draft-ietf-pim-reserved-bits-00

Stig Venaas, stig@cisco.com

Alvaro Retana, alvaro.retana@huawei.com
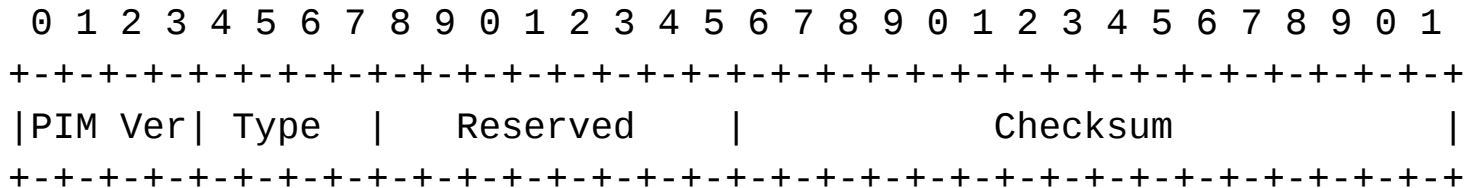
Use of reserved bits

- Several pim message types use reserved bits
  - Not clear that reserved bits should be per message type
  - RFCs using reserved bits should have updated RFC 4601 or RFC 7761
- This draft fixes the above and defines a registry as below.

```
Type bit(s) Name    Reference
------------------------------------------------------------
  4 7    No-Forward [RFC5059]
 10 4-7 Sub-type    [RFC5015]
 12 7    No-Forward [RFC8364]
 13 4-7 Extended type [this document]
 14 4-7 Extended type [this document]
 15 4-7 Extended type [this document]
```
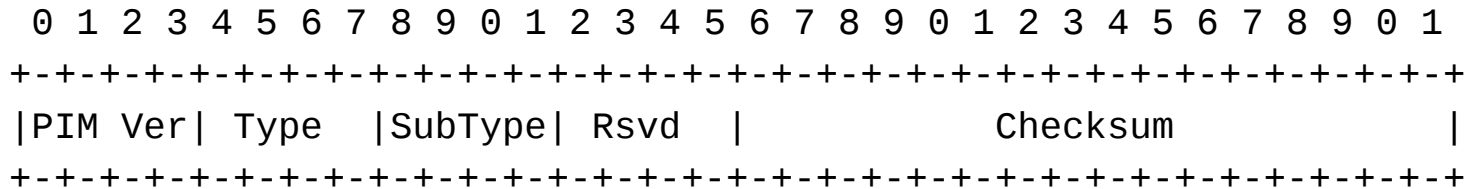
Extending the type space

- The current pim type space is only 4 bits. We have used 0-12.
- The current pim message header is:

```
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|PIM Ver| Type  |    Reserved     |           Checksum          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

- Extend the type space by defining types 13-15 using 4 reserved bits each to define a sub-type (similar to pim DF sub-types).
  - This gives us 3*16 additional pim message types denoted 13.0 – 13.15, 14.0 – 14.15 and 15.0 – 15.15.
  - The header for types 13-15 is defined as below.

```
  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |PIM Ver|  Type  |SubType| Rsvd  |             Checksum        |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Next steps

- The are drafts requesting new message types:
  - draft-ietf-pim-null-register-packing
  - draft-liu-pim-assert-packing
- These drafts should make use of the type extension mechanism defined here, so that we don't use up the remaining type space too quickly.

- I believe the draft is ready, or nearly ready, publication. Ready for WGLC?
  - Want to get this published soon, so that new RFCs can make use of the extended type space