Encoding multipath information in PCEP

Motivated by the need to encode multiple Segment Lists in SR-TE Policies, but results are applied to RSVP-TE as well.

Terminology:

- ERO == Explicit Route Object
- LSP == PCEP LSP (identified by LSP-IDENTIFIERS TLV)

Motivation

We want the PCE to be able to install not just a single path, but multiple ECMP/UCMP paths from source to destination.

https://tools.ietf.org/html/draft-ietf-spring-segment-routing-policy-03

> A dynamic candidate path expresses an optimization objective and a set of constraints. The headend (potentially with the help of a PCE) computes the solution Segment-List (or set of Segment-Lists) that solves the optimization problem.
>
> If a candidate path is associated with a set of Segment-Lists, each Segment-List is associated with a weight for weighted load balancing (refer Section 2.11 for details). The default weight is 1.

LSP objectives and constraints

The PCE runs a computation algorithm that takes objectives and constraints as the input and produces a set of EROs as the output. **PROBLEM TO SOLVE:** PCEP lacks a way to signal those EROs to the head-end.

SR Policies statically configured on the PCC may have more than one Segment List in a Candidate Path. **PROBLEM TO SOLVE:** PCEP lacks a way to signal multiple Segment Lists to the PCE.

Proposed solutions

Solution 1: allocate a different LSP for every ERO and create a new association type to bind these LSPs together.

Solution 2: one LSP contains many EROs.

- Option 2a: define a new object for ERO attributes.
- Option 2b: follow RFC 8623 for encoding multiple EROs.

.

Analysis of Solution 1

Solution 1: allocate a different LSP for every ERO and create a new association type to bind these LSPs together.

- May work for PCE-initiated LSPs, because the PCE knows how many EROs it computed and can instantiate that many LSPs.
- Does not work for PCC-initiated LSPs, because the PCC would need to know how many EROs the PCE will compute and delegate that many LSPs.
- Opens the possibility of each ERO in the set having different objectives and constraints, which is inconsistent.
- Does not allow all the EROs to be reported/updated/instantiated together, since each LSP is reported/updated/instantiated separately from the others.

Analysis of Solution 2a

Solution 2: one LSP contains many EROs.
Option 2a: define a new object for ERO attributes

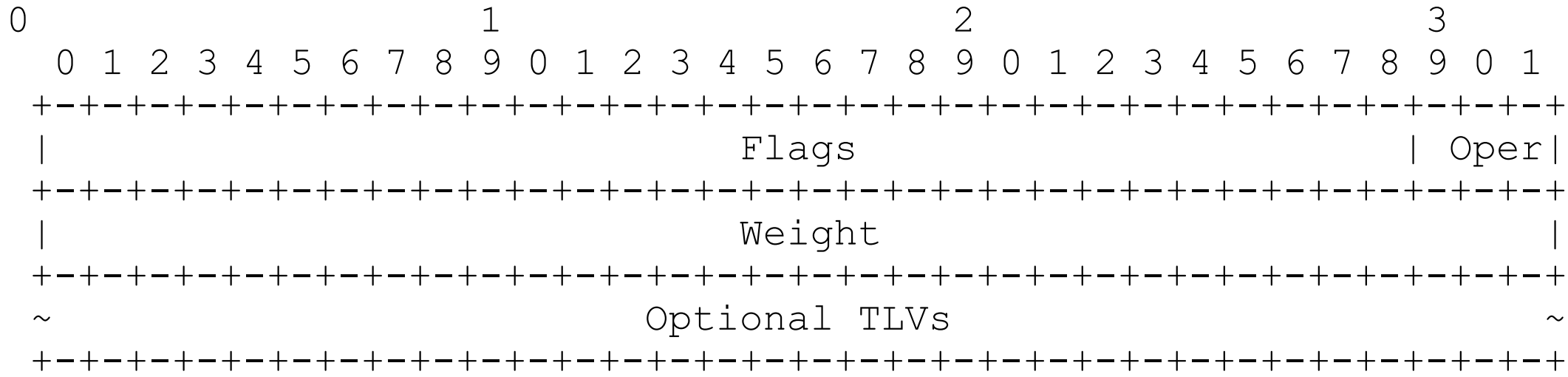
We can replace a single ERO object by multiple ERO objects, separated by a new ERO-ATTRIBUTES object.

Current BNF:
<intended-path> = <ERO>

Proposed BNF:
<intended-path> = <ero-list>
<ero-list> = [<ERO-ATTRIBUTES>]<ERO>[<ero-list>]

Analysis of Solution 2a (cont'd)

Define ERO-ATTRIBUTES object to carry some per-ERO attributes:



Support for ERO-ATTRIBUTES can be negotiated in the OPEN message, thus guaranteeing backward compatibility.

Analysis of Solution 2b

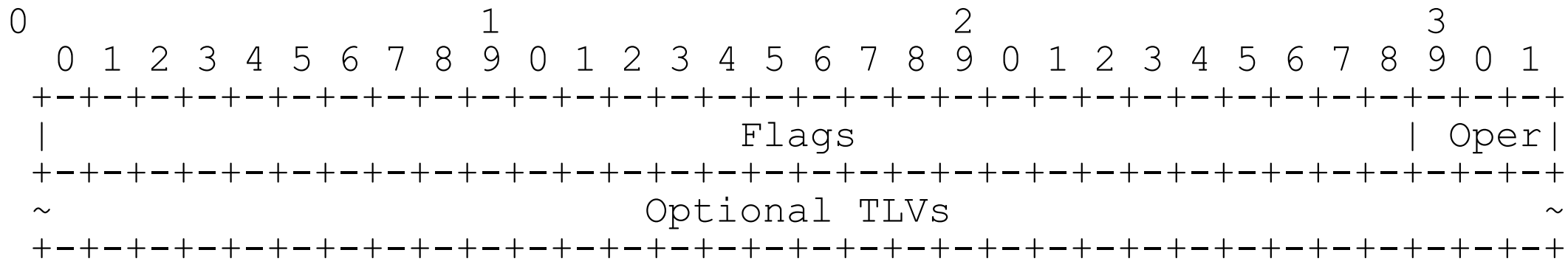
Solution 2: one LSP contains many EROs.
Option 2b: follow RFC 8623 for encoding multiple EROs.

https://tools.ietf.org/html/rfc8623 - Stateful Path Computation Element (PCE) Protocol Extensions for Usage with Point-to-Multipoint TE Label Switched Paths (LSPs)

RFC 8623 already defines a way to carry multiple ERO/RRO objects, by using a special type of END-POINTS object and an S2LS object. We can define a new END-POINTS and S2LS object types for P2P load-balancing and then we can follow the same encoding format as RFC 8623.

Analysis of Solution 2b (cont'd)

S2LS format is almost the same as ERO-ATTRIBUTES, except that it's missing the Weight field:



The weight can be either carried in an optional TLV, or it can be embedded directly as part of the new S2LS object type.

New END-POINTS object type would need to be defined, to specify that the S2LS objects that follow it are for ECMP/UCMP.

Analysis of Solution 2b (cont'd)

For example, suppose the PCE was computing a path from Source A to Destination X and the result was 2 EROs: {A,B,X} and {A,C,X}. Suppose that the 2 EROs have UCMP weights 2 and 3 respectively. Then the PCE would encode this as follows:

```
Common Header
  LSP
  END-POINTS (SRC=A, DEST=X)
   S2LS (O=UP, WEIGHT=2)
    ERO1={A,B,X}
  END-POINTS (SRC=A, DEST=X)
   S2LS (O=UP, WEIGHT=3)
    ERO2={A,C,X}
```

Note that we need to encode 2 END-POINTS objects here if we want to encode 2 S2LS objects, in order to conform to the RBNF of RFC 8623. If both EROs had the same weight (ECMP), then we would not need 2 S2LS objects and we would encode ERO2 directly after ERO1.

Comparison of solutions

Solution 1: allocate a different LSP for every ERO and create a new association type to bind these LSPs together.

- Multiple problems with this, would not really fit into the database structure

Solution 2: one LSP contains many EROs.

Option 2a: define a new object for ERO attributes

- Separate RBNFs for P2P and P2MP

Option 2b: follow RFC 8623 for encoding multiple EROs.

- Common RBNF for P2P and P2MP