Generalized Arguments of SRv6 Segment

draft-Im-spring-srv6-generalized-arguments-00

Zhenbin Li, Jianwei Mao, Cheng Li @ Huawei
Background

In these years, some new features are created:

- *Network slicing, IOAM, Alternate Marking, APN6, DetNet, etc.*

The instructions/commands of these new features can be processed at:

1. All nodes along an SR path: using the IPv6 Hop-by-Hop Options header (HBH).
2. Endpoints of an SR path: using the IPv6 Destination Options Header (DOH) or the SRH TLV.

The usage of the Options or TLVs will cause two issues:

1. **Lengthening** the packet header, and **reducing** transmission efficiency.
2. **Making** the forwarding processing **complex**, affecting forwarding performance.

Besides, another issue:

In the SRv6 C-SID compression (NEXT Flavor) solution, if all the C-SIDs of the SID list can be put in the IPv6 DA of a packet, 

→ **there is no SRH or DOH-before-SRH** anymore after the compression  ➔ **No space** for the instructions using DOH / SRH TLV
In order to address these challenges, use the Arguments of the SRv6 SID to carry those instructions.

Benefits:
1. Reducing the needed space of the IPv6 extension header or SRH TLV \(\Rightarrow\) Reduce the transmission overhead.
2. The SRv6 Arguments can be read and processed as a part of the IPv6 address
\(\Rightarrow\) Avoid processing the extension header or SRv6 TLV behind the basic IPv6 header \(\Rightarrow\) Better forwarding performance.
3. The instructions for the SRv6 and the new features are all put in the Arguments part of the SRv6 SID or IPv6 address
\(\Rightarrow\) Unify and simplify the packet processing.

In addition, there are several kinds of Arguments for the SRv6 End SID and End.X SID, which need to be compatible:
1. SRv6 C-SID compression (NEXT Flavor): using Arguments to carry multiple C-SIDs.
2. SRv6 C-SID compression (REPLACE Flavor): using Arguments to carry the CL field.
3. SRv6 C-SID compression (NEXT & REPLACE Flavor): using Arguments to carry multiple C-SIDs and the CL field.
What is Generalized Arguments?

- makes the SRv6 Arguments structured and generalized
- allocates spaces for the instructions of multiple new features and SRv6 SID

### SRv6 SID

<table>
<thead>
<tr>
<th>Locator</th>
<th>Function</th>
<th>Arguments</th>
</tr>
</thead>
</table>

#### Method A: Template

- Network devices have configured a template, then devices read and process the content of the Arguments according to it.

Example:

<table>
<thead>
<tr>
<th>SRv6 SID</th>
<th>Locator</th>
<th>Function</th>
<th>Arguments (z-bit)</th>
</tr>
</thead>
</table>

0 1 2 3 ... x y z

| Feature A (e.g. Network Slicing) | Feature B (e.g. IOAM) | Feature C (e.g. APN6) |
What is Generalized Arguments?

**Method B: Bitmap**

- Define a bitmap in the Arguments, used as an indicator.
- Each bit in the bitmap indicates whether the instructions of a specific feature exist and are valid.

<table>
<thead>
<tr>
<th>SRv6 SID</th>
<th>Locator</th>
<th>Function</th>
<th>Arguments</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 1 2 3 ...</td>
<td>Feature A (e.g. Network Slicing)</td>
<td>Feature B (e.g. IOAM)</td>
<td>Feature C (e.g. APN6)</td>
</tr>
</tbody>
</table>

**Bitmap**

MSB: A B C ... Feature A (e.g. Network Slicing) Feature B (e.g. IOAM) Feature C (e.g. APN6)

LSB: Feature A (e.g. Network Slicing) Feature B (e.g. IOAM) Feature C (e.g. APN6) ... C B A

Bitmap
Consideration of SRv6 C-SID Compression scenario

Background:
• For NEXT or NEXT-and-REPLACE flavor, it is required to shift the C-SID in the SRv6 SID.

Generalized Arguments:
• C-SIDs are always placed from the most significant bit (MSB).
• The remaining part of the Generalized Arguments following the C-SIDs should not be shifted.

<table>
<thead>
<tr>
<th>MSB:</th>
<th>C-SID 1</th>
<th>C-SID 2</th>
<th>... ...</th>
<th>C-SID n</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>... ...</th>
<th>Feature A</th>
<th>Feature B</th>
<th>Feature C</th>
</tr>
</thead>
<tbody>
<tr>
<td>LSB:</td>
<td>C-SID 1</td>
<td>C-SID 2</td>
<td>... ...</td>
<td>C-SID n</td>
<td>Feature A</td>
<td>Feature B</td>
<td>Feature C</td>
<td>... ...</td>
<td>C</td>
<td>B</td>
<td>A</td>
</tr>
</tbody>
</table>

Bitmap
Next step

To be defined:

- Which bit in the bitmap corresponds to which feature
- How long is the space of Generalized Arguments allocated for a specific feature
- What instructions/fields of the specific feature need to be carried in the Generalized Arguments