an IPv6 data plane:

1. There is a need or desire to impose source-routing semantics within an application or at the edge of a network (for example, a CPE or home gateway)

2. There is a strict lack of an MPLS dataplane

3. There is a need or desire to remove routing state from any node other than the source, such that the source is the only node that knows and will know the path a packet will take, a priori

4. There is a need to connect millions of addressable segment endpoints, thus high routing scalability is a requirement. IPv6 addresses are inherently summarizable: a very large operator could scale by summarizing IPv6 subnets at various internal boundaries. This is very simple and is a basic property of IP routing. MPLS node segments are not summarizable. To reach the same scale, an operator would need to introduce additional complexity, such as mechanisms described in [I-D.ietf-mpls-seamless-mpls]

In any environment with requirements such as those listed above, an IPv6 data plane provides a powerful combination of capabilities for a network operator to realize benefits in explicit routing, protection and restoration, high routing scalability, traffic engineering, service chaining, service differentiation and application flexibility via programmability.

This section will describe some scenarios where MPLS may not be present and it will highlight how the SPRING architecture could be used to address such use cases, particularly, when an MPLS data plane is neither present nor desired.

The use cases described in the section do not constitute an exhaustive list of all the possible scenarios; this section only includes some of the most common envisioned deployment models for IPv6 Segment Routing.

In addition to the use cases described in this document the SPRING architecture can be applied to all the use cases described in [I-D.ietf-spring-problem-statement] for the SPRING MPLS data plane, when an IPv6 data plane is present.

2.1. SPRING in the Home Network

An IPv6-enabled home network provides ample globally routed IP addresses for all devices in the home. An IPv6 home network with multiple egress points and associated provider-assigned prefixes will, in turn, provide multiple IPv6 addresses to hosts. A homenet performing Source and Destination Routing ([I-D.troan-homenet-sadr]) will ensure that packets exit the home at the appropriate egress based on the associated delegated prefix for that link.

A SPRING enabled home provides the possibility for imposition of a Segment List by end-hosts in the home, or a customer edge router in the home. If the Segment List is enabled at the customer edge router, that router is responsible for classifying traffic and inserting the appropriate Segment List. If hosts in the home have explicit source selection rules (see [I-D.lepape-6man-prefix-metadata]), classification can be based on source address or associated network egress point, avoiding the need for DPI-based implicit classification techniques. If the Segment List is inserted by the host itself, it is important to know which networks can interpret the SPRING header. This information can be provided as part of host configuration as a property of the configured IP address (see [I-D.bhandari-dhc-class-based-prefix]).

The ability to steer traffic to an appropriate egress or utilize a specific type of media (e.g., low-power, WIFI, wired, femto-cell, bluetooth, MOCA, HomePlug, etc.) within the home itself are obvious cases which may be of interest to an application running within a home network.

Steering to a specific egress point may be useful for a number of reasons, including:

- Regulatory
- Performance of a particular service associated with a particular link
- Cost imposed due to data-caps or per-byte charges
- Home vs. work traffic in homes with one or more teleworkers, etc.
- Specific services provided by one ISP vs. another

Information included in the Segment List, whether imposed by the end-host itself, a customer edge router, or within the access network of the ISP, may be of use at the far ends of the data communication as well. For example, an application running on an end-host with application-support in a data center can utilize the Segment List as a channel to include information that affects its treatment within the data center itself, allowing for application-level steering and load-balancing without relying upon implicit application classification techniques at the data-center edge. Further, as more and more application traffic is encrypted, the ability to extract (and include in the Segment List) just enough information to enable the network and data center to load-balance and steer traffic appropriately becomes more and more important.

2.2. SPRING in the Access Network

Access networks deliver a variety of types of traffic from the service provider's network to the home environment and from the home towards the service provider's network.

For bandwidth management or related purposes, the service provider may want to associate certain types of traffic to specific physical or logical downstream capacity pipes.

This mapping is not the same thing as classification and scheduling. In the Cable access network, each of these pipes are represented at the DOCSIS layer as different service flows, which are better identified as differing data links. As such, creating this separation allows an operator to differentiate between different types of content and perform a variety of differing functions on these pipes, such as egress vectoring, byte capping, regulatory compliance functions, and billing.

In a cable operator's environment, these downstream pipes could be a specific QAM, a DOCSIS service flow or a service group.

Similarly, the operator may want to map traffic from the home sent towards the service provider's network to specific upstream capacity pipes. Information carried in a packet's SPRING header could provide the target pipe for this specific packet. The access device would not need to know specific details about the packet to perform this mapping; instead the access device would only need to know how to map the SR SID value to the target pipe.

2.3. SPRING in the Data Center

A key use case for SPRING is to cause a packet to follow a specific path through the network. One can think of the service function performed at each SPRING node to be forwarding. More complex service functions could be applied to the packet by a SPRING node including accounting, IDS, load balancing, and fire walling.

The term "Service Function Chain", as defined in [I-D.ietf-sfc-problem-statement], it is used to describe an ordered set of service functions that must be applied to packets.

A service provider may choose to have these service functions performed external to the routing infrastructure, specifically on either dedicated physical servers or within VMs running on a virtualization platform.

[I-D.ietf-sfc-dc-use-cases] describes use cases that demonstrate the applicability of Service Function Chaining (SFC) within a data center environment and provides SFC requirements for data center centric use cases.

2.4. SPRING in the Content Delivery Networks

The rise of online video applications and new, video-capable IP devices has led to an explosion of video traffic traversing network operator infrastructures. In the drive to reduce the capital and operational impact of the massive influx of online video traffic, as well as to extend traditional TV services to new devices and screens, network operators are increasingly turning to Content Delivery Networks (CDNs).

Several studies showed the benefits of connecting caches in a hierarchical structure following the hierarchical nature of the Internet. In a cache hierarchy one cache establishes peering relationships with its neighbor caches. There are two types of relationship: parent and sibling. A parent cache is essentially one level up in a cache hierarchy. A sibling cache is on the same level. Multiple levels of hierarchy are commonly used in order to build efficient caches architecture.

In an environment, where each single cache system can be uniquely identified by its own IPv6 address, a Segment List containing a sequence of the caches in a hierarchy can be built. At each node (cache) present in the Segment List a TCP session to port 80 is established and if the requested content is found at the cache (cache hits scenario) the sequence ends, even if there are more nodes in the list.

To achieve the behavior described above, in addition to the Segment List, which specifies the path to be followed to explore the hierarchic architecture, a way to instruct the node to take a specific action is required. The function to be performed by a service node can be carried into a new header called Network Service Header (NSH) defined in [I-D.quinn-sfc-nsh]. A Network Service Header (NSH) is metadata added to a packet that is used to create a service plane. The service header is added by a service classification function that determines which packets require servicing, and correspondingly which service path to follow to apply the appropriate service.

In the above example the service to be performed by the service node was to establish a TCP session to port 80, but in other scenarios different functions may be required. Another example of action to be taken by the service node is the capability to perform transformations on payload data, like real-time video transcode option (for rate and/or resolution).

The use of SPRING together with the NSH allows building flexible service chains where the topological information related to the path to be followed is carried into the Segment List while the "service plane related information" (function/action to be performed) is encoded in the metadata, carried into the NSH. The details about using SPRING together with NSH will be described in a separate document.

2.5. SPRING in the Core networks

MPLS is a well-known technology widely deployed in many IP core networks. However there are some operators that do not run MPLS everywhere in their core network today, thus moving forward they would prefer to have an IPv6 native infrastructure for the core network.

While the overall amount of traffic offered to the network continues to grow and considering that multiple types of traffic with different characteristics and requirements are quickly converging over single network architecture, the network operators are starting to face new challenges.

Some operators are looking at the possibility to setup an explicit path based on the IPv6 source address for specific types of traffic in order to efficiently use their network infrastructure. In case of IPv6 some operators are currently assigning or plan to assign IPv6 prefix(es) to their IPv6 customers based on regions/geography, thus the subscriber's IPv6 prefix could be used to identify the region where the customer is located. In such environment the IPv6 source

address could be used by the Edge nodes of the network to steer traffic and forward it through a specific path other than the optimal path.

The need to setup a source-based path, going through some specific middle/intermediate points in the network may be related to different requirements:

- The operator may want to be able to use some high bandwidth links for specific type of traffic (like video) avoiding the need for over-dimensioning all the links of the network;

- The operator may want to be able to setup a specific path for delay sensitive applications;

- The operator may have the need to be able to select one (or multiple) specific exit point(s) at peering points when different peering points are available;

- The operator may have the need to be able to setup a source based path for specific services in order to be able to reach some servers hosted in some facilities not always reachable through the optimal path;

- The operator may have the need to be able to provision guaranteed disjoint paths (so-called dual-plane network) for diversity purposes

All these scenarios would require a form of traffic engineering capabilities in IP core networks not running MPLS and not willing to run it.

IPv4 protocol does not provide such functionalities today and it is not the intent of this document to address the IPv4 scenario, both because this may create a lot of backward compatibility issues with currently deployed networks and for the security issues that may raise.

The described use cases could be addressed with the SPRING architecture by having the Edge nodes of network to impose a Segment List on specific traffic flows, based on certain classification criteria that would include source IPv6 address.

3. Acknowledgements

The authors would like to thank Brian Field, Robert Raszuk, Wes George, John G. Scudder and Yakov Rekhter for their valuable comments and inputs to this document.

4. IANA Considerations

This document does not require any action from IANA.

5. Security Considerations

There are a number of security concerns with source routing at the IP layer [RFC5095]. The new IPv6-based routing header will be defined in way that blind attacks are never possible, i.e., attackers will be unable to send source routed packets that get successfully processed, without being part of the negations for setting up the source routes or being able to eavesdrop legitimate source routed packets. In some networks this base level security may be complemented with other mechanisms, such as packet filtering, cryptographic security, etc.

6. Informative References

[I-D.bhandari-dhc-class-based-prefix]
Systems, C., Halwasia, G., Gundavelli, S., Deng, H., Thiebaut, L., Korhonen, J., and I. Farrer, "DHCPv6 class based prefix", draft-bhandari-dhc-class-based-prefix-05 (work in progress), July 2013.

[I-D.filsfils-spring-segment-routing]
Filsfils, C., Previdi, S., Bashandy, A., Decraene, B., Litkowski, S., Horneffer, M., Milojevic, I., Shakir, R., Ytti, S., Henderickx, W., Tantsura, J., and E. Crabbe, "Segment Routing Architecture", draft-filsfils-spring-segment-routing-04 (work in progress), July 2014.

[I-D.filsfils-spring-segment-routing-mpls]
Filsfils, C., Previdi, S., Bashandy, A., Decraene, B., Litkowski, S., Horneffer, M., Milojevic, I., Shakir, R., Ytti, S., Henderickx, W., Tantsura, J., and E. Crabbe, "Segment Routing with MPLS data plane", draft-filsfils-spring-segment-routing-mpls-03 (work in progress), August 2014.

[I-D.ietf-mpls-ipv6-only-gap]
George, W. and C. Pignataro, "Gap Analysis for Operating IPv6-only MPLS Networks", draft-ietf-mpls-ipv6-only-gap-02 (work in progress), August 2014.

[I-D.ietf-mpls-seamless-mpls]
Leymann, N., Decraene, B., Filsfils, C., Konstantynowicz, M., and D. Steinberg, "Seamless MPLS Architecture", draft-ietf-mpls-seamless-mpls-07 (work in progress), June 2014.

[I-D.ietf-sfc-dc-use-cases]
Surendra, S., Tufail, M., Majee, S., Captari, C., and S. Homma, "Service Function Chaining Use Cases In Data Centers", draft-ietf-sfc-dc-use-cases-01 (work in progress), July 2014.

[I-D.ietf-sfc-problem-statement]
Quinn, P. and T. Nadeau, "Service Function Chaining Problem Statement", draft-ietf-sfc-problem-statement-10 (work in progress), August 2014.

[I-D.ietf-spring-problem-statement]
Previdi, S., Filsfils, C., Decraene, B., Litkowski, S., Horneffer, M., and R. Shakir, "SPRING Problem Statement and Requirements", draft-ietf-spring-problem-statement-03 (work in progress), October 2014.

[I-D.lepape-6man-prefix-metadata]
Pape, M., Systems, C., and I. Farrer, "IPv6 Prefix Meta-data and Usage", draft-lepape-6man-prefix-metadata-00 (work in progress), July 2013.

[I-D.previdi-6man-segment-routing-header]
Previdi, S., Filsfils, C., Field, B., and I. Leung, "IPv6 Segment Routing Header (SRH)", draft-previdi-6man-segment-routing-header-03 (work in progress), October 2014.

[I-D.quinn-sfc-nsh]
Quinn, P., Guichard, J., Fernando, R., Surendra, S., Smith, M., Yadav, N., Agarwal, P., Manur, R., Chauhan, A., Elzur, U., Garg, P., McConnell, B., and C. Wright, "Network Service Header", draft-quinn-sfc-nsh-03 (work in progress), July 2014.

[I-D.troan-homenet-sadr]
Troan, O. and L. Colitti, "IPv6 Multihoming with Source Address Dependent Routing (SADR)", draft-troan-homenet-sadr-01 (work in progress), September 2013.

[RFC4798] De Clercq, J., Ooms, D., Prevost, S., and F. Le Faucheur, "Connecting IPv6 Islands over IPv4 MPLS Using IPv6 Provider Edge Routers (6PE)", RFC 4798, February 2007.

[RFC5095] Abley, J., Savola, P., and G. Neville-Neil, "Deprecation of Type 0 Routing Headers in IPv6", RFC 5095, December 2007.

Authors' Addresses

John Brzozowski
Comcast

Email: john_brzozowski@cable.comcast.com

John Leddy
Comcast

Email: John_Leddy@cable.comcast.com

Ida Leung
Rogers Communications
8200 Dixie Road
Brampton, ON L6T 0C1
CANADA

Email: Ida.Leung@rci.rogers.com

Stefano Previdi
Cisco Systems
Via Del Serafico, 200
Rome 00142
Italy

Email: sprevidi@cisco.com

Mark Townsley
Cisco Systems

Email: townsley@cisco.com

Christian Martin
Cisco Systems

Email: martincj@cisco.com

Clarence Filsfils
Cisco Systems
Brussels
BE

Email: cfilsfil@cisco.com

Roberta Maglione (editor)
Cisco Systems
Via Torri Bianche 8
Vimercate 20871
Italy

Email: robmgl@cisco.com