FORWARDING ACTIONS INDICATORS

Kireeti Kompella
IETF 112
draft-kompella-mpls-mspl4fa
DISCUSSIONS IN AND OUT OF THE OPEN DT REGARDING ISD AND PSD

1. When should some indicator/data be in the ISD vs in the PSD?
2. What should be said about the PSD in the ISD
3. How should extensions of the indicators be handled?
4. What about standard indicators that are not understood by someone?
5. What about “non-standard” indicators?
   • Accommodating “user-defined” (provider-defined) indicators and data fields would be friendly and powerful
I. ISD

Philosophy:

• In-stack data must be processed with some urgency
  • if not, this indicator and data shouldn't be part of ISD
• ISD must be coded compactly, and processed fast
• ISD must follow the label format (BoS MUST be respected)
Philosophy

1. PSD need not be ultra-compact, nor does it have to fit into label fields, nor respect the BoS bit.
2. PSD should be self-describing. A TLV-type approach may be reasonable.

Given (2), too much detail in the ISD is redundant, and can lead to confusion

• If ISD says field $X$ is present, but $X$ is not in PSD, what should be done?
• If ISD says field $X$ is absent, but $X$ is in PSD, what should be done?
2B. SUGGESTION

- Indicators only say who should look at PSD
  - An optimization, where the “worst case” is that every hop looks at PSD
  - This means scanning the entire label stack (!)
- Use two bits:
  00: no one needs to look at PSD
  01: every node MAY look at PSD, but only egress MUST
  10: every hop SHOULD look at PSD
  11: every hop MUST look at PSD; a node that cannot must drop packet

FAILSAFE: egress will have popped entire label stack; it can tell whether there is PSD (see previous presentation); if so, it can parse and process it
3A. HANDLING EXTENSIONS

- Use a bit, the E bit to indicate that the indicators continue
  - When \( E = 0 \), indicators end
- Standard data fields start after indicators are done
- If you invert this, there is an “end of indicators” bit
  - This matches the “Bottom of Stack” logic
  - When \( S = 1 \), labels end
### 3B. CURRENT: “Bottom of Indicators” BIT
(if you invert the E bit)

#### Label Value

<table>
<thead>
<tr>
<th>Label Value</th>
<th>TC</th>
<th>S</th>
<th>TTL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Other labels</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>One FAI Block</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Bottom of stack</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

- **Indicator section**
- **Standard data**

Where does this start? Where does it end?
4. WHAT ABOUT STANDARD DATA?
5. WHAT ABOUT USER-DEFINED DATA?

- So, what if 50 standards data fields are defined, but an implementation only understands 5 of them? How does it know when the standard data is over?
- If there is user-defined data, how can an implementation know where it starts and where it ends?
- Solution: redefine the “bottom of indicators” bit for all these purposes
  - Think of this as the “bottom of section” bit
### 3-5. PROPOSAL: “Bottom of Section” BIT (modeled after “BoS” bit)

<table>
<thead>
<tr>
<th>Label Value</th>
<th>TC</th>
<th>S</th>
<th>TTL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Other labels</td>
<td>0 0</td>
<td>0 0</td>
<td>Indicator section</td>
</tr>
<tr>
<td>One FAI Block</td>
<td>0 1</td>
<td>0</td>
<td>Standard data</td>
</tr>
<tr>
<td>Other labels Bottom of stack</td>
<td>0 0 1</td>
<td>0 0 1</td>
<td>User-defined actions</td>
</tr>
</tbody>
</table>

The diagram illustrates the structure of the “Bottom of Section” bit ("BoS" bit) in the context of the proposed "Bottom of Section" BIT, showing the allocation of bits for different sections such as TC, S, and TTL, along with the implications for user-defined actions.
### IMPLEMENTATION X THAT ONLY UNDERSTANDS SOME FIELDS

<table>
<thead>
<tr>
<th>Label Value</th>
<th>TC</th>
<th>S</th>
<th>TTL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Other labels</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>One FAI Block</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>What Implementation X Understands</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>What Implementation X Doesn’t Understand</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

**Indicator section**
- TC: 0
- S: 0
- TTL: 1

**Standard data**
- TC: 1
- S: 0
- TTL: 0

**User-defined actions**
- TC: 0
- S: 0
- TTL: 0