

Global Routing Operations
Internet-Draft
Updates: 7854 (if approved)
Intended status: Standards Track
Expires: 23 October 2026

L. Hendriks
NLnet Labs
P. Lucente
C. Cardona
C. Petrie
NTT
21 April 2026

BMP Snapshots
draft-lucente-grow-bmp-offline-03

Abstract

BMP (BGP Monitoring Protocol) is perfectly suited for real-time consumption but less ideal in stream processing and off-wire historical scenarios. The issue is that the necessary information to produce a complete view and enabling correct processing of all messages in the stream, is only sent out at the beginning of the BMP session. This document introduces the concept of BMP Snapshots, enabling BMP stations to synchronize mid-stream, and, providing the basis for self-contained, time-binned archiving of BMP data.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 23 October 2026.

Copyright Notice

Copyright (c) 2026 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document.

Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction	3
1.1. Exporter vs Station	4
1.1.1. Considerations regarding alternative approaches	4
1.2. Flexibility and extensibility	5
2. Terminology	6
2.1. Terms in this document	6
3. Snapshot Message	6
3.1. Wireformat	6
3.2. Start-of-snapshot	8
3.3. End-of-snapshot	8
4. Snapshot Information TLVs	8
4.1. Snapshot Id TLV	8
4.2. Snapshot Error TLV	8
4.3. Optional meta-data TLVs	9
5. Backwards (in)compatibility	10
5.1. Possible workaround	10
6. Third party off-wire encoding formats	11
7. Operational Considerations	11
7.1. End-of-snapshot with an Error TLV	11
7.2. Snapshot Id scheme	12
7.2.1. Global uniqueness	12
7.2.2. Increasing identifiers	12
7.3. Requesting/triggering snapshots	12
7.3.1. Timer-based	12
7.3.2. Manual triggers	13
7.3.3. Out-of-band request via other protocols	13
8. Discussion	13
8.1. No EoRs in Snapshots	13
9. Security Considerations	14
10. IANA Considerations	14
11. References	15
11.1. Normative References	15
11.2. Informative References	16
Acknowledgements	16
Authors' Addresses	16

1. Introduction

BMP (BGP Monitoring Protocol) is perfectly suited for real-time consumption but less ideal in stream processing and off-wire historical scenarios. The issue is that the necessary information to produce a complete view and enabling correct processing of all messages in the stream, is only sent out at the beginning of the BMP session. This document introduces the concept of BMP Snapshots, enabling BMP stations to synchronize mid-stream, and, providing the basis for self-contained, time-binned archiving of BMP data.

At the very start of a BMP session, two types of information are sent from exporter to monitor. Firstly, session state, describing all established BGP sessions on the monitored router. Secondly, the RIB contents, i.e. all the routes the monitored router has learned thus far. Missing either of these cause different problems: the session state contains information (Capabilities in the BGP Open messages, encapsulated in the Peer Up Notification) crucial to correctly parse messages in the stream, and the RIB contents represent the starting state to apply deltas (BGP Updates encapsulated in Route Monitoring messages) in the BMP stream to. In order to construct a complete and correct view of the network, one can not rely on those deltas alone. While constructing the state purely on deltas might, eventually, get close to the 'actual' state of the network, there is no control over how long one is working with an incorrect state, nor is there any guarantee that it ever will be correct.

There is no mechanism in BMP to facilitate the synchronisation of either the session state or the RIB contents mid-stream. Stateless Parsing TLVs [I-D.ietf-grow-bmp-tlv] could provide the required parsing information, but there is no guarantee these are present and even if they are, they do not cover for all the missing session information.

This document introduces Snapshots, enabling synchronisation of both BGP session information and RIB contents anywhere in a BMP session. A Snapshot is not one single message, but a collection of messages, all containing a Snapshot Id TLV (introduced in this document) carrying the same Snapshot Id. The session state and RIB contents are carried in Peer Up Notification messages and Route Monitoring messages, respectively, exactly like the initial synchronisation upon establishment of a new BMP session. In addition to the Peer Up Notification and Route Monitoring messages, two Snapshot Messages (introduced in this document) are included in a Snapshot. One preceding all the other messages, signalling the start of the Snapshot and optionally containing TLVs carrying metadata about the Snapshot. And one Snapshot Message at the very end, signalling the end of the Snapshot, again optionally containing TLVs carrying metadata. These TLVs are described in Section 4.3.

1.1. Exporter vs Station

The new concepts described in this document are not restricted to either the BMP exporter or the BMP station, as all newly introduced pieces are in-protocol and do not rely on any specific characteristic of either exporter or station. However, as the process of emitting a Snapshot can presumably be expensive, it is not to be expected that BMP exporter implementations on routers will support the concepts introduced in this document. A BMP station on the other hand, likely has more resources available and will have received all the necessary information (in terms of session state and RIB contents) from the router already. To not burden limited routers more than necessary, the authors assume the 'first' BMP station implements the concepts described so any stations 'downstream' can leverage the Snapshot functionality, while not imposing any additional load on the router originally emitting the BMP stream.

1.1.1. Considerations regarding alternative approaches

// This section is included as a record of discussion, and is to be
// removed/reduced at a later stage.

Emitting all contents of a RIB is very similar, if not identical, to the process of a Route Refresh [RFC2918]. For routers exporting BMP streams, the assumption is that they support Route Refresh. The authors evaluated if mimicking a Route Refresh could provide functionality similar to Snapshots, but at lower (computational) cost. While a full overview of key features/requirements is listed in Table 1, it was concluded that a Route Refresh lacks (too) many of the features of a Snapshot. Moreover, it would still be considered too expensive to perform on any regular basis on routers.

	Snapshots	Route Refresh
Feature/quality:		
In-protocol	y	y
Signal begin/end	y	y
RIB contents	y	y
Session info	y	n
Distinguish sync vs live	y	n
Distinguish between syncs	y	n
Per address family	y	y
Additional metadata	y	n
Extensible	y	n
Expensive	y	Less
Protocol requirement:		
New message type or TLV	both	either

Table 1: Overview of feature/requirement comparison between the approach described in this document and a approach based on Route Refresh messages

1.2. Flexibility and extensibility

By building upon TLVs, supported in all BMP message types from BMPv4, the Snapshot approach imposes minimal requirements over the initial synchronisation in BMP today. Furthermore, if at any point in the future another message type needs to be incorporated in a Snapshot, it will be a simple matter of attaching the Snapshot Id TLV to those messages. No existing message types have to be adapted to support Snapshots.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 RFC 2119 [RFC2119] RFC 8174 [RFC8174] when, and only when, they appear in all capitals, as shown here.

2.1. Terms in this document

Exporter: The sender of BMP messages over the TCP session. This could be a router, or a BMP Station sending BMP messages onwards.

Station: The receiver of BMP messages. Colloquially also often called the 'collector'. A Station receives BMP messages from an Exporter. If a Station sends out BMP messages, it is considered an Exporter for those egress connections.

Snapshot: The logical concept of a collection of Peer Up Notification messages and Route Monitoring messages, preceded and followed by a Snapshot Message. A Snapshot is not a single message/PDU itself.

Snapshot Message: A BMP message type, introduced in this document, to signal the start or end of a Snapshot. The first and last message of a Snapshot are of this message type.

Snapshot message A BMP message that is part of a Snapshot, but not of type Snapshot Message. For example, a Route Monitoring message that is part of a Snapshot is considered a Snapshot message.

3. Snapshot Message

Every Snapshot contains exactly two messages of the Snapshot Message type: one Snapshot Message preceding all the Peer Up Notifications and Route Monitoring messages containing the actual data, and one Snapshot Message to signal the end. Both follow the same wireformat.

3.1. Wireformat

The Snapshot Message starts with a BMP Common Header as defined in Section 4.1 of [RFC7854]. The Common Header is directly followed by TLVs. The Snapshot Id TLV defined in Section 4.1 MUST be present and SHOULD be the first of the TLVs. All additional TLVs listed in Section 4.3 are optional.

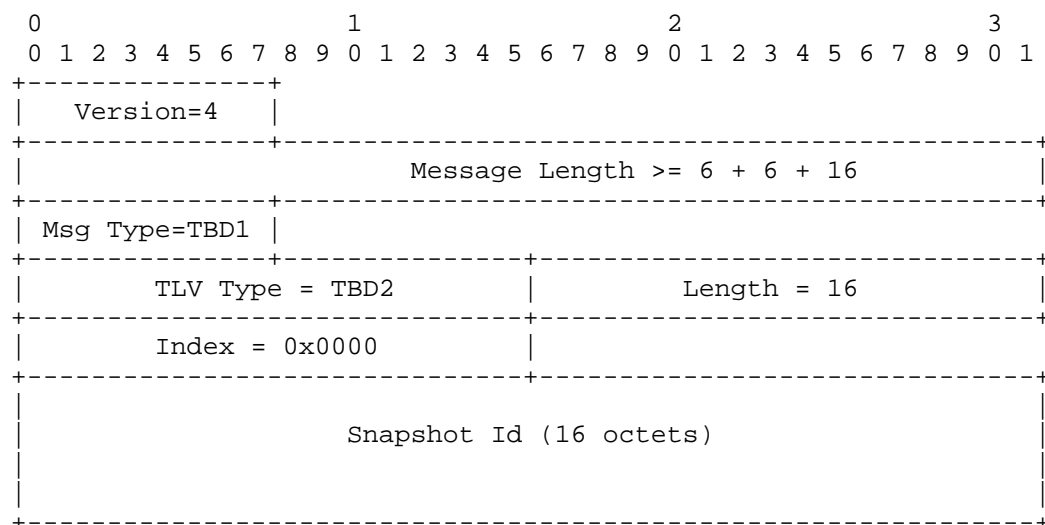


Figure 1: The Snapshot Message wireformat, including the BMP Common Header and the Snapshot Id TLV.

The Snapshot (Messages) can be generated by a BMP exporter or by a BMP station, as long as all required data for the Peer Up Notification and Route Monitoring messages pertaining to the Snapshot are available. The Snapshot Message MUST be followed by those messages, with the Snapshot Id TLV attached to every message. Finally, the end of the Snapshot is signalled by another Snapshot Message. An example flow of these messages is visualized in Figure 2.

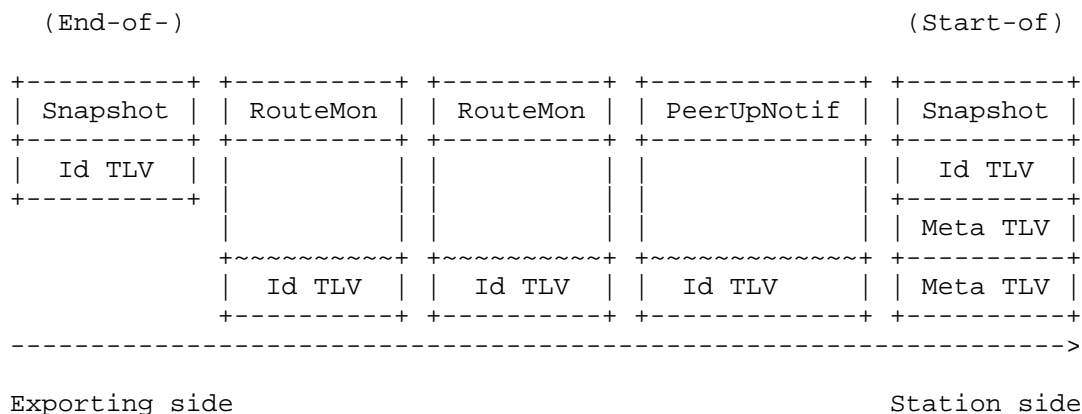


Figure 2: Flow of messages in a Snapshot, from exporter to station

3.2. Start-of-snapshot

The Snapshot Message preceding all other messages in the Snapshot MUST contain the Snapshot Id TLV, and should also contain TLVs providing useful metadata for the consumer of the Snapshot. For example, the timestamp of snapshot generation, and information about the exporter (IP address, software version). Furthermore, they can describe which address families are included in the snapshot, and for which RIB views (e.g. Adj-RIB-In Pre-policy). Note that, as all meta-data TLVs are optional, a consumer of snapshots should never rely on presence of any of these TLVs.

3.3. End-of-snapshot

After all Peer Up Notification and Route Monitoring messages of the Snapshot, the end of the Snapshot is signalled by means of another Snapshot Message. This Snapshot Message MUST contain the Snapshot Id TLV like all other messages comprising the Snapshot. This Snapshot Message can also contain meta-data TLVs, most notably the Snapshot Error TLV. If, for whatever reason, the sending side can not complete the Snapshot, it MUST send a Snapshot Message containing a Snapshot Error TLV to signal the Snapshot is incomplete and/or incorrect. For discussion on operational consequences, see Section 7.1.

4. Snapshot Information TLVs

4.1. Snapshot Id TLV

The Snapshot Id TLV is the only mandatory TLV of a Snapshot Message as defined in Section 3. It is an indexed TLV (as it will be included in Route Monitoring messages, with index zero), structured as defined in Section 4 of [I-D.ietf-grow-bmp-tlv], with a fixed value length of 16 bytes. This allows the use of UUID identifiers, or provides sufficient space for alternative schemes. Different approaches for schemes are discussed in Section 7.

4.2. Snapshot Error TLV

The Snapshot Error TLV is used to signal that a sender can not complete a currently in-flight Snapshot, and MUST be included in the End-of-snapshot Snapshot Message the sender sends to signal the end of the (in this case incorrect/incomplete) Snapshot. Sending out the End-of-snapshot Snapshot Message is crucial, as failing to do so leaves the consuming end in a state where it expects more messages pertaining to the Snapshot. The value in a Snapshot Error TLV is a UTF-8 string of arbitrary length, describing the failure reason. Note that the Snapshot Error TLV is useful for live connections, but

less so for offline data on persistent storage: as it's clear the Snapshot is incorrect or incomplete, one would likely want to replace it with a valid Snapshot instead of archiving the incorrect one.

4.3. Optional meta-data TLVs

The Snapshot Message SHOULD carry TLVs providing additional information on the BMP session being summarized. These Snapshot Information TLVs describe the BMP exporter and station involved, and the date and time the snapshot was generated. By embedding these TLVs in the offline file, a consumer of the file does not have to rely on the filename or other external data to get these types of information. All TLVs are non-indexed.

* Type = TBD4: Datetime of snapshot

Length: 8 bytes

Value: 64bit UNIX epoch, in seconds

* Type = TBD5: Exporter IP address

Length: 16 bytes

Value: IPv6 or IPv4-mapped IPv6 address

* Type = TBD6: Exporter sysName

Length: variable, non-zero, describing the number of bytes

Value: UTF-8 string

* Type = TBD7: Exporter sysDesc

Length: variable, non-zero, describing the number of bytes

Value: UTF-8 string

* Type = TBD8: Station IP address

Length: variable, non-zero, describing the number of bytes

Value: IPv6 or IPv4-mapped IPv6 address

* Type = TBD9: Station sysName

Length: variable, non-zero, describing the number of bytes

Value: UTF-8 string

* Type = TBD10: Station sysDesc

Length: variable, non-zero, describing the number of bytes

Value: UTF-8 string

5. Backwards (in)compatibility

If an implementation lacking support for Snapshots receives a Snapshot, it should ignore the Snapshot Messages and the Snapshot Id TLVs, as these are all optional. However, this reduces a Snapshot to a set of Peer Up Notification and Route Monitoring messages indistinguishable from 'normal' Peer Up Notification and Route Monitoring messages. This introduces the following problems:

- * Peer Up Notifications from a Snapshot will be interpreted as normal Peer Up Notifications, though for a peer that was already considered 'up'. This situation is not explicitly described in Section 3.3 of [RFC7854], and can be considered unexpected. Even though application of Peer Up Notifications could be considered idempotent (the state after processing the message is 'this peer is up' regardless of how often that message is received and processed), receiving the message implicitly signals the peer was down before, which is incorrect, and is possibly a source of confusion.
- * Route Monitoring messages from a Snapshot interpreted as normal Route Monitoring messages would not come as unexpected in the life cycle of a BMP session. Similarly to Peer Up Notification messages, their application is idempotent in nature as the result of processing the message is 'this route is reachable'. Note that Route Monitoring messages from a Snapshot will always be announcements, not withdrawals. However, as with Peer Up Notification messages, a normal Route Monitoring message implicitly signals 'this route was not reachable before' or 'attributes for this route have changed', which is incorrect in the case of Route Monitoring messages that were actually part of a Snapshot.

5.1. Possible workaround

- * Partial support on the consuming side: dropping messages based on Snapshot Id TLV presence

If a station implementation does not support Snapshots, it should be able to recognize Snapshot Id TLVs relatively easily. With BMPv4, the BGP Update is carried in a TLV in the Route Monitoring message, so any implementation will be able to process TLVs to a certain extent. It then only needs to be aware of the type code of the Snapshot Id TLV type: as all messages in a Snapshot will carry a TLV of this type, the implementation can simply skip/drop those messages, and preclude any of the problems described above.

* In-protocol: Introducing a flag in the Per Peer Header

Setting a flag for messages that are part of a Snapshot will allow consumers to distinguish between such messages and 'normal' Peer Up Notification and Route Monitoring messages. An additional benefit of a flag would be the enabling of optimisations on the receiving end, where messages pertaining to a Snapshot can now easily be treated differently (specific queue or thread) or discarded in case the receiver has not interest in Snapshots.

Note that this approach also requires (minimal) changes to the receiving side, namely the recognition of such a flag.

6. Third party off-wire encoding formats

While this document does define a way to facilitate stream processing, replay and, more in general, consumption of raw BMP data offline, similar benefits may be harnessed by third party off-wire formats in replay and, more in general, consumption of raw BMP data offline, similar benefits may be harnessed by third party off-wire formats in which BMP can be encapsulated into, for example MRT (Multi-Threaded Routing Toolkit) as defined by RFC 6396 [RFC6396]. As a result of that, this document does not recommend a preferred way to stream process or store BMP data offline.

7. Operational Considerations

7.1. End-of-snapshot with an Error TLV

Receiving a Snapshot Message containing a Snapshot Error TLV signals the Snapshot is incomplete or incorrect. Possible actions at this point depend on the situation, on who is consuming the Snapshot, and to which end. In an online scenario where the Snapshot was explicitly requested (see also Section 7.3), one could opt for a retry by requesting another Snapshot. If one encounters an Error TLV in a Snapshot Message part of archived, offline data, there is no way to request a new Snapshot. While such a Snapshot could be used to infer some things, it can not be used to rebuild a complete view of the network.

7.2. Snapshot Id scheme

The generation and form of the Snapshot Ids introduced in this document is left to implementations. This document does not enforce any specific approach, though at least the following points should be considered. Note that implementations are not limited to supporting only one Id scheme, but ideally support multiple schemes via local configuration.

7.2.1. Global uniqueness

In deployments where information is received from multiple BMP vantage points, unique Snapshot Ids might prove handy or even crucial in order to distinguish Snapshot A originally sent by BMP exporter X, from Snapshot B sent by exporter Y. If all exporting processes rely on an algorithm producing globally unique identifiers, e.g. UUID version 4, they all can send out Snapshots without possibly using an identical Snapshot Id generated by another exporter.

7.2.2. Increasing identifiers

Generating (linearly) increasing identifiers enable the BMP station to order Snapshots, and, to spot any missing Snapshots. Furthermore, in a (long running) BMP session where the exporter generates Snapshots, the Snapshot Id doubles as a counter signalling how many Snapshots have been sent so far. Note that some of these can be deduced via other means: ordering of Snapshots can be done based on the Timestamp TLV (TBD3), and the number of sent Snapshots could be included in the Stats Report message (Section 4.8 of [RFC7854]).

7.3. Requesting/triggering snapshots

BMP being a unidirectional protocol from exporter to station means there is no way for a station to request or trigger the creation of a Snapshot in-protocol. This document does not define or advocate for any specific out-of-band way to do such triggering. But as any implementation of Snapshots requires at least one way to trigger their creation, some possible approaches are briefly discussed.

7.3.1. Timer-based

An exporter could be configured to generate and emit Snapshots on a regular interval, without a request from the station. Depending on the timing parameters and the size of the Snapshots (i.e, the amount of sessions and routes), this might cause high loads on either or both sides of the BMP session. This approach should only be considered if the station needs a full view on a regular basis, which is not necessary in typical deployments.

A timer-based approach could be a suitable approach for a station generating Snapshots and writing them to persistent storage, e.g. doing time-binned historical archiving.

7.3.2. Manual triggers

In situations where a Snapshot is only used to recover from a broken state at a station (either directly connected to the exporter or a station further downstream), a dedicated command on the exporter to generate and emit a Snapshot could be used.

Note that, for most situations where the station is directly connected to the exporter, a re-establishing the TCP connection for the BMP stream might be a simpler and perhaps even cheaper alternative.

7.3.3. Out-of-band request via other protocols

For more complex setups where e.g. the BMP messages continue on a messages bus, the generation and sending of Snapshots could be requested via that message bus or another protocol/API. Such requests could include parameters to ask for a Snapshot containing only a subset of all the data, e.g. only a certain address family, or only a certain RIB view. The concept of Snapshots as described in this document allows for such subsets, but such an out-of-band protocol and its parameters are out of scope.

8. Discussion

8.1. No EoRs in Snapshots

This document describes the logical concept of a Snapshot as a collection of Peer Up Notifications and Route Monitoring messages, preceded and followed by a Snapshot Message. Comparing this to the initial synchronisation at the start of a BMP session, there is a discrepancy, as the Snapshot does not include EoRs to signal a RIB (view) has been fully sent. An EoR in this context is a RouteMonitoring message for a certain RIB view (e.g. Adj-RIB-In Post-policy, as defined by the flags in the Per Peer Header), containing a BGP Update for a certain address family with no announcements and no withdrawals.

With the Snapshot Message signalling end-of-snapshot, the EoRs are not a necessity. One reason for inclusion would be consistency, though the current state of EoRs seems to be inconsistent between implementations, and the text describing them leaves room for interpretation. Including EoRs in a Snapshot perhaps only adds to the confusion, and therefore leaving them out could be preferable.

9. Security Considerations

It is not believed that this document adds any additional security considerations.

10. IANA Considerations

IANA is asked to allocate a new Snapshot Message type in the BMP Message Types registry with value TBD1. IANA is also asked to create a registry within the BMP group, named "BMP Snapshot Message TLVs".

Registration procedures for this new registry are:

// Note that these have been adapted to the proposed ranges as
// described in [I-D.ietf-grow-bmp-tlv] version -19.

Range	Registration Procedures
0-16383	Standards Action
16384-32767	First Come, First Served
65535	Reserved

Table 2

Initial values for this registry are:

// Update this table once we have converged on Section 4.3.

Type	Description	Reference
TBD2	Snapshot Id	this document
TBD3	Snapshot Error	this document

Table 3

IANA is also asked to allocate codepoints for the Snapshot Id TLVs in the BMP Peer Up Message TLV registry and the BMP Route Monitoring TLV registry. Considering the Snapshot Id TLV then appears in three registries, ideally the same codepoint is allocated in all registries. However, this will not be possible without introducing gaps in the registries, which might be undesirable.

11. References

11.1. Normative References

[I-D.ietf-grow-bmp-rel]

Lucente, P. and C. Cardona, "Logging of routing events in BGP Monitoring Protocol (BMP)", Work in Progress, Internet-Draft, draft-ietf-grow-bmp-rel-05, 2 March 2026, <<https://datatracker.ietf.org/doc/html/draft-ietf-grow-bmp-rel-05>>.

[I-D.ietf-grow-bmp-tlv]

Lucente, P., Gu, Y., Younsi, M., and P. Francois, "BMP v4: Extended TLV Support for BGP Monitoring Protocol (BMP)", Work in Progress, Internet-Draft, draft-ietf-grow-bmp-tlv-20, 2 March 2026, <<https://datatracker.ietf.org/doc/html/draft-ietf-grow-bmp-tlv-20>>.

[I-D.petrie-grow-mrt-bmp]

Petrie, C. and L. Hendriks, "Storing BMP messages in MRT Format", Work in Progress, Internet-Draft, draft-petrie-grow-mrt-bmp-01, 20 October 2025, <<https://datatracker.ietf.org/doc/html/draft-petrie-grow-mrt-bmp-01>>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.

[RFC6396] Blunk, L., Karir, M., and C. Labovitz, "Multi-Threaded Routing Toolkit (MRT) Routing Information Export Format", RFC 6396, DOI 10.17487/RFC6396, October 2011, <<https://www.rfc-editor.org/info/rfc6396>>.

[RFC7854] Scudder, J., Ed., Fernando, R., and S. Stuart, "BGP Monitoring Protocol (BMP)", RFC 7854, DOI 10.17487/RFC7854, June 2016, <<https://www.rfc-editor.org/info/rfc7854>>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

11.2. Informative References

[RFC2918] Chen, E., "Route Refresh Capability for BGP-4", RFC 2918, DOI 10.17487/RFC2918, September 2000, <<https://www.rfc-editor.org/info/rfc2918>>.

Acknowledgements

TBD

Authors' Addresses

Luuk Hendriks
NLnet Labs
Science Park 400
1098 XH Amsterdam
Netherlands
Email: luuk@nlnetlabs.nl

Paolo Lucente
NTT
Veemweg 23
3771 MT Barneveld
Netherlands
Email: paolo@ntt.net

Camilo Cardona
NTT
164-168, Carrer de Numancia
08029 Barcelona
Spain
Email: camilo@ntt.net

Colin Petrie
NTT
Veemweg 23
3771 MT Barneveld
Netherlands
Email: colin@ntt.net