IPv6 Support for Segment Routing: SRv6+

draft-bonica-spring-srv6-plus-04
R. Bonica, S. Hegde, Y. Kamite, A. Alston, D. Henriques, J. Halpern, J. Linkova, G. Chen
Topological Instructions Versus Service Instructions
Topological Instructions

• Executed on segment ingress node
• Cause a routing action
  • Forward the packet to the segment egress node
• Details
  • Overwrite IPv6 Destination Address with the address of the segment egress node
  • Forward the packet to the segment egress node, either through a specified link or least cost path
• Encoded in IPv6 Routing header (RH)
Service Instructions

- Executed on segment egress node
- Per-segment service instructions
  - Executed on any segment endpoint
  - Typically do not influence routing
    - Example: expose a packet to a firewall rule
  - Encoded in Destination Option header (DOH) that precedes RH
- Per-path service instructions
  - Executed on final segment endpoint
  - Typically influence demultiplexing and forwarding of packet payload
    - Example: de-encapsulate and forward the payload through VPN interface
  - Encoded in DOH that precedes upper-layer header
Why Decouple Topological Instructions from Service Instructions
Using The Most Appropriate IPv6 Extension Header

• RH and topological instructions
  • Both intended to affect routing
  • Both executed on segment ingress node

• DOH preceding RH and per-segment service instruction
  • Both have can have scope beyond routing
    • Example: both can carry an OAM instruction
  • Both executed on any segment egress node

• DOH preceding upper-layer header and per-path service instruction
  • Both have can have scope beyond routing
    • Example: both can influence de-encapsulation and payload forwarding
  • Both executed on the final segment egress node only (i.e., the path egress node)
Simplified Identifier Semantics

• A service instruction identifier (SII) identifies a service instruction
  • Appears in a DOH
  • Not polluted with SID or IPv6 Address semantics

• A SID identifies a segment and the topological instruction that controls it
  • Appears in the RH
  • Not polluted with SII or IPv6 address semantics

• An IPv6 address identifies an interface
  • Appears in IPv6 header
  • Not polluted with SII or SID semantics

• Never copy an identifier of one type into a field that is meant for an identifier of another type
Cost / Benefit Analysis
Cost

- One more layer of indirection
  - SFIB maps SIDs to IPv6 addresses
  - Required to maintain separation between SIDs and IPv6 addresses

- One more RH type
  - Albeit, simpler

- Two new Destination Options
Benefits

• Simplified RH
  • No need for Tag field
  • No need for TLVs

• SID identifies, but does not contain, an instruction
  • Therefore, the SID can be encoded in relatively few bits

• The RH can be short, even when the SID list that it contains is long
  • Regardless of how strictly and loosely routed segments are interspersed in the SID list
  • Regardless of the network numbering scheme

• No need to augment IPv6 OAM
Benefits (continued)

• Mix and Match deployment
  • RH with legacy demultiplexing (e.g., RH followed by vxlan header)
  • Least cost routing (no RH) with DOH for demultiplexing
• IPv6 Authentication header can be used to authenticate RH and DH
• Overall simplicity
  • Existing draft cover the subject
## SRH Versus CRH Overhead

<table>
<thead>
<tr>
<th>SIDs</th>
<th>SRv6 SRH (128-bit SID)</th>
<th>SRv6+ CRH (16-bit SID)</th>
<th>SRv6+ CRH (32-bit SID)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>24</td>
<td>16</td>
<td>16</td>
</tr>
<tr>
<td>2</td>
<td>40</td>
<td>16</td>
<td>16</td>
</tr>
<tr>
<td>3</td>
<td>56</td>
<td>16</td>
<td>24</td>
</tr>
<tr>
<td>4</td>
<td>72</td>
<td>16</td>
<td>24</td>
</tr>
<tr>
<td>5</td>
<td>88</td>
<td>24</td>
<td>32</td>
</tr>
<tr>
<td>6</td>
<td>104</td>
<td>24</td>
<td>32</td>
</tr>
<tr>
<td>7</td>
<td>120</td>
<td>24</td>
<td>40</td>
</tr>
<tr>
<td>8</td>
<td>136</td>
<td>24</td>
<td>40</td>
</tr>
<tr>
<td>9</td>
<td>152</td>
<td>32</td>
<td>48</td>
</tr>
<tr>
<td>10</td>
<td>168</td>
<td>32</td>
<td>48</td>
</tr>
<tr>
<td>11</td>
<td>184</td>
<td>32</td>
<td>56</td>
</tr>
<tr>
<td>12</td>
<td>200</td>
<td>32</td>
<td>56</td>
</tr>
<tr>
<td>13</td>
<td>216</td>
<td>40</td>
<td>64</td>
</tr>
<tr>
<td>14</td>
<td>232</td>
<td>40</td>
<td>64</td>
</tr>
<tr>
<td>15</td>
<td>248</td>
<td>40</td>
<td>72</td>
</tr>
<tr>
<td>16</td>
<td>264</td>
<td>40</td>
<td>72</td>
</tr>
</tbody>
</table>
Implementation

• LINUX Demo
• JUNOS PoC
Next Steps

• SPRING WG to adopt draft-bonica-spring-srv6-plus
• 6man WG to adopt
  • Draft-bonica-6man-com-rtg-hdr
  • Draft-bonica-6man-vpn-dest-opt
  • Draft-bonica-6man-seg-end-opt