Spring   J. Brzozowski
Internet-Draft   J. Leddy
Intended status: Informational   Comcast
Expires: September 4, 2016   I. Leung
                                               Rogers Communications
                                                           S. Previdi
                                                          M. Townsley
                                                            C. Martin
                                                          C. Filsfils
                                                    R. Maglione, Ed.
                                                        Cisco Systems
                                                        March 3, 2016

                        IPv6 SPRING Use Cases
                    draft-ietf-spring-ipv6-use-cases-06

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.

Status of This Memo

   This Internet-Draft is submitted in full conformance with the
   provisions of BCP 78 and BCP 79.

   Internet-Drafts are working documents of the Internet Engineering
   Task Force (IETF). Note that other groups may also distribute
   working documents as Internet-Drafts. The list of current Internet-
   Drafts is at http://datatracker.ietf.org/drafts/current/.

   Internet-Drafts are draft documents valid for a maximum of six months
   and may be updated, replaced, or obsoleted by other documents at any
   time. It is inappropriate to use Internet-Drafts as reference
   material or to cite them other than as "work in progress."

   This Internet-Draft will expire on September 4, 2016.

Copyright Notice

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.ietf-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.ietf-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 [RFC7439] 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 known with the industry term
    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. Here there is a summary of those use cases:

1. Traffic Engineering

2. Disjoint paths in dual-plane networks

3. Fast Reroute: Protecting node and adjacency segments

4. OAM/monitoring

5. Egress Peering Engineering

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.ietf-rtgwg-dst-src-routing]) 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, 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.ietf-mif-mpvd-dhcp-support]).

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:

o Regulatory

o Performance of a particular service associated with a particular
   link

o Cost imposed due to data-caps or per-byte charges

o Home vs. work traffic in homes with one or more teleworkers, etc.

o 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 [RFC7498], 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.

2.3.1. VM isolation in a Data Center

One of the fundamental requirements for Data Center architecture is
to provide scalable, isolated tenant networks. Today with OpenStack
Networking (Neutron) this can be achieved via L2 segmentation using
either a) standard 802.1Q VLANs or b) an overlay approach based on
one of several L2 over L3 encapsulation techniques available today
such as 802.1ad, VXLAN, NVGRE. However, these approaches still
struggle to provide scalable, transparent, manageable, high
performance, isolated tenant networks.

The 128-bit PE Ingress ID in the Segment Router Header (SRH) policy
list defined in [I-D.previdi-6man-segment-routing-header] provides a
natural place to encode origin information of VM to VM traffic within
the Data Center. The Segment List provides a method to direct
traffic to a specific enforcement point based on traffic destination.
Together, these allow for a simple tagging and permit/deny comparison

performed between twin SR-capable nodes (e.g., the Neutron Virtual
Router) among VMs in a Data Center.

[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.ietf-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:

o 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;

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

o 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;

o 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;

o 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, Eric Vyncke, Fred Baker, 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]. Security mechanisms applied to Segment Routing over
IPv6 networks are detailed in section 9 of
[I-D.previdi-6man-segment-routing-header]

6. Informative References

[I-D.ietf-mif-mpvd-dhcp-support]
          Krishnan, S., Korhonen, J., and S. Bhandari, "Support for
          multiple provisioning domains in DHCPv6", draft-ietf-mif-
          mpvd-dhcp-support-02 (work in progress), October 2015.

[I-D.ietf-rtgwg-dst-src-routing]
          Lamparter, D., "Destination/Source Routing", draft-ietf-
          rtgwg-dst-src-routing-00 (work in progress), October 2015.

[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-04 (work in
          progress), January 2016.

[I-D.ietf-sfc-nsh]
          Quinn, P. and U. Elzur, "Network Service Header", draft-
          ietf-sfc-nsh-02 (work in progress), January 2016.

[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-07
          (work in progress), March 2016.

[I-D.ietf-spring-segment-routing]
          Filsfils, C., Previdi, S., Decraene, B., Litkowski, S.,
          and R. Shakir, "Segment Routing Architecture", draft-ietf-
          spring-segment-routing-07 (work in progress), December
          2015.

[I-D.ietf-spring-segment-routing-mpls]
          Filsfils, C., Previdi, S., Bashandy, A., Decraene, B.,
          Litkowski, S., Horneffer, M., Shakir, R., Tantsura, J.,
          and E. Crabbe, "Segment Routing with MPLS data plane",
          draft-ietf-spring-segment-routing-mpls-03 (work in
          progress), February 2016.

[I-D.previdi-6man-segment-routing-header]
          Previdi, S., Filsfils, C., Field, B., Leung, I., Linkova,
          J., Kosugi, T., Vyncke, E., and D. Lebrun, "IPv6 Segment
          Routing Header (SRH)", draft-previdi-6man-segment-routing-
          header-08 (work in progress), October 2015.

[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,
          DOI 10.17487/RFC4798, February 2007,
          <http://www.rfc-editor.org/info/rfc4798>.

   [RFC5095]   Abley, J., Savola, P., and G. Neville-Neil, "Deprecation
               of Type 0 Routing Headers in IPv6", RFC 5095,
               DOI 10.17487/RFC5095, December 2007,
               <http://www.rfc-editor.org/info/rfc5095>.

   [RFC7439]   George, W., Ed. and C. Pignataro, Ed., "Gap Analysis for
               Operating IPv6-Only MPLS Networks", RFC 7439,
               DOI 10.17487/RFC7439, January 2015,
               <http://www.rfc-editor.org/info/rfc7439>.

   [RFC7498]   Quinn, P., Ed. and T. Nadeau, Ed., "Problem Statement for
               Service Function Chaining", RFC 7498,
               DOI 10.17487/RFC7498, April 2015,
               <http://www.rfc-editor.org/info/rfc7498>.

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