Service Programming with Segment Routing

draft-ietf-spring-sr-service-programming-02

Authors :

Francois Clad, Cisco (presenter)
Xiaoahu Xu, Alibaba
Clarence Filsfils, Cisco
Daniel Bernier, Bell Canada
Cheng Li, Huawei
Bruno Decraene, Orange
Shaowen Ma, Juniper
Chaitanya Yadlapalli, AT&T
Wim Henderickx, Nokia
Stefano Salsano, Universita di Roma "Tor Vergata"

IETF108, July 2020

Summary of changes

- Updated SRv6 endpoint behaviors pseudocodes to align with RFC8754 and draft-ietf-spring-srv6-network-programming
- Updated SR-MPLS pseudocodes to follow a similar model
- Added missing pseudocode for the caching flavor of the masquerading proxy
- Added missing behavior identifiers in IANA section
- Integrated feedback received on the mailing list (thanks!)

SRv6 behavior update

- More formal pseudocode (aligned with RFC8754 and draft-ietf-spring-srv6-network-programming)
- High-level behavior unchanged
- Example: SRv6 Static proxy for Inner type IPv4 (sec. 6.1.2.2)

```
When processing an IPv6 packet matching a FIB entry locally
instantiated as an SRv6 static proxy SID for IPv4 traffic, the
following pseudocode is executed.

S01. When an SRH is processed {
S02.   If (Segments Left == 0) {
S03.     Proceed to process the next header in the packet.
S04.   }
S05.   If (IPv6 Hop Limit <= 1) {
S06.     Send an ICMP Time Exceeded message to the Source Address,
         Code 0 (hop limit exceeded in transit),
         Interrupt packet processing and discard the packet.
S07.   }
S08.   max_last_entry = (Hdr Ext Len / 2) - 1
S09.   If ((Last Entry > max_last_entry) or
           (Segments Left > (Last Entry + 1))) {
S10.     Send an ICMP Parameter Problem message to the Source Address,
         Code 0 (Erroneous header field encountered),
         Pointer set to the Segments Left field,
         Interrupt packet processing and discard the packet.
S11.   }
S12.   Decrement Hop Limit by 1.
S13.   Decrement Segments Left by 1.
S14.   Copy Segment List[Segments Left] from the SRH to the
       Destination Address of the IPv6 header.
S15.   If (Upper-layer header type != 4 (IPv4)) {
S16.     Resubmit the packet to the IPv6 module for transmission to
         the new destination.
S17.   }
S18.   Perform IPv6 decapsulation.
S19.   Submit the packet to the IPv4 module for transmission on
       interface IFACE-OUT via NH-ADDR.
S20. }
```

SRH processing

```
When processing the Upper-layer header of a packet matching a FIB
entry locally instantiated as an SRv6 static proxy SID for IPv4
traffic, the following pseudocode is executed.

S01. If (Upper-layer header type != 4 (IPv4)) {
S02.   Process as per [I-D.ietf-spring-srv6-network-programming] Section 4.1.1
S03. }
S04. Perform IPv6 decapsulation.
S05. Submit the packet to the IPv4 module for transmission on
     interface IFACE-OUT via NH-ADDR.
```

Upper-layer header processing

```
When processing an IPv4 packet received on the interface IFACE-IN and
with a destination address that does not match any address of IFACE-
IN, the following pseudocode is executed.

S01. Retrieve the CACHE entry associated with IFACE-IN.
S02. If the CACHE entry is not empty {
S03.   Decrement the TTL and adjust the checksum accordingly.
S04.   Perform IPv6 encapsulation with an SRH
         Source Address of the IPv6 header is set to CACHE.SA,
         Destination Address of the IPv6 header is set to
         CACHE.LIST[0],
         Next Header of the SRH is set to 4 (IPv4),
         Segment List of the SRH is set to CACHE.LIST.
S05.   Submit the packet to the IPv6 module for transmission to the
       next destination.
S06. }
```

Return traffic processing

SR-MPLS behavior update

- More formal pseudocode
- High-level behavior unchanged
- Example: SR-MPLS Static proxy for Inner type IPv4 (sec. 6.1.1.2)

```
When processing an MPLS packet whose top label matches a locally
instantiated MPLS static proxy SID for IPv4 traffic, the following
pseudocode is executed.

S01. POP all labels in the MPLS label stack.
S02. Submit the packet to the IPv4 module for transmission on
     interface IFACE-OUT via NH-ADDR.
```

SID processing

```
When processing an IPv4 packet received on the interface IFACE-IN and
with a destination address that does not match any address of IFACE-
IN, the following pseudocode is executed.

S01. Retrieve the CACHE entry associated with IFACE-IN.
S02. If the CACHE entry is not empty {
S03.    Decrement the TTL and adjust the checksum accordingly.
S04.    PUSH all labels from the retrieved CACHE entry.
S05.    Submit the packet to the MPLS module for transmission as per
        the top label in the MPLS label stack.
S06. }
```

Return traffic processing

SRv6 masquerading with caching

- Caching flavor allows the proxy to support NF-generated packets
- Caching mechanism is similar to dynamic proxy (sec. 6.2)
- Processing of returning packets that already contain an SRH is unchanged

```
The caching flavor of the SRv6 masquerading proxy is enabled by:

o  Adding the following instruction between lines S14 and S15 of the
   masquerading pseudocode in Figure 23.

(... S14.   Copy Segment List[0] from the SRH to the Destination
            Address of the IPv6 header.
S14.1. Copy the IPv6 encapsulation in a CACHE entry associated with
       the interface IFACE-IN.
(S15.   Submit the packet to the IPv6 module for transmission on
        interface IFACE-OUT via NH-ADDR.)

o  Updating the de-masquerading pseudocode such that, in addition to
   the SRH processing in Figure 24, the following pseudocode is
   executed when processing an IPv6 packet (received on the interface
   IFACE-IN and with a destination address that does not match any
   address of IFACE-IN) that does not contain an SRH.

S01. Retrieve the CACHE entry associated with IFACE-IN.
S02. If the CACHE entry is not empty {
S03.   If (IPv6 Hop Limit <= 1) {
S04.     Send an ICMP Time Exceeded message to the Source Address,
         Code 0 (hop limit exceeded in transit),
         Interrupt packet processing and discard the packet.
S05.   }
S06.   Decrement Hop Limit by 1.
S07.   Update the IPv6 encapsulation according to the retrieved CACHE
       entry.
S08.   Submit the packet to the IPv6 module for transmission to the
       next destination.
S09. }
```

SRv6 behavior identifiers update

- IANA SRv6 Endpoint Behaviors table (sec. 10.1) updated with separate entries for masquerading proxy flavors
  - Aligned base registry in draft-ietf-spring-srv6-network-programming

Value	Description	Reference
TBA1-1	End.AN - SR-aware function (native)	[This.ID]
TBA1-2	End.AS - Static proxy	[This.ID]
TBA1-3	End.AD - Dynamic proxy	[This.ID]
TBA1-4	End.AM - Masquerading proxy	[This.ID]
TBA1-5	End.AM - Masquerading proxy with NAT	[This.ID]
TBA1-6	End.AM - Masquerading proxy with Caching	[This.ID]
TBA1-7	End.AM - Masquerading proxy with NAT & Caching	[This.ID]

Next steps

- Seek WG input and feedback