draft-petrie-grow-mrt-bmp-01

IETF 124, November 2025, Montréal, Canada

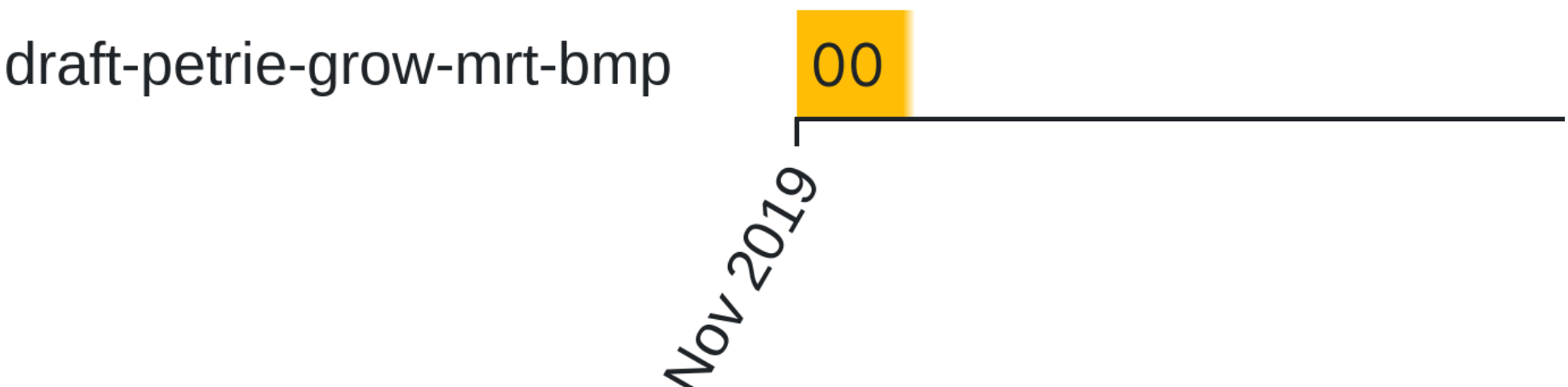
Colin Petrie (NTT)
Luuk Hendriks (NLnet Labs)

Motivation

- Store and share BMP *messages*
  - And really only messages, like `BGP4MP_MESSAGE_AS4`
    - No new `TABLE_DUMP_V2` types
    - Not touch `PEER_INDEX_TABLE`, no `RIB_ENTRIES`
- MRT is the de facto format used by route collection projects
  - Adding BMP message support opens up the opportunity to archive BMP feeds

–00?

–00?



Why –00 stalled

No ergonomic solution for stateful parsing:

- BMP RouteMonitoring encapsulate BGP UPDATEs
  - Correctly parsing BGP UPDATEs requires context from BGP OPENs:
    - 32 vs 16 bit ASNs
    - ADDPATH
    - multi-label
    - …

How MRT handles this for BGP UPDATEs

RFC6396 (original MRT Standard):

```
  4.4.2.  BGP4MP_MESSAGE Subtype . . . . . .
  4.4.3.  BGP4MP_MESSAGE_AS4 Subtype . . . . .
```

RFC8050 (adding ADDPATH support):

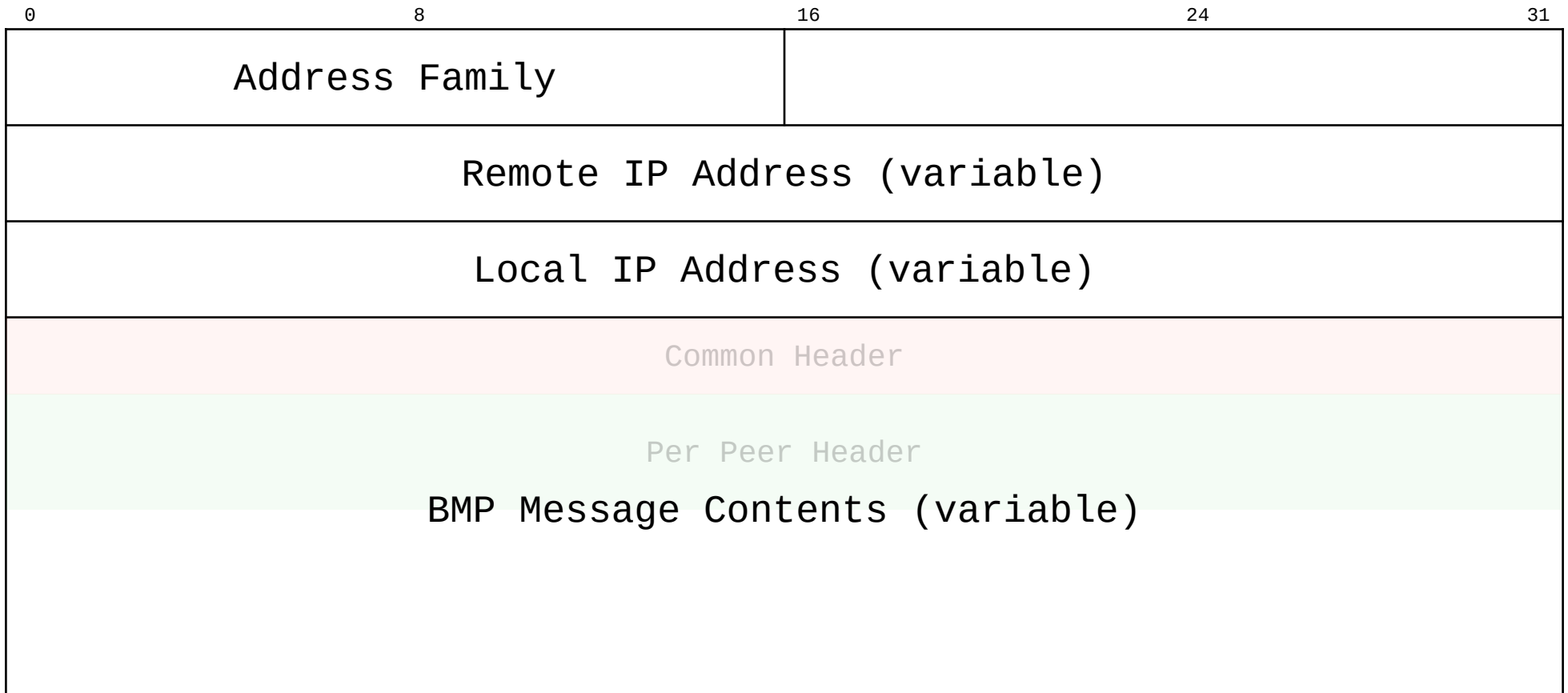
```
  o  BGP4MP_MESSAGE_ADDPATH
  o  BGP4MP_MESSAGE_AS4_ADDPATH
  o  BGP4MP_MESSAGE_LOCAL_ADDPATH
  o  BGP4MP_MESSAGE_AS4_LOCAL_ADDPATH

  o  RIB_IPV4_UNICAST_ADDPATH
  o  RIB_IPV4_MULTICAST_ADDPATH
  o  RIB_IPV6_UNICAST_ADDPATH
  o  RIB_IPV6_MULTICAST_ADDPATH
  o  RIB_GENERIC_ADDPATH
```

... a cumbersome process, at best

This draft: adding support for BMP messages

- Just one[1] single new MRT message type: `BMP`
- No subtypes: we leverage the BMP Common Header
- Simple message format: we leverage the Per Peer Header

[1] Well, two, because `BMP_ET` for Extended Timestamps


0
8
16
24
31
Address Family
Remote IP Address (variable)
Local IP Address (variable)
Common Header
Per Peer Header
BMP Message Contents (variable)

The stateless parsing problem?

✓ 32 vs 16 bit ASNs: flag in Per Peer Header

- With draft-ietf-grow-bmp-tlv↗, we'll get the Stateless Parsing TLV:
  - Solves the ADDPATH case,
  - and multi-label,
  - and anything else that might popup later (so, we're flexible!)

But

> If no Stateless Parsing TLV is present in a Route Monitoring message, the receiver MUST fall back to use capabilities present in the BGP Open PDU contained in the relevant BMP Peer Up message in order to properly parse BGP Update PDUs. Each BGP capability is to be encoded in a separate Stateless Parsing TLV.
>
> — draft-ietf-grow-bmp-tlv-19

Is this what we want in MRT, too?

RouteMonitoring with/without Stateless Parsing TLV

Proposal:

> In MRT, if context for parsing is required, Stateless Parsing TLVs MUST be present in the encapsulated Route Monitoring message. Without Stateless Parsing TLVs, the Route Monitoring message contains either 32bit or 16bit ASNs (per the Per Peer Flag), but no Path IDs, no multi-label, …

- Producers of MRT files presumably have the required state available to attach the TLVs if necessary;
- Consumers of MRT files do not have to piece together multiple messages, making the process less error-prone, and allowing MRT files to be split up.

Relation to BMP Snapshots (draft-lucente-grow-bmp-offline-02↗)

BMP Snapshots could be leveraged to

- provide `PEER_INDEX_TABLE` functionality (Peer Up Notifications)
- provide `TABLE_DUMP_V2` functionality (Route Monitoring messages)

Storing all messages comprising a BMP Snapshot would be the equivalent of a ‘dump’ or ‘bview’.

Questions to the WG

- BMP in MRT, useful?

If so,

- one single MRT message type, the right approach?
- rely on BMP Snapshots to facilitate ‘RIB dump’-like archives, sensible?