

NMOP
Internet-Draft
Intended status: Standards Track
Expires: 17 August 2026

TG. Graf
Swisscom
PL. Lucente
NTT
LR. Rodoni
Swisscom
MY. Younsi
INSA-Lyon
13 February 2026

BMP YANG Model for Network Telemetry Messages
draft-netana-nmop-message-broker-bmp-telemetry-msg-02

Abstract

This document defines an BGP Monitoring Protocol (BMP) message schema extension in YANG to be used at data collection to transform Network Telemetry messages into external systems such as Message Brokers.

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 17 August 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	2
2. Conventions and Definitions	3
2.1. Terminology	3
3. Relationship to other documents	5
3.1. Relationship to draft-ietf-nmop-message-broker-telemetry-message	5
3.2. Relationship to draft-ietf-idr-bgp-model	5
3.3. Relationship to draft-ietf-grow-bmp-yang	6
4. BMP YANG Modules	6
4.1. BMP BGP RIB YANG Submodules	48
5. YANG Message Broker BMP Indexing and Topic Naming	65
5.1. BMP Message Keys and Indexes	65
5.2. Message Broker BMP Topic Naming	66
5.2.1. YANG Message Broker Producer	66
5.2.2. YANG Message Broker Consumer	67
6. IANA Considerations	67
7. Security Considerations	68
8. References	69
8.1. Normative References	69
8.2. Informative References	71
Acknowledgements	72
Contributors	72
Authors' Addresses	72

1. Introduction

Nowadays network operators are using BGP Monitoring Protocol (BMP) [RFC7854] to monitor the BGP peerings and RIB's. With [I-D.ietf-nmop-message-broker-telemetry-message] a machine and human readable extensible Network Telemetry message schema in YANG [RFC7950] has been defined.

Network operators organize their data in a Data Mesh [Deh22] where a Message Broker such as Apache Kafka [Kaf11] or Apache Pulsar [Pul16] facilitates the exchange of messages among data processing components.

With YANG Message Keys for Message Broker Integration [I-D.netana-nmop-yang-message-broker-message-key] how Dimensional YANG data can be indexed and addressed in Message Brokers are being described.

This document describes how [I-D.ietf-nmop-message-broker-telemetry-message] defined Network Telemetry message is being extended to support BMP data transformation to YANG at the Network Telemetry data collector and being indexed and addressed in Message Broker based on [I-D.netana-nmop-yang-message-broker-message-key] principles at the YANG Message Broker Producer.

2. Conventions and Definitions

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 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

2.1. Terminology

The following terms are used as defined in [I-D.ietf-nmop-terminology]:

- * Network Telemetry
- * Network Analytics
- * Value
- * State
- * Change

The following terms are used as defined in [I-D.ietf-nmop-yang-message-broker-integration]:

- * Message Broker
- * YANG Message Broker Producer
- * YANG Message Broker Consumer

The following terms are used as defined in Apache Kafka [Kaf11] and Apache Pulsar [Pul16] Message Broker:

- * **Subject:** Is used for Messages within a Topic sharing the same schema tree and also to identify a unique schema tree within a schema registry.
- * **Topic:** A communication channel for publishing and subscribing messages with one or more subjects and partitions.
- * **Topic Compaction:** The act of compressing messages in a topic to the latest state. As used with Apache Pulsar. Apache Kafka uses the term Log Compaction with identical meaning.
- * **Partition:** Messages in a topic are spread over hash buckets where a hash bucket refers to a partition being stored on one message broker node. Message ordering is guaranteed within a partition.
- * **Message:** A piece of structured data sent between data processing components to facilitate communication in a distributed system
- * **Message Key:** Metadata associated with a message to facilitate deterministic hash bucketing.

The following terms are used as defined in [RFC7854]:

- * BMP
- * BMP Message
- * Adj-RIB-In
- * Pre-policy Adj-RIB-In
- * Post-Policy Adj-RIB-In

The following terms are used as defined in [RFC8671]:

- * Adj-RIB-Out
- * Pre-policy Adj-RIB-Out
- * Post-policy Adj-RIB-Out

The following terms are used as defined in [RFC9069]:

- * BGP Instance
- * Loc-RIB

This document defines the following term:

- * BMP Index: A unique identifier created depending on BMP Message Type from multiple BMP and BGP data dimensions. The BMP Index is used to generate the Message Key. See Section 5.1.

3. Relationship to other documents

3.1. Relationship to draft-ietf-nmop-message-broker-telemetry-message

[I-D.ietf-nmop-message-broker-telemetry-message] defines with 'ietf-telemetry-message' YANG module a base Network Telemetry Message schema being used between Section 4.6 of Message Broker Producer [I-D.ietf-nmop-yang-message-broker-integration] and Section 4.7 of Message Broker Consumer [I-D.ietf-nmop-yang-message-broker-integration] and with 'ietf-yang-push-telemetry-message' YANG module an extension that supports YANG-Push specific subscription metadata.

This document defines one BMP YANG module Figure 2, and 3 BGP RIB submodules Section 4.1 which are being used in the anydata "payload" [I-D.ietf-nmop-message-broker-telemetry-message] defined in 'ietf-telemetry-message' YANG module.

3.2. Relationship to draft-ietf-idr-bgp-model

[I-D.ietf-idr-bgp-model] defines with 'ietf-bgp' a YANG module and with 'iana-bgp-community-types', 'iana-bgp-notification', 'iana-bgp-rib-types', 'iana-bgp-types', 'ietf-bgp-capabilities', 'ietf-bgp-common', 'ietf-bgp-common-multiprotocol', 'ietf-bgp-common-structure', 'ietf-bgp-neighbor', 'ietf-bgp-policy', 'ietf-bgp-rib', 'ietf-bgp-rib-attributes' and 'ietf-bgp-rib-tables' submodules.

This document is using the content of the 'rib' YANG grouping defined in 'ietf-bgp-rib' submodule and renames the YANG grouping to 'rib-entry' and restructures the data taxonomy to allow a single BGP RIB entry to be serialized in a Telemetry Message on Message Broker in the 'ietf-bmp-bgp-rib-entry' submodule. The 'clear-routes' YANG grouping used in the 'rib' YANG grouping has been removed since a Telemetry Message only covers operational metrics.

Since 'ietf-bgp-rib' uses groupings from 'ietf-bgp-rib-attributes' and 'ietf-bgp-rib-tables' and both 'ietf-bgp-rib-attributes' and 'ietf-bgp-rib-tables' YANG submodules Section 7.2.2 of belongs to [RFC7950] 'ietf-bgp' main module, the YANG submodules have been renamed to 'ietf-bmp-bgp-rib-attributes' and 'ietf-bmp-bgp-rib-tables'.

If 'ietf-bgp-rib-attributes' and 'ietf-bgp-rib-tables' would not be 'ietf-bgp' submodules, they could be reused as is with 'ietf-bmp-bgp-rib-entry'. Since this would imply than a normative document dependency a reasonable timeline needs to be coordinated. Once these clarifications have been concluded this section remains. The [I-D.ietf-idr-bgp-model] authors are being honored under Appendix "Contributors" for their contribution on the BGP YANG modelling this document uses.

3.3. Relationship to draft-ietf-grow-bmp-yang

[I-D.ietf-grow-bmp-yang] defines with 'ietf-bmp' YANG module the configuration and monitoring of BMP. While there is no direct relationship between this document and its defined YANG modules, it is worth mentioning that the with [I-D.ietf-grow-bmp-yang] subscribed BMP data is being transformed at BMP monitoring stations to Message Brokers as described in this document.

4. BMP YANG Modules

The BMP YANG module for Network Telemetry messages defines the BMP subscription metadata and depending on BMP Message type different metadata structures.

Where the 3 BGP RIB YANG submodules extend the BMP YANG module with BGP RIB structures for BMP route-monitoring and route-mirroring Message types.

module: ietf-bmp-telemetry-message

```

structure message:
  +-- version?                uint8
  +-- (message-type)?
    +--:(route-monitoring)
      +-- route-monitoring
        +-- peer-type?        peer-type
        +-- peer-flags?       uint8
        +-- peer-distinguisher? rt-types:route-distinguisher
        +-- peer-address?     inet:ip-address
        +-- peer-as?          uint32
        +-- timestamp?        yang:date-and-time
        +-- afi-safi-type     identityref
        +-- rib-entry
          +-- (afi-safi)
            +--:(ipv4-unicast)
              +-- ipv4-unicast
                +-- (rib-type)
                  +--:(loc-rib)

```

```

+-- loc-rib
  +-- route
    +-- prefix
    |   inet:ipv4-prefix
    +-- origin?
    |   union
    +-- path-id?
    |   uint32
    +-- attributes
    |   +-- origin?
    |   |   bt:bgp-origin-attr-type
    +-- as-path
    |   +-- segment*
    |   |   +-- type?
    |   |   |   identityref
    |   |   +-- member*
    |   |       inet:as-number
    +-- next-hop?
    |   inet:ip-address
    +-- link-local-next-hop?
    |   inet:ipv6-address
    +-- med?
    |   uint32
    +-- local-pref?
    |   uint32
    +-- as4-path
    |   +-- segment*
    |   |   +-- type?
    |   |   |   identityref
    |   |   +-- member*
    |   |       inet:as-number
    +-- aggregator
    |   +-- as?
    |   |   inet:as-number
    |   +-- identifier?
    |       yang:dotted-quad
    +-- aggregator4
    |   +-- as4?
    |   |   inet:as-number
    |   +-- identifier?
    |       yang:dotted-quad
    +-- atomic-aggregate?
    |   boolean
    +-- originator-id?
    |   yang:dotted-quad
    +-- cluster-list*
    |   yang:dotted-quad
    +-- aigp-metric?

```

```

|         uint64
+-- community*
|   union
+-- ext-community*
|   bct:bgp-ext-community-type
+-- ext-community-raw*
|   string
+-- ipv6-ext-community*
|   bct:bgp-ipv6-ext-community-type
+-- ipv6-ext-community-raw*
|   string
+-- large-community*
|   bct:bgp-large-community-type
+-- last-modified?
|   yang:timeticks
+-- eligible-route?
|   boolean
+-- ineligible-reason?
|   identityref
+-- unknown-attributes
|   +-- unknown-attribute*
|       [attr-type]
|       +-- attr-type      uint8
|       +-- optional?     boolean
|       +-- transitive?   boolean
|       +-- partial?      boolean
|       +-- extended?     boolean
|       +-- attr-len?     uint16
|       +-- attr-value?   binary
+-- reject-reason?
|   union
+--:(adj-rib-in-pre)
+-- adj-rib-in-pre
+-- route
+-- prefix?
|   inet:ipv4-prefix
+-- path-id?
|   uint32
+-- attributes
|   +-- origin?
|       |   bt:bgp-origin-attr-type
+-- as-path
|   +-- segment*
|       +-- type?
|           |   identityref
|       +-- member*
|           |   inet:as-number
+-- next-hop?

```



```

|         inet:ip-address
+-- link-local-next-hop?
|         inet:ipv6-address
+-- med?
|         uint32
+-- local-pref?
|         uint32
+-- as4-path
|   +-- segment*
|     +-- type?
|       |         identityref
|     +-- member*
|       |         inet:as-number
+-- aggregator
|   +-- as?
|     |         inet:as-number
|   +-- identifier?
|     |         yang:dotted-quad
+-- aggregator4
|   +-- as4?
|     |         inet:as-number
|   +-- identifier?
|     |         yang:dotted-quad
+-- atomic-aggregate?
|   boolean
+-- originator-id?
|   yang:dotted-quad
+-- cluster-list*
|   yang:dotted-quad
+-- aigp-metric?
|   uint64
+-- community*
|   union
+-- ext-community*
|   bct:bgp-ext-community-type
+-- ext-community-raw*
|   string
+-- ipv6-ext-community*
|   bct:bgp-ipv6-ext-community-type
+-- ipv6-ext-community-raw*
|   string
+-- large-community*
|   bct:bgp-large-community-type
+-- last-modified?
|   yang:timeticks
+-- eligible-route?
|   boolean
+-- ineligible-reason?

```

```

|         identityref
+-- unknown-attributes
|   +-- unknown-attribute*
|       [attr-type]
|         +-- attr-type      uint8
|         +-- optional?     boolean
|         +-- transitive?   boolean
|         +-- partial?      boolean
|         +-- extended?     boolean
|         +-- attr-len?     uint16
|         +-- attr-value?   binary
+-- reject-reason?
|   union
+--:(adj-rib-in-post)
+-- adj-rib-in-post
+-- route
+-- prefix?
|   inet:ipv4-prefix
+-- path-id?
|   uint32
+-- attributes
+-- origin?
|   bt:bgp-origin-attr-type
+-- as-path
+-- segment*
+-- type?
|   identityref
+-- member*
|   inet:as-number
+-- next-hop?
|   inet:ip-address
+-- link-local-next-hop?
|   inet:ipv6-address
+-- med?
|   uint32
+-- local-pref?
|   uint32
+-- as4-path
+-- segment*
+-- type?
|   identityref
+-- member*
|   inet:as-number
+-- aggregator
+-- as?
|   inet:as-number
+-- identifier?
|   yang:dotted-quad

```

```

| | | aggregator4
| | |   +-- as4?
| | |     | inet:as-number
| | |   +-- identifier?
| | |     | yang:dotted-quad
+-- atomic-aggregate?
|   boolean
+-- originator-id?
|   yang:dotted-quad
+-- cluster-list*
|   yang:dotted-quad
+-- aigp-metric?
|   uint64
+-- community*
|   union
+-- ext-community*
|   bct:bgp-ext-community-type
+-- ext-community-row*
|   string
+-- ipv6-ext-community*
|   bct:bgp-ipv6-ext-community-type
+-- ipv6-ext-community-row*
|   string
+-- large-community*
|   bct:bgp-large-community-type
+-- last-modified?
|   yang:timeticks
+-- eligible-route?
|   boolean
+-- ineligible-reason?
|   identityref
+-- best-path?
|   boolean
+-- unknown-attributes
|   +-- unknown-attribute*
|       [attr-type]
|           +-- attr-type      uint8
|           +-- optional?    boolean
|           +-- transitive?  boolean
|           +-- partial?    boolean
|           +-- extended?   boolean
|           +-- attr-len?   uint16
|           +-- attr-value?  binary
+-- reject-reason?
|   union
+---:(adj-rib-out-pre)
|   +-- adj-rib-out-pre
|       +-- route

```

```

+-- prefix?
|   inet:ipv4-prefix
+-- path-id?
|   uint32
+-- attributes
|   +-- origin?
|   |   bt:bgp-origin-attr-type
|   +-- as-path
|   |   +-- segment*
|   |   |   +-- type?
|   |   |   |   identityref
|   |   |   +-- member*
|   |   |       inet:as-number
|   +-- next-hop?
|   |   inet:ip-address
|   +-- link-local-next-hop?
|   |   inet:ipv6-address
|   +-- med?
|   |   uint32
|   +-- local-pref?
|   |   uint32
|   +-- as4-path
|   |   +-- segment*
|   |   |   +-- type?
|   |   |   |   identityref
|   |   |   +-- member*
|   |   |       inet:as-number
|   +-- aggregator
|   |   +-- as?
|   |   |   inet:as-number
|   |   +-- identifier?
|   |       yang:dotted-quad
|   +-- aggregator4
|   |   +-- as4?
|   |   |   inet:as-number
|   |   +-- identifier?
|   |       yang:dotted-quad
|   +-- atomic-aggregate?
|   |   boolean
|   +-- originator-id?
|   |   yang:dotted-quad
|   +-- cluster-list*
|   |   yang:dotted-quad
|   +-- aigp-metric?
|   |   uint64
+-- community*
|   union
+-- ext-community*

```

```

|         bct:bgp-ext-community-type
+-- ext-community-raw*
|         string
+-- ipv6-ext-community*
|         bct:bgp-ipv6-ext-community-type
+-- ipv6-ext-community-raw*
|         string
+-- large-community*
|         bct:bgp-large-community-type
+-- last-modified?
|         yang:timeticks
+-- eligible-route?
|         boolean
+-- ineligible-reason?
|         identityref
+-- unknown-attributes
|   +-- unknown-attribute*
|     [attr-type]
|     +-- attr-type      uint8
|     +-- optional?     boolean
|     +-- transitive?    boolean
|     +-- partial?       boolean
|     +-- extended?      boolean
|     +-- attr-len?      uint16
|     +-- attr-value?    binary
+-- reject-reason?
|         union
+--:(adj-rib-out-post)
+-- adj-rib-out-post
+-- route
+-- prefix?
|   inet:ipv4-prefix
+-- path-id?
|   uint32
+-- attributes
|   +-- origin?
|     |   bt:bgp-origin-attr-type
+-- as-path
|   +-- segment*
|     +-- type?
|       |   identityref
|     +-- member*
|       |   inet:as-number
+-- next-hop?
|   inet:ip-address
+-- link-local-next-hop?
|   inet:ipv6-address
+-- med?

```

```

|         uint32
+-- local-pref?
|         uint32
+-- as4-path
|   +-- segment*
|   |   +-- type?
|   |   |   identityref
|   |   +-- member*
|   |       inet:as-number
+-- aggregator
|   +-- as?
|   |   inet:as-number
|   +-- identifier?
|       yang:dotted-quad
+-- aggregator4
|   +-- as4?
|   |   inet:as-number
|   +-- identifier?
|       yang:dotted-quad
+-- atomic-aggregate?
|   boolean
+-- originator-id?
|   yang:dotted-quad
+-- cluster-list*
|   yang:dotted-quad
+-- aigp-metric?
|   uint64
+-- community*
|   union
+-- ext-community*
|   bct:bgp-ext-community-type
+-- ext-community-row*
|   string
+-- ipv6-ext-community*
|   bct:bgp-ipv6-ext-community-type
+-- ipv6-ext-community-row*
|   string
+-- large-community*
|   bct:bgp-large-community-type
+-- last-modified?
|   yang:timeticks
+-- eligible-route?
|   boolean
+-- ineligible-reason?
|   identityref
+-- unknown-attributes
|   +-- unknown-attribute*
|       [attr-type]

```

```

+--- attr-type          uint8
+--- optional?         boolean
+--- transitive?       boolean
+--- partial?          boolean
+--- extended?         boolean
+--- attr-len?         uint16
+--- attr-value?       binary
+--- reject-reason?    union
+---:(ipv6-unicast)
+--- ipv6-unicast
+--- (rib-type)
+---:(loc-rib)
+--- loc-rib
+--- route
+--- prefix
+--- | inet:ipv6-prefix
+--- origin?
+--- | union
+--- path-id?
+--- | uint32
+--- attributes
+--- | origin?
+--- | | bt:bgp-origin-attr-type
+--- as-path
+--- | +--- segment*
+--- | | +--- type?
+--- | | | identityref
+--- | | +--- member*
+--- | | | inet:as-number
+--- next-hop?
+--- | inet:ip-address
+--- link-local-next-hop?
+--- | inet:ipv6-address
+--- med?
+--- | uint32
+--- local-pref?
+--- | uint32
+--- as4-path
+--- | +--- segment*
+--- | | +--- type?
+--- | | | identityref
+--- | | +--- member*
+--- | | | inet:as-number
+--- aggregator
+--- | +--- as?
+--- | | inet:as-number
+--- | +--- identifier?

```

```

|         yang:dotted-quad
+-- aggregator4
|   +-- as4?
|   |     inet:as-number
|   +-- identifier?
|       yang:dotted-quad
+-- atomic-aggregate?
|   boolean
+-- originator-id?
|   yang:dotted-quad
+-- cluster-list*
|   yang:dotted-quad
+-- aigp-metric?
|   uint64
+-- community*
|   union
+-- ext-community*
|   bct:bgp-ext-community-type
+-- ext-community-row*
|   string
+-- ipv6-ext-community*
|   bct:bgp-ipv6-ext-community-type
+-- ipv6-ext-community-row*
|   string
+-- large-community*
|   bct:bgp-large-community-type
+-- last-modified?
|   yang:timeticks
+-- eligible-route?
|   boolean
+-- ineligible-reason?
|   identityref
+-- unknown-attributes
|   +-- unknown-attribute*
|       [attr-type]
|       +-- attr-type      uint8
|       +-- optional?     boolean
|       +-- transitive?   boolean
|       +-- partial?      boolean
|       +-- extended?     boolean
|       +-- attr-len?     uint16
|       +-- attr-value?   binary
+-- reject-reason?
|   union
+--:(adj-rib-in-pre)
|   +-- adj-rib-in-pre
|       +-- route
|       +-- prefix?

```



```

|         inet:ipv6-prefix
+-- path-id?
|         uint32
+-- attributes
|   +-- origin?
|   |     bt:bgp-origin-attr-type
+-- as-path
|   +-- segment*
|   |     +-- type?
|   |     |     identityref
|   |     +-- member*
|   |         inet:as-number
+-- next-hop?
|   inet:ip-address
+-- link-local-next-hop?
|   inet:ipv6-address
+-- med?
|   uint32
+-- local-pref?
|   uint32
+-- as4-path
|   +-- segment*
|   |     +-- type?
|   |     |     identityref
|   |     +-- member*
|   |         inet:as-number
+-- aggregator
|   +-- as?
|   |     inet:as-number
|   +-- identifier?
|       yang:dotted-quad
+-- aggregator4
|   +-- as4?
|   |     inet:as-number
|   +-- identifier?
|       yang:dotted-quad
+-- atomic-aggregate?
|   boolean
+-- originator-id?
|   yang:dotted-quad
+-- cluster-list*
|   yang:dotted-quad
+-- aigp-metric?
|   uint64
+-- community*
|   union
+-- ext-community*
|   bct:bgp-ext-community-type

```

```

    +-- ext-community-raw*
    |   string
    +-- ipv6-ext-community*
    |   bct:bgp-ipv6-ext-community-type
    +-- ipv6-ext-community-raw*
    |   string
    +-- large-community*
    |   bct:bgp-large-community-type
    +-- last-modified?
    |   yang:timeticks
    +-- eligible-route?
    |   boolean
    +-- ineligible-reason?
    |   identityref
    +-- unknown-attributes
    |   +-- unknown-attribute*
    |   |   [attr-type]
    |   |   +-- attr-type      uint8
    |   |   +-- optional?    boolean
    |   |   +-- transitive?  boolean
    |   |   +-- partial?     boolean
    |   |   +-- extended?    boolean
    |   |   +-- attr-len?    uint16
    |   |   +-- attr-value?  binary
    |   +-- reject-reason?
    |   |   union
    +--:(adj-rib-in-post)
    +-- adj-rib-in-post
    |   +-- route
    |   |   +-- prefix?
    |   |   |   inet:ipv6-prefix
    |   |   +-- path-id?
    |   |   |   uint32
    |   |   +-- attributes
    |   |   |   +-- origin?
    |   |   |   |   bt:bgp-origin-attr-type
    |   |   |   +-- as-path
    |   |   |   |   +-- segment*
    |   |   |   |   |   +-- type?
    |   |   |   |   |   |   identityref
    |   |   |   |   |   +-- member*
    |   |   |   |   |   |   inet:as-number
    |   |   |   +-- next-hop?
    |   |   |   |   inet:ip-address
    |   |   |   +-- link-local-next-hop?
    |   |   |   |   inet:ipv6-address
    |   |   |   +-- med?
    |   |   |   |   uint32

```

```

+-- local-pref?
|   uint32
+-- as4-path
|   +-- segment*
|   |   +-- type?
|   |   |   identityref
|   |   +-- member*
|   |       inet:as-number
+-- aggregator
|   +-- as?
|   |   inet:as-number
|   +-- identifier?
|       yang:dotted-quad
+-- aggregator4
|   +-- as4?
|   |   inet:as-number
|   +-- identifier?
|       yang:dotted-quad
+-- atomic-aggregate?
|   boolean
+-- originator-id?
|   yang:dotted-quad
+-- cluster-list*
|   yang:dotted-quad
+-- aigp-metric?
|   uint64
+-- community*
|   union
+-- ext-community*
|   bct:bgp-ext-community-type
+-- ext-community-row*
|   string
+-- ipv6-ext-community*
|   bct:bgp-ipv6-ext-community-type
+-- ipv6-ext-community-row*
|   string
+-- large-community*
|   bct:bgp-large-community-type
+-- last-modified?
|   yang:timeticks
+-- eligible-route?
|   boolean
+-- ineligible-reason?
|   identityref
+-- best-path?
|   boolean
+-- unknown-attributes
|   +-- unknown-attribute*

```

```

        [attr-type]
        +-- attr-type      uint8
        +-- optional?     boolean
        +-- transitive?   boolean
        +-- partial?      boolean
        +-- extended?     boolean
        +-- attr-len?     uint16
        +-- attr-value?   binary
        +-- reject-reason?
            union
+--:(adj-rib-out-pre)
+-- adj-rib-out-pre
+-- route
+-- prefix?
+-- | inet:ipv6-prefix
+-- path-id?
+-- | uint32
+-- attributes
+-- | origin?
+-- | | bt:bgp-origin-attr-type
+-- as-path
+-- | +-- segment*
+-- | | +-- type?
+-- | | | identityref
+-- | | +-- member*
+-- | | | inet:as-number
+-- next-hop?
+-- | inet:ip-address
+-- link-local-next-hop?
+-- | inet:ipv6-address
+-- med?
+-- | uint32
+-- local-pref?
+-- | uint32
+-- as4-path
+-- | +-- segment*
+-- | | +-- type?
+-- | | | identityref
+-- | | +-- member*
+-- | | | inet:as-number
+-- aggregator
+-- | +-- as?
+-- | | inet:as-number
+-- | +-- identifier?
+-- | | yang:dotted-quad
+-- aggregator4
+-- | +-- as4?
+-- | | inet:as-number

```

```

| | | +-- identifier?
| | | | yang:dotted-quad
| | | +-- atomic-aggregate?
| | | | boolean
| | | +-- originator-id?
| | | | yang:dotted-quad
| | | +-- cluster-list*
| | | | yang:dotted-quad
| | | +-- aigp-metric?
| | | | uint64
| | +-- community*
| | | union
| | +-- ext-community*
| | | bct:bgp-ext-community-type
| | +-- ext-community-raw*
| | | string
| | +-- ipv6-ext-community*
| | | bct:bgp-ipv6-ext-community-type
| | +-- ipv6-ext-community-raw*
| | | string
| | +-- large-community*
| | | bct:bgp-large-community-type
| | +-- last-modified?
| | | yang:timeticks
| | +-- eligible-route?
| | | boolean
| | +-- ineligible-reason?
| | | identityref
| | +-- unknown-attributes
| | | +-- unknown-attribute*
| | | | [attr-type]
| | | | +-- attr-type      uint8
| | | | +-- optional?    boolean
| | | | +-- transitive?  boolean
| | | | +-- partial?     boolean
| | | | +-- extended?    boolean
| | | | +-- attr-len?    uint16
| | | | +-- attr-value?  binary
| | +-- reject-reason?
| | | union
+--:(adj-rib-out-post)
+-- adj-rib-out-post
+-- route
+-- prefix?
| | inet:ipv6-prefix
+-- path-id?
| | uint32
+-- attributes

```

```

+-- origin?
|   bt:bgp-origin-attr-type
+-- as-path
|   +-- segment*
|   |   +-- type?
|   |   |   identityref
|   |   +-- member*
|   |       inet:as-number
+-- next-hop?
|   inet:ip-address
+-- link-local-next-hop?
|   inet:ipv6-address
+-- med?
|   uint32
+-- local-pref?
|   uint32
+-- as4-path
|   +-- segment*
|   |   +-- type?
|   |   |   identityref
|   |   +-- member*
|   |       inet:as-number
+-- aggregator
|   +-- as?
|   |   inet:as-number
|   +-- identifier?
|       yang:dotted-quad
+-- aggregator4
|   +-- as4?
|   |   inet:as-number
|   +-- identifier?
|       yang:dotted-quad
+-- atomic-aggregate?
|   boolean
+-- originator-id?
|   yang:dotted-quad
+-- cluster-list*
|   yang:dotted-quad
+-- aigp-metric?
|   uint64
+-- community*
|   union
+-- ext-community*
|   bct:bgp-ext-community-type
+-- ext-community-raw*
|   string
+-- ipv6-ext-community*
|   bct:bgp-ipv6-ext-community-type

```

```

+-- ipv6-ext-community-row*
|   string
+-- large-community*
|   bct:bgp-large-community-type
+-- last-modified?
|   yang:timeticks
+-- eligible-route?
|   boolean
+-- ineligible-reason?
|   identityref
+-- unknown-attributes
|   +-- unknown-attribute*
|       [attr-type]
|       +-- attr-type      uint8
|       +-- optional?      boolean
|       +-- transitive?    boolean
|       +-- partial?       boolean
|       +-- extended?      boolean
|       +-- attr-len?      uint16
|       +-- attr-value?    binary
+-- reject-reason?
    union

+--:(statistics-report)
+-- statistics-report
+-- peer-type?          peer-type
+-- peer-flags?         uint8
+-- peer-distinguisher? rt-types:route-distinguisher
+-- peer-address?       inet:ip-address
+-- peer-as?            uint32
+-- timestamp?          yang:date-and-time
+-- statistics-type?    string
+-- statistics-data?    uint64

+--:(peer-down-notification)
+-- peer-down-notification
+-- peer-type?          peer-type
+-- peer-flags?         uint8
+-- peer-distinguisher? rt-types:route-distinguisher
+-- peer-address?       inet:ip-address
+-- peer-as?            uint32
+-- timestamp?          yang:date-and-time

+--:(peer-up-notification)
+-- peer-up-notification
+-- peer-type?          peer-type
+-- peer-flags?         uint8
+-- peer-distinguisher? rt-types:route-distinguisher
+-- peer-address?       inet:ip-address
+-- peer-as?            uint32
+-- timestamp?          yang:date-and-time

```

```

+---:(initiation-message)
|   +-- initiation-message
|       +-- string?      string
|       +-- sys-descr?   string
|       +-- sys-name?    string
+---:(termination-message)
|   +-- termination-message
|       +-- string?      string
|       +-- reason?      enumeration
+---:(route-mirroring-message)
|   +-- route-mirroring-message
|       +-- peer-type?      peer-type
|       +-- peer-flags?     uint8
|       +-- peer-distinguisher? rt-types:route-distinguisher
|       +-- peer-address?   inet:ip-address
|       +-- peer-as?        uint32
|       +-- timestamp?      yang:date-and-time
|       +-- afi-safi-type   identityref
|       +-- rib-entry
|           +-- (afi-safi)
|               +---:(ipv4-unicast)
|                   +-- ipv4-unicast
|                       +-- (rib-type)
|                           +---:(loc-rib)
|                               +-- loc-rib
|                                   +-- route
|                                       +-- prefix
|                                           |
|                                           |   inet:ipv4-prefix
|                                       +-- origin?
|                                           |
|                                           |   union
|                                       +-- path-id?
|                                           |
|                                           |   uint32
|                                       +-- attributes
|                                           +-- origin?
|                                               |
|                                               |   bt:bgp-origin-attr-type
|                                       +-- as-path
|                                           +-- segment*
|                                               +-- type?
|                                                   |
|                                                   |   identityref
|                                           +-- member*
|                                               |
|                                               |   inet:as-number
|                                       +-- next-hop?
|                                           |
|                                           |   inet:ip-address
|                                       +-- link-local-next-hop?
|                                           |
|                                           |   inet:ipv6-address
|                                       +-- med?
|                                           |
|                                           |   uint32
|                                       +-- local-pref?

```



```

|         uint32
+-- as4-path
|   +-- segment*
|       +-- type?
|           |
|           identityref
|       +-- member*
|           inet:as-number
+-- aggregator
|   +-- as?
|       |
|       inet:as-number
|   +-- identifier?
|       yang:dotted-quad
+-- aggregator4
|   +-- as4?
|       |
|       inet:as-number
|   +-- identifier?
|       yang:dotted-quad
+-- atomic-aggregate?
|   boolean
+-- originator-id?
|   yang:dotted-quad
+-- cluster-list*
|   yang:dotted-quad
+-- aigp-metric?
|   uint64
+-- community*
|   union
+-- ext-community*
|   bct:bgp-ext-community-type
+-- ext-community-raw*
|   string
+-- ipv6-ext-community*
|   bct:bgp-ipv6-ext-community-type
+-- ipv6-ext-community-raw*
|   string
+-- large-community*
|   bct:bgp-large-community-type
+-- last-modified?
|   yang:timeticks
+-- eligible-route?
|   boolean
+-- ineligible-reason?
|   identityref
+-- unknown-attributes
|   +-- unknown-attribute*
|       [attr-type]
|       +-- attr-type      uint8
|       +-- optional?      boolean

```

```

|         +-- transitive?    boolean
|         +-- partial?      boolean
|         +-- extended?     boolean
|         +-- attr-len?     uint16
|         +-- attr-value?   binary
|         +-- reject-reason?
|             union
+--:(adj-rib-in-pre)
+-- adj-rib-in-pre
+-- route
+-- prefix?
|   inet:ipv4-prefix
+-- path-id?
|   uint32
+-- attributes
+-- origin?
|   bt:bgp-origin-attr-type
+-- as-path
+-- segment*
+-- type?
|   identityref
+-- member*
|   inet:as-number
+-- next-hop?
|   inet:ip-address
+-- link-local-next-hop?
|   inet:ipv6-address
+-- med?
|   uint32
+-- local-pref?
|   uint32
+-- as4-path
+-- segment*
+-- type?
|   identityref
+-- member*
|   inet:as-number
+-- aggregator
+-- as?
|   inet:as-number
+-- identifier?
|   yang:dotted-quad
+-- aggregator4
+-- as4?
|   inet:as-number
+-- identifier?
|   yang:dotted-quad
+-- atomic-aggregate?

```

```

|         boolean
|         +-- originator-id?
|         |         yang:dotted-quad
|         +-- cluster-list*
|         |         yang:dotted-quad
|         +-- aigp-metric?
|         |         uint64
+-- community*
|         union
+-- ext-community*
|         bct:bgp-ext-community-type
+-- ext-community-raw*
|         string
+-- ipv6-ext-community*
|         bct:bgp-ipv6-ext-community-type
+-- ipv6-ext-community-raw*
|         string
+-- large-community*
|         bct:bgp-large-community-type
+-- last-modified?
|         yang:timeticks
+-- eligible-route?
|         boolean
+-- ineligible-reason?
|         identityref
+-- unknown-attributes
|         +-- unknown-attribute*
|         |         [attr-type]
|         |         +-- attr-type          uint8
|         |         +-- optional?         boolean
|         |         +-- transitive?       boolean
|         |         +-- partial?          boolean
|         |         +-- extended?         boolean
|         |         +-- attr-len?         uint16
|         |         +-- attr-value?       binary
+-- reject-reason?
|         union
+--:(adj-rib-in-post)
+-- adj-rib-in-post
+-- route
+-- prefix?
|         inet:ipv4-prefix
+-- path-id?
|         uint32
+-- attributes
|         +-- origin?
|         |         bt:bgp-origin-attr-type
+-- as-path

```

```

    +-- segment*
    |   +-- type?
    |   |   identityref
    |   +-- member*
    |       inet:as-number
+-- next-hop?
|   inet:ip-address
+-- link-local-next-hop?
|   inet:ipv6-address
+-- med?
|   uint32
+-- local-pref?
|   uint32
+-- as4-path
|   +-- segment*
|   |   +-- type?
|   |   |   identityref
|   |   +-- member*
|   |       inet:as-number
+-- aggregator
|   +-- as?
|   |   inet:as-number
|   +-- identifier?
|       yang:dotted-quad
+-- aggregator4
|   +-- as4?
|   |   inet:as-number
|   +-- identifier?
|       yang:dotted-quad
+-- atomic-aggregate?
|   boolean
+-- originator-id?
|   yang:dotted-quad
+-- cluster-list*
|   yang:dotted-quad
+-- aigp-metric?
|   uint64
+-- community*
|   union
+-- ext-community*
|   bct:bgp-ext-community-type
+-- ext-community-raw*
|   string
+-- ipv6-ext-community*
|   bct:bgp-ipv6-ext-community-type
+-- ipv6-ext-community-raw*
|   string
+-- large-community*

```

```

|         bct:bgp-large-community-type
+-- last-modified?
|         yang:timeticks
+-- eligible-route?
|         boolean
+-- ineligible-reason?
|         identityref
+-- best-path?
|         boolean
+-- unknown-attributes
|   +-- unknown-attribute*
|     [attr-type]
|       +-- attr-type      uint8
|       +-- optional?     boolean
|       +-- transitive?   boolean
|       +-- partial?      boolean
|       +-- extended?     boolean
|       +-- attr-len?     uint16
|       +-- attr-value?   binary
+-- reject-reason?
|   union
+--:(adj-rib-out-pre)
+-- adj-rib-out-pre
+-- route
+-- prefix?
|   inet:ipv4-prefix
+-- path-id?
|   uint32
+-- attributes
|   +-- origin?
|   |   bt:bgp-origin-attr-type
+-- as-path
|   +-- segment*
|   |   +-- type?
|   |   |   identityref
|   |   +-- member*
|   |       inet:as-number
+-- next-hop?
|   inet:ip-address
+-- link-local-next-hop?
|   inet:ipv6-address
+-- med?
|   uint32
+-- local-pref?
|   uint32
+-- as4-path
|   +-- segment*
|   |   +-- type?

```

```

|         identityref
|         +-- member*
|           inet:as-number
+-- aggregator
|   +-- as?
|     |   inet:as-number
|   +-- identifier?
|     yang:dotted-quad
+-- aggregator4
|   +-- as4?
|     |   inet:as-number
|   +-- identifier?
|     yang:dotted-quad
+-- atomic-aggregate?
|   boolean
+-- originator-id?
|   yang:dotted-quad
+-- cluster-list*
|   yang:dotted-quad
+-- aigp-metric?
|   uint64
+-- community*
|   union
+-- ext-community*
|   bct:bgp-ext-community-type
+-- ext-community-row*
|   string
+-- ipv6-ext-community*
|   bct:bgp-ipv6-ext-community-type
+-- ipv6-ext-community-row*
|   string
+-- large-community*
|   bct:bgp-large-community-type
+-- last-modified?
|   yang:timeticks
+-- eligible-route?
|   boolean
+-- ineligible-reason?
|   identityref
+-- unknown-attributes
|   +-- unknown-attribute*
|     [attr-type]
|       +-- attr-type      uint8
|       +-- optional?     boolean
|       +-- transitive?   boolean
|       +-- partial?      boolean
|       +-- extended?     boolean
|       +-- attr-len?     uint16

```

```

|         |         +-- attr-value?   binary
|         |         +-- reject-reason?
|         |         union
+---:(adj-rib-out-post)
+--- adj-rib-out-post
+--- route
+--- prefix?
|   inet:ipv4-prefix
+--- path-id?
|   uint32
+--- attributes
|   +-- origin?
|   |   bt:bgp-origin-attr-type
+--- as-path
|   +-- segment*
|   |   +-- type?
|   |   |   identityref
|   |   +-- member*
|   |       inet:as-number
+--- next-hop?
|   inet:ip-address
+--- link-local-next-hop?
|   inet:ipv6-address
+--- med?
|   uint32
+--- local-pref?
|   uint32
+--- as4-path
|   +-- segment*
|   |   +-- type?
|   |   |   identityref
|   |   +-- member*
|   |       inet:as-number
+--- aggregator
|   +-- as?
|   |   inet:as-number
|   +-- identifier?
|       yang:dotted-quad
+--- aggregator4
|   +-- as4?
|   |   inet:as-number
|   +-- identifier?
|       yang:dotted-quad
+--- atomic-aggregate?
|   boolean
+--- originator-id?
|   yang:dotted-quad
+--- cluster-list*

```

```

| | | yang:dotted-quad
| | +-- aigp-metric?
| | | uint64
| +-- community*
| | union
| +-- ext-community*
| | bct:bgp-ext-community-type
| +-- ext-community-raw*
| | string
| +-- ipv6-ext-community*
| | bct:bgp-ipv6-ext-community-type
| +-- ipv6-ext-community-raw*
| | string
| +-- large-community*
| | bct:bgp-large-community-type
| +-- last-modified?
| | yang:timeticks
| +-- eligible-route?
| | boolean
| +-- ineligible-reason?
| | identityref
| +-- unknown-attributes
| | +-- unknown-attribute*
| | | [attr-type]
| | | +-- attr-type      uint8
| | | +-- optional?     boolean
| | | +-- transitive?   boolean
| | | +-- partial?      boolean
| | | +-- extended?     boolean
| | | +-- attr-len?     uint16
| | | +-- attr-value?   binary
| +-- reject-reason?
| | union
+--:(ipv6-unicast)
+-- ipv6-unicast
+-- (rib-type)
+--:(loc-rib)
+-- loc-rib
+-- route
+-- prefix
| | inet:ipv6-prefix
+-- origin?
| | union
+-- path-id?
| | uint32
+-- attributes
| | +-- origin?
| | | bt:bgp-origin-attr-type

```



```

+-- as-path
|   +-- segment*
|       +-- type?
|           | identityref
|       +-- member*
|           inet:as-number
+-- next-hop?
|   inet:ip-address
+-- link-local-next-hop?
|   inet:ipv6-address
+-- med?
|   uint32
+-- local-pref?
|   uint32
+-- as4-path
|   +-- segment*
|       +-- type?
|           | identityref
|       +-- member*
|           inet:as-number
+-- aggregator
|   +-- as?
|       | inet:as-number
|   +-- identifier?
|       yang:dotted-quad
+-- aggregator4
|   +-- as4?
|       | inet:as-number
|   +-- identifier?
|       yang:dotted-quad
+-- atomic-aggregate?
|   boolean
+-- originator-id?
|   yang:dotted-quad
+-- cluster-list*
|   yang:dotted-quad
+-- aigp-metric?
|   uint64
+-- community*
|   union
+-- ext-community*
|   bct:bgp-ext-community-type
+-- ext-community-row*
|   string
+-- ipv6-ext-community*
|   bct:bgp-ipv6-ext-community-type
+-- ipv6-ext-community-row*
|   string

```

```

    +-- large-community*
    |   bct:bgp-large-community-type
    +-- last-modified?
    |   yang:timeticks
    +-- eligible-route?
    |   boolean
    +-- ineligible-reason?
    |   identityref
    +-- unknown-attributes
    |   +-- unknown-attribute*
    |   |   [attr-type]
    |   |   +-- attr-type      uint8
    |   |   +-- optional?     boolean
    |   |   +-- transitive?   boolean
    |   |   +-- partial?      boolean
    |   |   +-- extended?     boolean
    |   |   +-- attr-len?     uint16
    |   |   +-- attr-value?   binary
    |   +-- reject-reason?
    |   |   union
    +--:(adj-rib-in-pre)
    +-- adj-rib-in-pre
    |   +-- route
    |   |   +-- prefix?
    |   |   |   inet:ipv6-prefix
    |   |   +-- path-id?
    |   |   |   uint32
    |   |   +-- attributes
    |   |   |   +-- origin?
    |   |   |   |   bt:bgp-origin-attr-type
    |   |   |   +-- as-path
    |   |   |   |   +-- segment*
    |   |   |   |   |   +-- type?
    |   |   |   |   |   |   identityref
    |   |   |   |   |   +-- member*
    |   |   |   |   |   |   inet:as-number
    |   |   |   +-- next-hop?
    |   |   |   |   inet:ip-address
    |   |   |   +-- link-local-next-hop?
    |   |   |   |   inet:ipv6-address
    |   |   |   +-- med?
    |   |   |   |   uint32
    |   |   |   +-- local-pref?
    |   |   |   |   uint32
    |   |   |   +-- as4-path
    |   |   |   |   +-- segment*
    |   |   |   |   |   +-- type?
    |   |   |   |   |   |   identityref

```

```

|         +-- member*
|         |         inet:as-number
+-- aggregator
|   +-- as?
|   |     inet:as-number
|   +-- identifier?
|       yang:dotted-quad
+-- aggregator4
|   +-- as4?
|   |     inet:as-number
|   +-- identifier?
|       yang:dotted-quad
+-- atomic-aggregate?
|   boolean
+-- originator-id?
|   yang:dotted-quad
+-- cluster-list*
|   yang:dotted-quad
+-- aigp-metric?
|   uint64
+-- community*
|   union
+-- ext-community*
|   bct:bgp-ext-community-type
+-- ext-community-raw*
|   string
+-- ipv6-ext-community*
|   bct:bgp-ipv6-ext-community-type
+-- ipv6-ext-community-raw*
|   string
+-- large-community*
|   bct:bgp-large-community-type
+-- last-modified?
|   yang:timeticks
+-- eligible-route?
|   boolean
+-- ineligible-reason?
|   identityref
+-- unknown-attributes
|   +-- unknown-attribute*
|       [attr-type]
|       +-- attr-type      uint8
|       +-- optional?     boolean
|       +-- transitive?   boolean
|       +-- partial?      boolean
|       +-- extended?     boolean
|       +-- attr-len?     uint16
|       +-- attr-value?   binary

```

```

    +-- reject-reason?
        union
+---:(adj-rib-in-post)
    +-- adj-rib-in-post
        +-- route
            +-- prefix?
                |
                inet:ipv6-prefix
            +-- path-id?
                |
                uint32
            +-- attributes
                +-- origin?
                    |
                    bt:bgp-origin-attr-type
                +-- as-path
                    |
                    +-- segment*
                        |
                        +-- type?
                            |
                            identityref
                        +-- member*
                            |
                            inet:as-number
                +-- next-hop?
                    |
                    inet:ip-address
                +-- link-local-next-hop?
                    |
                    inet:ipv6-address
                +-- med?
                    |
                    uint32
                +-- local-pref?
                    |
                    uint32
                +-- as4-path
                    |
                    +-- segment*
                        |
                        +-- type?
                            |
                            identityref
                        +-- member*
                            |
                            inet:as-number
                +-- aggregator
                    |
                    +-- as?
                        |
                        inet:as-number
                    +-- identifier?
                        |
                        yang:dotted-quad
                +-- aggregator4
                    |
                    +-- as4?
                        |
                        inet:as-number
                    +-- identifier?
                        |
                        yang:dotted-quad
                +-- atomic-aggregate?
                    |
                    boolean
                +-- originator-id?
                    |
                    yang:dotted-quad
                +-- cluster-list*
                    |
                    yang:dotted-quad

```

```

|   +-- aigp-metric?
|       uint64
+-- community*
|   union
+-- ext-community*
|   bct:bgp-ext-community-type
+-- ext-community-raw*
|   string
+-- ipv6-ext-community*
|   bct:bgp-ipv6-ext-community-type
+-- ipv6-ext-community-raw*
|   string
+-- large-community*
|   bct:bgp-large-community-type
+-- last-modified?
|   yang:timeticks
+-- eligible-route?
|   boolean
+-- ineligible-reason?
|   identityref
+-- best-path?
|   boolean
+-- unknown-attributes
|   +-- unknown-attribute*
|       [attr-type]
|       +-- attr-type      uint8
|       +-- optional?     boolean
|       +-- transitive?   boolean
|       +-- partial?      boolean
|       +-- extended?     boolean
|       +-- attr-len?     uint16
|       +-- attr-value?   binary
+-- reject-reason?
|   union
+--:(adj-rib-out-pre)
+-- adj-rib-out-pre
+-- route
+-- prefix?
|   inet:ipv6-prefix
+-- path-id?
|   uint32
+-- attributes
|   +-- origin?
|       bt:bgp-origin-attr-type
+-- as-path
|   +-- segment*
|       +-- type?
|           identityref

```

```

|         +-- member*
|         |         inet:as-number
+-- next-hop?
|         inet:ip-address
+-- link-local-next-hop?
|         inet:ipv6-address
+-- med?
|         uint32
+-- local-pref?
|         uint32
+-- as4-path
|   +-- segment*
|   |   +-- type?
|   |   |   identityref
|   |   +-- member*
|   |       inet:as-number
+-- aggregator
|   +-- as?
|   |   inet:as-number
|   +-- identifier?
|       yang:dotted-quad
+-- aggregator4
|   +-- as4?
|   |   inet:as-number
|   +-- identifier?
|       yang:dotted-quad
+-- atomic-aggregate?
|   boolean
+-- originator-id?
|   yang:dotted-quad
+-- cluster-list*
|   yang:dotted-quad
+-- aigp-metric?
|   uint64
+-- community*
|   union
+-- ext-community*
|   bct:bgp-ext-community-type
+-- ext-community-raw*
|   string
+-- ipv6-ext-community*
|   bct:bgp-ipv6-ext-community-type
+-- ipv6-ext-community-raw*
|   string
+-- large-community*
|   bct:bgp-large-community-type
+-- last-modified?
|   yang:timeticks

```

```

    |-- eligible-route?
    |   boolean
    |-- ineligible-reason?
    |   identityref
    |-- unknown-attributes
    |   |-- unknown-attribute*
    |   |   [attr-type]
    |   |   |-- attr-type      uint8
    |   |   |-- optional?     boolean
    |   |   |-- transitive?   boolean
    |   |   |-- partial?      boolean
    |   |   |-- extended?     boolean
    |   |   |-- attr-len?     uint16
    |   |   |-- attr-value?   binary
    |   |-- reject-reason?
    |   |   union
    |--:(adj-rib-out-post)
    |-- adj-rib-out-post
    |-- route
    |   |-- prefix?
    |   |   inet:ipv6-prefix
    |-- path-id?
    |   uint32
    |-- attributes
    |   |-- origin?
    |   |   bt:bgp-origin-attr-type
    |-- as-path
    |   |-- segment*
    |   |   |-- type?
    |   |   |   identityref
    |   |   |-- member*
    |   |   |   inet:as-number
    |-- next-hop?
    |   inet:ip-address
    |-- link-local-next-hop?
    |   inet:ipv6-address
    |-- med?
    |   uint32
    |-- local-pref?
    |   uint32
    |-- as4-path
    |   |-- segment*
    |   |   |-- type?
    |   |   |   identityref
    |   |   |-- member*
    |   |   |   inet:as-number
    |-- aggregator
    |   |-- as?

```

```

| | | inet:as-number
| | | +--- identifier?
| | | | yang:dotted-quad
+--- aggregator4
| | | +--- as4?
| | | | inet:as-number
| | | +--- identifier?
| | | | yang:dotted-quad
+--- atomic-aggregate?
| | | boolean
+--- originator-id?
| | | yang:dotted-quad
+--- cluster-list*
| | | yang:dotted-quad
+--- aigp-metric?
| | | uint64
+--- community*
| | | union
+--- ext-community*
| | | bct:bgp-ext-community-type
+--- ext-community-row*
| | | string
+--- ipv6-ext-community*
| | | bct:bgp-ipv6-ext-community-type
+--- ipv6-ext-community-row*
| | | string
+--- large-community*
| | | bct:bgp-large-community-type
+--- last-modified?
| | | yang:timeticks
+--- eligible-route?
| | | boolean
+--- ineligible-reason?
| | | identityref
+--- unknown-attributes
| | | +--- unknown-attribute*
| | | | [attr-type]
| | | | +--- attr-type      uint8
| | | | +--- optional?    boolean
| | | | +--- transitive?  boolean
| | | | +--- partial?     boolean
| | | | +--- extended?    boolean
| | | | +--- attr-len?    uint16
| | | | +--- attr-value?  binary
+--- reject-reason?
| | | union
+--- information?
| | | enumeration

```


Figure 1: YANG tree diagram for 'ietf-bmp-telemetry-message' module.

```
<CODE BEGINS> file "ietf-bmp-telemetry-message@2026-02-09.yang"
module ietf-bmp-telemetry-message {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-bmp-telemetry-message";
  prefix bmptm;

  import ietf-telemetry-message {
    prefix tm;
    reference
      "draft-netana-nmop-message-broker-telemetry-message: Extensible
       YANG Model for Network Telemetry Messages";
  }
  import ietf-inet-types {
    prefix inet;
    reference
      "RFC 9911: Common YANG Data Types";
  }
  import ietf-yang-types {
    prefix yang;
    reference
      "RFC 9911: Common YANG Data Types";
  }
  import ietf-routing-types {
    prefix rt-types;
    reference
      "RFC 8294: Common YANG Data Types for the Routing Area.";
  }
  import ietf-yang-structure-ext {
    prefix sx;
    reference
      "RFC 8791: YANG Data Structure Extensions";
  }
  include ietf-bmp-bgp-rib-entry {
    revision-date 2026-02-09;
  }
  include ietf-bmp-bgp-rib-tables {
    revision-date 2026-02-09;
  }
  include ietf-bmp-bgp-rib-attributes {
    revision-date 2026-02-09;
  }

  organization
    "IETF Draft";
  contact
    "Author:      Thomas Graf
```

```
<mailto:thomas.graf@swisscom.com>
Paolo Lucente
<mailto:paolo@ntt.net>
Leonardo Rodoni
<mailto:leonardo.rodoni@swisscom.com>;
```

description

"Adds BMP specific subscription metadata to the data collection protocol provenance of the ietf-telemetry-message envelope.

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) (RFC 8174) when, and only when, they appear in all capitals, as shown here.

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

Redistribution and use in source and binary forms, with or without modification, is permitted pursuant to, and subject to the license terms contained in, the Revised BSD License set forth in Section 4.c of the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>).

This version of this YANG module is part of RFC XXXX; see the RFC itself for full legal notices.";

revision 2026-02-09 {

description

"Initial revision.";

reference

"RFC XXXX";

}

typedef peer-type {

type enumeration {

enum global-instance-peer {

value 0;

description

"Global Instance Peer";

reference

"RFC 7854: BGP Monitoring Protocol (BMP), Section 4.2.";

}

enum rd-instance-peer {

value 1;

description

```
        "RD Instance Peer";
    reference
        "RFC 7854: BGP Monitoring Protocol (BMP), Section 4.2.";
}
enum local-instance-peer {
    value 2;
    description
        "Local Instance Peer";
    reference
        "RFC 7854: BGP Monitoring Protocol (BMP), Section 4.2.";
}
enum loc-rib-instance-peer {
    value 3;
    description
        "Loc-RIB Instance Peer";
    reference
        "RFC 9069: Support for Local RIB in the BGP Monitoring Protocol (BMP)";
}
}
description
    "BMP Peer Type.";
reference
    "RFC 7854: BGP Monitoring Protocol (BMP), Section 4.2.";
}

grouping per-peer {
    description
        "BMP per-peer header.";
    reference
        "RFC 7854: BGP Monitoring Protocol (BMP), Section 4.2.";
    leaf peer-type {
        type peer-type;
        description
            "BMP peer type from per-peer header.";
        reference
            "RFC 7854: BGP Monitoring Protocol (BMP), Section 4.2.";
    }
    leaf peer-flags {
        type uint8;
        description
            "BMP peer flags from per-peer header.";
        reference
            "RFC 7854: BGP Monitoring Protocol (BMP), Section 4.2.";
    }
    leaf peer-distinguisher {
        type rt-types:route-distinguisher;
        description
            "BMP peer distinguisher from per-peer header.";
```

```
    reference
      "RFC 7854: BGP Monitoring Protocol (BMP), Section 4.2.
      The routing instance on which the encapsulated PDU was
      received.";
  }
  leaf peer-address {
    type inet:ip-address;
    description
      "BMP peer IP address from per-peer header.";
    reference
      "RFC 7854: BGP Monitoring Protocol (BMP), Section 4.2. The
      remote IP address associated with the TCP session over which
      the encapsulated PDU was received.";
  }
  leaf peer-as {
    type uint32;
    description
      "BMP peer autonomous system number from per-peer header. The
      Autonomous System number of the peer from which the
      encapsulated PDU was received.";
    reference
      "RFC 7854: BGP Monitoring Protocol (BMP), Section 4.2.";
  }
  leaf timestamp {
    type yang:date-and-time;
    description
      "BMP timestamp from per-peer header. The time when the
      encapsulated routes were received (one may also think of this
      as the time when they were installed in the Adj-RIB-In).";
    reference
      "RFC 7854: BGP Monitoring Protocol (BMP), Section 4.2.";
  }
}

grouping information {
  description
    "BMP Information.";
  reference
    "RFC 7854: BGP Monitoring Protocol (BMP), Section 4.4.";
  leaf string {
    type string {
      length "1..65535";
    }
    description
      "BMP string type. The Information field contains a free-form
      UTF-8 string.";
  }
  leaf sys-descr {
```

```
    type string {
      length "1..65535";
    }
    description
      "BMP sysDescr type. The Information field contains an ASCII
      string whose value MUST be set to be equal to the value of the
      sysDescr MIB-II RFC 1213 object.";
  }
  leaf sys-name {
    type string {
      length "1..65535";
    }
    description
      "BMP sysName type. The Information field contains an ASCII
      string whose value MUST be set to be equal to the value of the
      sysName MIB-II RFC 1213 object.";
    reference
      "RFC 7854: BGP Monitoring Protocol (BMP), Section 4.2.";
  }
}

grouping termination {
  description
    "BMP Termination.";
  reference
    "RFC 7854: BGP Monitoring Protocol (BMP), Section 4.5.";
  leaf string {
    type string {
      length "1..65535";
    }
    description
      "BMP string type. The Information field contains a free-form
      UTF-8 string.";
  }
  leaf reason {
    type enumeration {
      enum administratively-closed {
        value 0;
      }
      enum unspecified {
        value 1;
      }
      enum out-of-resources {
        value 2;
      }
      enum redundant-connection {
        value 3;
      }
    }
  }
}
```

```
        enum permanently-administratively-closed {
            value 4;
        }
    }
}

grouping mirroring {
    description
        "BMP Mirroring.";
    reference
        "RFC 7854: BGP Monitoring Protocol (BMP), Section 4.7.";
    leaf information {
        type enumeration {
            enum error-pdu {
                value 0;
            }
            enum message-lost {
                value 1;
            }
        }
    }
}

grouping statistics {
    description
        "BMP Statistics.";
    reference
        "RFC 7854: BGP Monitoring Protocol (BMP), Section 4.8.";
    leaf statistics-type {
        type string {
        }
        description
            "BMP statistics type. Defines the type of the statistic carried
            in the Stat Data field.";
    }
    leaf statistics-data {
        type uint64;
        description
            "BMP statistics data. Defines the value of the statistic.";
    }
}

sx:structure message {
    description
        "Augments telemetry-message-metadata with BMP specific
        subscription metadata";
```

```
leaf version {
  type uint8;
  description
    "BMP version from common header.";
  reference
    "RFC 7854: BGP Monitoring Protocol (BMP), Section 4.1.";
}
choice message-type {
  description
    "BMP message type from common header.";
  reference
    "RFC 7854: BGP Monitoring Protocol (BMP), Section 4.1.";
  case route-monitoring {
    container route-monitoring {
      uses per-peer;
      uses rib-entry;
    }
  }
  case statistics-report {
    container statistics-report {
      uses per-peer;
      uses statistics;
    }
  }
  case peer-down-notification {
    container peer-down-notification {
      uses per-peer;
    }
  }
  case peer-up-notification {
    container peer-up-notification {
      uses per-peer;
    }
  }
  case initiation-message {
    container initiation-message {
      uses information;
    }
  }
  case termination-message {
    container termination-message {
      uses termination;
    }
  }
  case route-mirroring-message {
    container route-mirroring-message {
      uses per-peer;
      uses rib-entry;
    }
  }
}
```

```

        uses mirroring;
    }
}
}
}
}
<CODE ENDS>

```

Figure 2: BMP YANG 'ietf-bmp-telemetry-message' module.

4.1. BMP BGP RIB YANG Submodules

```

<CODE BEGINS> file "ietf-bmp-bgp-rib-entry@2026-02-09.yang"
(Artwork only available as YANG: see https://www.ietf.org/archive/id/
draft-netana-nmop-message-broker-bmp-telemetry-msg-02.html)
<CODE ENDS>

```

Figure 3: BMP YANG 'ietf-bmp-bgp-rib-entry' submodule.

```

<CODE BEGINS> file "ietf-bmp-bgp-rib-tables@2026-02-09.yang"
submodule ietf-bmp-bgp-rib-tables {
    yang-version 1.1;
    belongs-to ietf-bmp-telemetry-message {
        prefix bmptm;
    }

    // import some basic types
    import ietf-inet-types {
        prefix inet;
        reference
            "RFC 6991: Common YANG Data Types.";
    }
    import ietf-yang-types {
        prefix yang;
        reference
            "RFC 6991: Common YANG Data Types.";
    }
    import ietf-routing {
        prefix rt;
        reference
            "RFC 8022: A YANG Data Model for Routing Management.";
    }
    import iana-bgp-rib-types {
        prefix brt;
        reference
            "RFC XXXX: YANG module for Border Gateway Protocol (BGP-4).";
    }
}

```


organization

"IETF Draft";

contact

"Author: Thomas Graf
<mailto:thomas.graf@swisscom.com>
Paolo Lucente
<mailto:paolo@ntt.net>
Leonardo Rodoni
<mailto:leonardo.rodoni@swisscom.com>";

description

"This submodule contains structural data definitions for
BGP routing tables.

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

Redistribution and use in source and binary forms, with or
without modification, is permitted pursuant to, and subject to
the license terms contained in, the Revised BSD License set
forth in Section 4.c of the IETF Trust's Legal Provisions
Relating to IETF Documents
(<https://trustee.ietf.org/license-info>).

This version of this YANG module is part of RFC XXXX
(<https://www.rfc-editor.org/info/rfcXXXX>); see the RFC itself
for full legal notices.

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) (RFC 8174) when, and only when,
they appear in all capitals, as shown here.";

```
revision 2026-02-09 {  
  description  
    "Initial Version";  
  reference  
    "RFC XXXX";  
}
```

```
grouping bgp-common-route-annotations-state {  
  description  
    "Data definitions for flags and other information attached  
    to routes in both LOC-RIB and Adj-RIB";  
  leaf last-modified {  
    type yang:timeticks;  
    description
```

```
    "Timestamp when this path was last modified.

    The value is the timestamp in seconds relative to
    the Unix Epoch (Jan 1, 1970 00:00:00 UTC).";
}
leaf eligible-route {
    type boolean;
    description
        "Indicates that the route is eligible for selection for the
        best route in the Loc-Rib in BGP's Decision Process.";
    reference
        "RFC 4271: A Border Gateway Protocol 4 (BGP-4),
        Section 9.1.";
}
leaf ineligible-reason {
    type identityref {
        base brt:ineligible-route-reason;
    }
    description
        "If the route is ineligible for selection for the best route
        in the Loc-Rib in BGP's Decision process, this indicates the
        reason.";
    reference
        "RFC 4271: A Border Gateway Protocol 4 (BGP-4),
        Section 9.1.";
}
}

grouping bgp-adj-rib-in-post-route-annotations-state {
    description
        "Data definitions for information attached to routes in the
        Adj-RIB-In post-policy table";
    leaf best-path {
        type boolean;
        default "false";
        description
            "Current path was selected as the best path. Best path
            should indicate that the route is present in BGP Loc-RIB.";
    }
}

grouping rib-ext-route-annotations {
    description
        "Extended annotations for routes in the routing tables";
    leaf reject-reason {
        type union {
            type identityref {
                base brt:bgp-not-selected-bestpath;
            }
        }
    }
}
```

```
    }
    type identityref {
      base brt:bgp-not-selected-policy;
    }
  }
  description
    "Indicates the reason the route is not used, either due to
    policy filtering or bestpath selection";
}
}

grouping bgp-loc-rib-common-keys {
  description
    "Common references used in keys for IPv4 and IPv6
    Loc-RIB entries.";
  leaf origin {
    type union {
      type inet:ip-address;
      type identityref {
        base rt:routing-protocol;
      }
    }
  }
  description
    "Indicates the origin of the route. If the route is learned
    from a neighbor, this value is the neighbor address. If
    the route was injected or redistributed from another
    protocol, the origin indicates the source protocol for the
    route.";
}
  leaf path-id {
    type uint32;
    description
      "If the route is learned from a neighbor, the path-id
      corresponds to the path-id for the route in the
      corresponding adj-rib-in-post table. If the route is
      injected from another protocol, or the neighbor does not
      support BGP add-paths, the path-id should be set
      to zero, also the default value.

      However, YANG does not allow default values to be set
      for parameters that form the key, so a default value
      cannot be set here.";
  }
}

grouping ipv4-loc-rib-common {
  description
    "Common structural grouping for the IPv4 Loc-RIB table.";
```

```
    container route {
      config false;
      description
        "A single route entry in the Loc-RIB.";
      leaf prefix {
        type inet:ipv4-prefix;
        mandatory true;
        description
          "The IPv4 prefix corresponding to the route.";
      }
      uses bgp-loc-rib-common-keys;
      uses bgp-common-route-attributes;
      uses bgp-common-route-annotations-state;
      uses bgp-unknown-attr-top;
      uses rib-ext-route-annotations;
    }
  }

  grouping ipv6-loc-rib-common {
    description
      "Common structural grouping for the IPv6 Loc-RIB table.";
    container route {
      config false;
      description
        "A single route entry in the Loc-RIB.";
      leaf prefix {
        type inet:ipv6-prefix;
        mandatory true;
        description
          "The IPv6 prefix corresponding to the route.";
      }
      uses bgp-loc-rib-common-keys;
      uses bgp-common-route-attributes;
      uses bgp-common-route-annotations-state;
      uses bgp-unknown-attr-top;
      uses rib-ext-route-annotations;
    }
  }

  grouping ipv4-adj-rib-common {
    description
      "Common structural grouping for each IPv4 Adj-RIB table.";
    container route {
      config false;
      description
        "A single route entry in the IPv4 Adj-RIB table.";
      leaf prefix {
        type inet:ipv4-prefix;
```

```
        description
            "The IPv4 prefix corresponding to the route.";
    }
    uses bgp-adj-rib-attr-state;
    uses bgp-common-route-attributes;
    uses bgp-common-route-annotations-state;
    uses bgp-unknown-attr-top;
    uses rib-ext-route-annotations;
}
}

grouping ipv4-adj-rib-in-post {
    description
        "Common structural grouping for the IPv4 Adj-RIB-In
        post-policy table.";
    container route {
        config false;
        description
            "A single route entry in the IPv4 Adj-RIB-In post-policy table.";
        leaf prefix {
            type inet:ipv4-prefix;
            description
                "The IPv4 prefix corresponding to the route.";
        }
        uses bgp-adj-rib-attr-state;
        uses bgp-common-route-attributes;
        uses bgp-common-route-annotations-state;
        uses bgp-adj-rib-in-post-route-annotations-state;
        uses bgp-unknown-attr-top;
        uses rib-ext-route-annotations;
    }
}

grouping ipv6-adj-rib-common {
    description
        "Common structural grouping for each IPv6 Adj-RIB table.";
    container route {
        config false;
        description
            "A single route entry in the IPv6 Adj-RIB table.";
        leaf prefix {
            type inet:ipv6-prefix;
            description
                "The IPv6 prefix corresponding to the route.";
        }
        uses bgp-adj-rib-attr-state;
        uses bgp-common-route-attributes;
        uses bgp-common-route-annotations-state;
    }
}
```

```
        uses bgp-unknown-attr-top;
        uses rib-ext-route-annotations;
    }
}

grouping ipv6-adj-rib-in-post {
    description
        "Common structural grouping for the IPv6 Adj-RIB-In
        post-policy table.";
    container route {
        config false;
        description
            "A single route entry in the IPv6 Adj-RIB-In post-policy table.";
        leaf prefix {
            type inet:ipv6-prefix;
            description
                "The IPv6 prefix corresponding to the route.";
        }
        uses bgp-adj-rib-attr-state;
        uses bgp-common-route-attributes;
        uses bgp-common-route-annotations-state;
        uses bgp-adj-rib-in-post-route-annotations-state;
        uses bgp-unknown-attr-top;
        uses rib-ext-route-annotations;
    }
}
}
<CODE ENDS>
```

Figure 4: BMP YANG 'ietf-bmp-bgp-rib-tables' submodule.

```
<CODE BEGINS> file "ietf-bmp-bgp-rib-attributes@2026-02-09.yang"
submodule ietf-bmp-bgp-rib-attributes {
    yang-version 1.1;
    belongs-to ietf-bmp-telemetry-message {
        prefix bmptm;
    }

    // import some basic types
    import iana-bgp-types {
        prefix bt;
        reference
            "RFC XXXX: YANG Model for Border Gateway Protocol (BGP-4).";
    }
    import iana-bgp-community-types {
        prefix bct;
        reference
            "RFC XXXX: YANG Model for Border Gateway Protocol (BGP-4).";
    }
}
```

```
}
import ietf-inet-types {
  prefix inet;
  reference
    "RFC 6991: Common YANG Data Types.";
}
import ietf-yang-types {
  prefix yang;
  reference
    "RFC 6991: Common YANG Types.";
}

// meta
organization
  "IETF Draft";
contact
  "Author:      Thomas Graf
                <mailto:thomas.graf@swisscom.com>
                Paolo Lucente
                <mailto:paolo@ntt.net>
                Leonardo Rodoni
                <mailto:leonardo.rodoni@swisscom.com>";

description
  "This submodule contains common data definitions for BGP
  attributes for use in BGP RIB tables.

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

  Redistribution and use in source and binary forms, with or
  without modification, is permitted pursuant to, and subject to
  the license terms contained in, the Revised BSD License set
  forth in Section 4.c of the IETF Trust's Legal Provisions
  Relating to IETF Documents
  (https://trustee.ietf.org/license-info).

  This version of this YANG module is part of RFC XXXX
  (https://www.rfc-editor.org/info/rfcXXXX); see the RFC itself
  for full legal notices.

  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) (RFC 8174) when, and only when,
  they appear in all capitals, as shown here.";

revision 2026-02-09 {
```

```
    description
      "Initial version";
    reference
      "RFC XXXX";
  }

  grouping bgp-common-route-attributes {
    description
      "BGP path attributes and communities.";
    uses bgp-path-attributes;
    uses bgp-community-attr-state;
    uses ext-community-attributes;
    uses ipv6-ext-community-attributes;
    uses large-community-attributes;
  }

  grouping bgp-path-attributes {
    description
      "A grouping for all BGP Path Attribute parameters.";

    container attributes {
      description
        "A container for attribute parameters.";
      leaf origin {
        type bt:bgp-origin-attr-type;
        description
          "BGP attribute defining the origin of the path
            information.";
        reference
          "RFC 4271: A Border Gateway Protocol 4 (BGP-4), Section
            5.1.1.";
      }
      container as-path {
        description
          "Enclosing container for the list of AS path segments.

          In the Adj-RIB-In or Adj-RIB-Out, this list should show
          the received or sent AS_PATH, respectively. For
          example, if the local router is not 4-byte capable, this
          value should consist of 2-octet ASNs or the AS_TRANS
          (AS 23456) values received or sent in BGP updates.

          In the Loc-RIB, this list should reflect the effective
          AS path for the route, e.g., a 4-octet value if the
          local router is 4-octet capable.";
        reference
          "RFC 4271: A Border Gateway Protocol 4 (BGP-4),
            RFC 6793: BGP Support for Four-octet AS Number Space
```



```
    RFC 5065: Autonomous System Confederations for BGP.";
  list segment {
    config false;
    uses bgp-as-path-segment;
    description
      "List of AS PATH segments";
  }
}
leaf next-hop {
  type inet:ip-address;
  description
    "BGP next hop attribute defining the IP address of the
    router that should be used as the next hop to the
    destination. Used when the BGP routes' nexthop for that
    AFI/SAFI can be represented as an IPv4/IPv6 address.";
  reference
    "RFC 4271: A Border Gateway Protocol 4 (BGP-4), Section
    5.1.3.";
}
leaf link-local-next-hop {
  type inet:ipv6-address;
  description
    "When both a global and a link-local next-hop are sent
    when following RFC 2545 procedures, this leaf contains
    the link-local next-hop.";
  reference
    "RFC 2545: Use of BGP-4 Multiprotocol Extensions for IPv6
    Inter-Domain Routing.";
}
leaf med {
  type uint32;
  description
    "BGP multi-exit discriminator attribute used in the BGP
    route selection process.";
  reference
    "RFC 4271: A Border Gateway Protocol 4 (BGP-4), Section
    5.1.4.";
}
leaf local-pref {
  type uint32;
  description
    "BGP local preference attribute sent to internal peers to
    indicate the degree of preference for externally learned
    routes. The route with the highest local preference
    value is preferred.";
  reference
    "RFC 4271: A Border Gateway Protocol 4 (BGP-4), Section
    5.1.5.";
```

```
}
container as4-path {
  description
    "This is the path encoded with 4-octet
    AS numbers in the optional transitive AS4_PATH attribute.
    This value is populated with the received or sent
    attribute in Adj-RIB-In or Adj-RIB-Out, respectively.
    It should not be populated in Loc-RIB since the Loc-RIB
    is expected to store the effective AS-Path in the
    as-path leaf regardless of being 4-octet or 2-octet.";
  reference
    "RFC 6793: BGP Support for Four-octet AS Number Space";
  list segment {
    config false;
    uses bgp-as-path-segment;
    description
      "List of AS PATH segments";
  }
}
container aggregator {
  config false;
  description
    "BGP attribute indicating the prefix has been
    aggregated by the specified AS and router.";
  reference
    "RFC 4271: A Border Gateway Protocol 4 (BGP-4), Section
    5.1.7,
    RFC 6793: BGP Support for Four-octet AS Number Space.";
  leaf as {
    type inet:as-number;
    description
      "AS number of the autonomous system that performed the
      aggregation.";
    reference
      "RFC 4271: A Border Gateway Protocol 4 (BGP-4), Section
      4.3, Path Attributes (g).";
  }
  leaf identifier {
    type yang:dotted-quad;
    description
      "BGP Identifier of the router that performed the
      aggregation.";
    reference
      "RFC 4271: A Border Gateway Protocol 4 (BGP-4), Section
      4.3, Path Attributes (g).";
  }
}
container aggregator4 {
```

```
config false;
description
  "BGP attribute indicating the prefix has been
  aggregated by the specified AS and router.
  This value is populated with the received or sent
  attribute in Adj-RIB-In or Adj-RIB-Out, respectively.
  It should not be populated in Loc-RIB since the Loc-RIB
  is expected to store the effective AGGREGATOR in the
  aggregator/as leaf regardless of being 4-octet or
  2-octet.";
reference
  "RFC 6793: BGP Support for Four-octet AS Number Space.";
leaf as4 {
  type inet:as-number;
  description
    "AS number of the autonomous system that performed the
    aggregation (4-octet representation). This value is
    populated if an upstream router is not 4-octet capable.
    Its semantics are similar to the AS4_PATH optional
    transitive attribute";
  reference
    "RFC 6793: BGP Support for Four-octet AS Number Space,
    Section 3,
    RFC 4271: A Border Gateway Protocol 4 (BGP-4), Section
    4.3, Path Attributes (g).";
}
leaf identifier {
  type yang:dotted-quad;
  description
    "BGP Identifier of the router that performed the
    aggregation.";
  reference
    "RFC 6793: BGP Support for Four-octet AS Number Space,
    Section 3,
    RFC 4271: A Border Gateway Protocol 4 (BGP-4), Section
    4.3, Path Attributes (g).";
}
}
leaf atomic-aggregate {
  type boolean;
  description
    "BGP attribute indicating that the prefix is an atomic
    aggregate; i.e., the peer selected is a less specific
    route without selecting a more specific route that is
    subsumed by it.";
  reference
    "RFC 4271: A Border Gateway Protocol 4 (BGP-4), Section
    5.1.6.";
```

```
    }
    leaf originator-id {
      type yang:dotted-quad;
      description
        "BGP attribute that provides the id as an IPv4 address
        of the originator of the announcement.";
      reference
        "RFC 4456: BGP Route Reflection: An Alternative to Full
        Mesh Internal BGP (IBGP).";
    }
    leaf-list cluster-list {
      type yang:dotted-quad;
      description
        "Represents the reflection path that the route has
        passed.";
      reference
        "RFC 4456: BGP Route Reflection: An Alternative to Full
        Mesh Internal BGP (IBGP).";
    }
    leaf aigp-metric {
      type uint64;
      description
        "BGP path attribute representing the accumulated IGP
        metric for the path";
      reference
        "RFC 7311: The Accumulated IGP Metric Attribute for BGP.";
    }
  }
}

grouping bgp-as-path-segment {
  description
    "Data for representing BGP AS-PATH segment";

  leaf type {
    type identityref {
      base bt:as-path-segment-type;
    }
    description
      "The type of AS-PATH segment";
  }
  leaf-list member {
    type inet:as-number;
    description
      "List of the AS numbers in the AS-PATH segment";
  }
}
```

```
grouping bgp-unknown-attr-top {
  description
    "Unknown path attributes that are not expected to be shared
    across route entries, common to LOC-RIB and Adj-RIB";
  container unknown-attributes {
    description
      "Unknown path attributes that were received in the UPDATE
      message which contained the prefix.";

    list unknown-attribute {
      key "attr-type";
      description
        "This list contains received attributes that are
        unrecognized or unsupported by the local router. The list
        may be empty.";

      leaf optional {
        type boolean;
        description
          "Defines whether the attribute is optional (if
          set to true) or well-known (if set to false).
          Set in the high-order bit of the BGP attribute
          flags octet.";
        reference
          "RFC 4271: A Border Gateway Protocol 4 (BGP-4).";
      }

      leaf transitive {
        type boolean;
        description
          "Defines whether an optional attribute is transitive
          (if set to true) or non-transitive (if set to false).
          For well-known attributes, the transitive flag must be
          set to true. Set in the second high-order bit of the BGP
          attribute flags octet.";
        reference
          "RFC 4271: A Border Gateway Protocol 4 (BGP-4).";
      }

      leaf partial {
        type boolean;
        description
          "Defines whether the information contained in the
          optional transitive attribute is partial (if set to
          true) or complete (if set to false). For well-known
          attributes and for optional non-transitive attributes,
          the partial flag must be set to false. Set in the third
          high-order bit of the BGP attribute flags octet.";
      }
    }
  }
}
```

```
        reference
          "RFC 4271: A Border Gateway Protocol 4 (BGP-4).";
      }

      leaf extended {
        type boolean;
        description
          "Defines whether the attribute length is one octet
           (if set to false) or two octets (if set to true). Set in
           the fourth high-order bit of the BGP attribute flags
           octet.";
        reference
          "RFC 4271: A Border Gateway Protocol 4 (BGP-4).";
      }

      leaf attr-type {
        type uint8;
        description
          "1-octet value encoding the attribute type code";
        reference
          "RFC 4271: A Border Gateway Protocol 4 (BGP-4).";
      }

      leaf attr-len {
        type uint16;
        description
          "One or two octet attribute length field indicating the
           length of the attribute data in octets.  If the Extended
           Length attribute flag is set, the length field is 2
           octets, otherwise it is 1 octet";
        reference
          "RFC 4271: A Border Gateway Protocol 4 (BGP-4).";
      }

      leaf attr-value {
        type binary {
          length "0..65535";
        }
        description
          "Raw attribute value, not including the attribute
           flags, type, or length.  The maximum length
           of the attribute value data is 2^16-1 per the max value
           of the attr-len field (2 octets).";
        reference
          "RFC 4271: A Border Gateway Protocol 4 (BGP-4).";
      }
    }
  }
}
```

```
}

grouping bgp-adj-rib-attr-state {
  description
    "Path attributes that are not expected to be shared across
    route entries, specific to Adj-RIB";
  leaf path-id {
    type uint32;
    description
      "When the BGP speaker supports advertisement of multiple
      paths for a prefix, the path identifier is used to
      uniquely identify a route based on the combination of the
      prefix and path id. In the Adj-RIB-In, the path-id value is
      the value received in the update message. In the Loc-RIB,
      if used, it should represent a locally generated path-id
      value for the corresponding route. In Adj-RIB-Out, it
      should be the value sent to a neighbor when add-paths is
      used, i.e., the capability has been negotiated.";
    reference
      "RFC 7911: Advertisement of Multiple Paths in BGP.";
  }
}

grouping bgp-community-attr-state {
  description
    "Common definition of BGP community attributes";
  leaf-list community {
    type union {
      type bct:bgp-well-known-community-type;
      type bct:bgp-std-community-type;
    }
    description
      "List of standard or well-known BGP community
      attributes.";
  }
}

grouping ext-community-attributes {
  description
    "A grouping for all extended community parameters.";

  leaf-list ext-community {
    type bct:bgp-ext-community-type;
    description
      "List of BGP extended community attributes. The received
      extended community may be an explicitly modeled
      type or unknown, represented by an 8-octet value
      formatted according to RFC 4360.";
  }
}
```

```
    reference
      "RFC 4360: BGP Extended Communities Attribute.";
  }

  leaf-list ext-community-raw {
    type string {
      // raw with 8 octets
      pattern
        'raw:([0-9A-F][0-9A-F]){7}[0-9A-F][0-9A-F]';
    }
    description
      "ext-community type in raw format only.";
  }
}

grouping ipv6-ext-community-attributes {
  description
    "A grouping for all IPv6 extended community parameters.";

  leaf-list ipv6-ext-community {
    type bct:bgp-ipv6-ext-community-type;
    description
      "List of BGP IPv6 extended community attributes. The received
      IPv6 extended community may be an explicitly modeled
      type or unknown, represented by an 20-octet value
      formatted according to RFC 5701.";
    reference
      "RFC 5701: IPv6 Address Specific BGP Extended Community
      Attribute, Section 3.";
  }

  leaf-list ipv6-ext-community-raw {
    type string {
      // raw with 20 octets
      pattern
        'ipv6-raw: '
        + '([0-9A-Fa-f][0-9A-Fa-f]){19}'
        + '[0-9A-Fa-f][0-9A-Fa-f]';
    }
    description
      "IPv6 ext-community type in raw format only.";
  }
}

grouping large-community-attributes {
  description
    "A grouping for all large community parameters.";
```



```
leaf-list large-community {  
  type bct:bgp-large-community-type;  
  description  
    "List of BGP large community attributes.";  
  reference  
    "RFC 8092: BGP Large Communities Attribute.";  
}  
}  
<CODE ENDS>
```

Figure 5: BMP YANG 'ietf-bmp-bgp-rib-attributes' submodule.

5. YANG Message Broker BMP Indexing and Topic Naming

To identify which network node produced which BMP data into which Message Broker Topic, Partition and Subject, BMP Message Keys and Indexes (Section 5.1) are being introduced. These keys enable a deterministic distribution of BMP messages accross Topics and Partitions enabling applications to consume only the needed data from specific topics and partitions.

In order to facilitate Message Broker Topic Compaction, a BMP subscription type based topic naming scheme (Section 5.2) is defined. This segregates statistical (Value), State and State change BMP metrics and facilitates a BMP Message Broker Consumer to use the Topic wild card consumption method to select based on BMP subscription type.

5.1. BMP Message Keys and Indexes

A Message Broker uses a Message Key to index the Message and a value to carry the Message content. If no Message Key is defined then the Messages are distributed in a round robin fashion across partitions. If a Message Key is defined, then the value of the Message Key is being used as input for the Message Broker Producer hash function to distribute across Partitions. Therefore, Message Keys facilitate Message deterministic distribution.

The Message Key not only used for Message indexing at the Message Producer but also at the Message Broker for topic compaction.

For BMP, the network node hostname, what BMP Message Type the data reprints and apart from the initiation-message and termination-message the BMP index is used to generate the Message Key. The BMP index is calculated depending on BMP Message Type as following:

- * peer-up-notification: "peer-distinguisher" and "peer-address".

- * peer-down-notification: "peer-distinguisher" and "peer-address".
- * statistics: "peer-distinguisher" and "peer-address".
- * route-monitoring: "peer-distinguisher", "peer-address" afi-safi name, address family, name of the BGP RIB, "prefix" "origin" "path-id" route attributes and the index of the used attr-set.
- * route-mirroring: "peer-distinguisher", "peer-address" afi-safi name, address family, name of the BGP RIB, "prefix" "origin" "path-id" route attributes, index of the used attr-set and information enumeration.

5.2. Message Broker BMP Topic Naming

BMP data can be subscribed for BMP session and BGP peering session state-changes, BGP peering statistics or BGP RIB states. as defined in Section 3.1 of [RFC7854].

Message Brokers topics are addressed with a unique name. Usually topics are named hierarchically similar to the DNS namespace where "." delimitates hierarchies.

This document defines "statistics", "states" and "state-changes" in the topic name as the first part to denote the types of data. Followed by "bmp" to denote BMP data. Followed by the BGP RIB type and the BMP message type name.

For example, if BMP route-monitoring is being subscribed to BGP Local-RIB, the Message Broker topic name would be as following. In the example the project name and environment (prod, dev, test etc.) is prefixed.

project.environment.state.bmp.local-rib.route-monitoring

Figure 6: BMP BGP Local-RIB route-monitoring Topic Name Example

5.2.1. YANG Message Broker Producer

For the Message Broker topic creation, the "message-type" contained data in "bmp-subscription" from "ietf-bmp-telemetry-message", YANG module defined in Figure 2, notifications MUST be used to derive wherever subscribed BMP data is "statistics" "states" or "state-changes".

5.2.2. YANG Message Broker Consumer

The consumer has the ability to consume with a wildcard denoted with "*" in the topic name to consume from more than one topic.

For example, if BMP states should be consumed and indexed in Time Series database or stream processor than below Topic Name could be used, and the BMP data could be ingested into tables according to topic names and indexed per Message Key. If Topic Compaction is enabled, only current state is consumed.

```
project.environment.states.bmp.*
```

Figure 7: BMP Wildcard Topic Name Example

6. IANA Considerations

This document registers the following two namespace URIs in the IETF XML Registry [RFC3688]:

- * URI: urn:ietf:params:xml:ns:yang:ietf-bmp-telemetry-message
- * Registrant Contact: The IESG.
- * XML: N/A; the requested URI is an XML namespace.
- * URI: urn:ietf:params:xml:ns:yang:ietf-bmp-bgp-rib-entry
- * Registrant Contact: The IESG.
- * XML: N/A; the requested URI is an XML namespace.
- * URI: urn:ietf:params:xml:ns:yang:ietf-bmp-bgp-rib-tables
- * Registrant Contact: The IESG.
- * XML: N/A; the requested URI is an XML namespace.
- * URI: urn:ietf:params:xml:ns:yang:ietf-bmp-bgp-rib-attributes
- * Registrant Contact: The IESG.
- * XML: N/A; the requested URI is an XML namespace.

This document registers the following two YANG modules in the YANG Module Names registry [RFC3688]:

- * Name: ietf-bmp-telemetry-message

- * Namespace: urn:ietf:params:xml:ns:yang:ietf-bmp-telemetry-message
- * Prefix: bmptm
- * Reference: RFC XXXX

- * Name: ietf-bmp-bgp-rib-entry
- * Namespace: urn:ietf:params:xml:ns:yang:ietf-bmp-bgp-rib-entry
- * Prefix: bmptm
- * Reference: RFC XXXX

- * Name: ietf-bmp-bgp-rib-tables
- * Namespace: urn:ietf:params:xml:ns:yang:ietf-bmp-bgp-rib-tables
- * Prefix: bmptm
- * Reference: RFC XXXX

- * Name: ietf-bmp-bgp-rib-attributes
- * Namespace: urn:ietf:params:xml:ns:yang:ietf-bmp-bgp-rib-attributes
- * Prefix: bmptm
- * Reference: RFC XXXX

7. Security Considerations

This section is modeled after the template described in Section 3.7 of [I-D.ietf-netmod-rfc8407bis].

The "ietf-bmp-telemetry-message" YANG module and "ietf-bmp-bgp-rib-entry", "ietf-bmp-bgp-rib-tables" and "ietf-bmp-bgp-rib-attributes" YANG submodules defines one combined data model that are designed to be accessed via YANG-based management protocols, such as NETCONF [RFC6141] and RESTCONF [RFC8040]. These protocols have to use a secure transport layer (e.g., SSH [RFC4252], TLS [RFC8446], and QUIC [RFC9000]) and have to use mutual authentication.

The Network Configuration Access Control Model (NACM) [RFC8341] provides the means to restrict access for particular NETCONF or RESTCONF users to a preconfigured subset of all available NETCONF or RESTCONF protocol operations and content.

There are a number of data nodes defined in this YANG module that are writable/creatable/deletable (i.e., "config true", which is the default). All writable data nodes are likely to be reasonably sensitive or vulnerable in some network environments. Write operations (e.g., edit-config) and delete operations to these data nodes without proper protection or authentication can have a negative effect on network operations. The following subtrees and data nodes have particular sensitivities/vulnerabilities:

"There are no particularly sensitive writable data nodes."

Some of the readable data nodes in this YANG module may be considered sensitive or vulnerable in some network environments. It is thus important to control read access (e.g., via get, get-config, or notification) to these data nodes. Specifically, the following subtrees and data nodes have particular sensitivities/vulnerabilities:

"There are no particularly sensitive readable data nodes."

8. References

8.1. Normative References

- [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>>.
- [RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <<https://www.rfc-editor.org/info/rfc3688>>.
- [RFC4252] Ylonen, T. and C. Lonvick, Ed., "The Secure Shell (SSH) Authentication Protocol", RFC 4252, DOI 10.17487/RFC4252, January 2006, <<https://www.rfc-editor.org/info/rfc4252>>.
- [RFC6141] Camarillo, G., Ed., Holmberg, C., and Y. Gao, "Re-INVITE and Target-Refresh Request Handling in the Session Initiation Protocol (SIP)", RFC 6141, DOI 10.17487/RFC6141, March 2011, <<https://www.rfc-editor.org/info/rfc6141>>.
- [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>>.

- [RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <<https://www.rfc-editor.org/info/rfc7950>>.
- [RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <<https://www.rfc-editor.org/info/rfc8040>>.
- [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>>.
- [RFC8341] Bierman, A. and M. Bjorklund, "Network Configuration Access Control Model", STD 91, RFC 8341, DOI 10.17487/RFC8341, March 2018, <<https://www.rfc-editor.org/info/rfc8341>>.
- [RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <<https://www.rfc-editor.org/info/rfc8446>>.
- [RFC8671] Evens, T., Bayraktar, S., Lucente, P., Mi, P., and S. Zhuang, "Support for Adj-RIB-Out in the BGP Monitoring Protocol (BMP)", RFC 8671, DOI 10.17487/RFC8671, November 2019, <<https://www.rfc-editor.org/info/rfc8671>>.
- [RFC9000] Iyengar, J., Ed. and M. Thomson, Ed., "QUIC: A UDP-Based Multiplexed and Secure Transport", RFC 9000, DOI 10.17487/RFC9000, May 2021, <<https://www.rfc-editor.org/info/rfc9000>>.
- [RFC9069] Evens, T., Bayraktar, S., Bhardwaj, M., and P. Lucente, "Support for Local RIB in the BGP Monitoring Protocol (BMP)", RFC 9069, DOI 10.17487/RFC9069, February 2022, <<https://www.rfc-editor.org/info/rfc9069>>.
- [I-D.ietf-nmop-terminology]
Davis, N., Farrel, A., Graf, T., Wu, Q., and C. Yu, "Some Key Terms for Network Fault and Problem Management", Work in Progress, Internet-Draft, draft-ietf-nmop-terminology-23, 18 August 2025, <<https://datatracker.ietf.org/doc/html/draft-ietf-nmop-terminology-23>>.
- [I-D.ietf-nmop-yang-message-broker-integration]
Graf, T. and A. Elhassany, "An Architecture for YANG-Push to Message Broker Integration", Work in Progress, Internet-Draft, draft-ietf-nmop-yang-message-broker-

integration-10, 18 January 2026,
<<https://datatracker.ietf.org/doc/html/draft-ietf-nmop-yang-message-broker-integration-10>>.

[I-D.ietf-nmop-message-broker-telemetry-message]

Elhassany, A., Graf, T., and P. Lucente, "Extensible YANG Model for Network Telemetry Messages", Work in Progress, Internet-Draft, draft-ietf-nmop-message-broker-telemetry-message-04, 18 January 2026,
<<https://datatracker.ietf.org/doc/html/draft-ietf-nmop-message-broker-telemetry-message-04>>.

[I-D.netana-nmop-yang-message-broker-message-key]

Graf, T., Elhassany, A., Feng, A. H., Claise, B., and P. Lucente, "YANG Message Keys for Message Broker Integration", Work in Progress, Internet-Draft, draft-netana-nmop-yang-message-broker-message-key-03, 24 January 2026, <<https://datatracker.ietf.org/doc/html/draft-netana-nmop-yang-message-broker-message-key-03>>.

8.2. Informative References

[I-D.ietf-idr-bgp-model]

Jethanandani, M., Patel, K., Hares, S., and J. Haas, "YANG Model for Border Gateway Protocol (BGP-4)", Work in Progress, Internet-Draft, draft-ietf-idr-bgp-model-18, 21 October 2024, <<https://datatracker.ietf.org/doc/html/draft-ietf-idr-bgp-model-18>>.

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

Cardona, C., Lucente, P., Graf, T., Claise, B., Patki, D., and N. Prasad, "BMP YANG Module", Work in Progress, Internet-Draft, draft-ietf-grow-bmp-yang-07, 20 October 2025, <<https://datatracker.ietf.org/doc/html/draft-ietf-grow-bmp-yang-07>>.

[I-D.ietf-netmod-rfc8407bis]

Bierman, A., Boucadair, M., and Q. Wu, "Guidelines for Authors and Reviewers of Documents Containing YANG Data Models", Work in Progress, Internet-Draft, draft-ietf-netmod-rfc8407bis-28, 5 June 2025,
<<https://datatracker.ietf.org/doc/html/draft-ietf-netmod-rfc8407bis-28>>.

[Deh22]

Dehghani, Z., "Data Mesh", O'Reilly Media, ISBN 9781492092391, March 2022,
<<https://www.oreilly.com/library/view/data-mesh/9781492092384/>>.

- [Kaf11] Narkhede, N., "Apache Kafka", Apache Software Foundation, January 2011, <<https://kafka.apache.org/>>.
- [Pul16] Guo, S. and M. Merli, "Apache Pulsar", Apache Software Foundation, January 2016, <<https://pulsar.apache.org/>>.

Acknowledgements

Thanks to xxx for their comments and reviews.

Contributors

Many thanks goes to the authors of [I-D.ietf-idr-bgp-model] from which the BGP RIB submodules defined in Section 7.6 of [I-D.ietf-idr-bgp-model] have been obtained and adjusted for BMP integration.

Mahesh Jethanandani
Arrcus
CA
United States of America
Email: mjethanandani@gmail.com

Keyur Patel
Arrcus
CA
United States of America
Email: keyur@arrcus.com

Susan Hares
Huawei
7453 Hickory Hill
Saline, MI 48176
United States of America
Email: shares@ndzh.com

Jeffrey Haas
Juniper Networks
Email: jhaas@pfr.org

Authors' Addresses

Thomas Graf
Swisscom
Binzring 17
CH-8045 Zurich
Switzerland
Email: thomas.graf@swisscom.com

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

Leonardo Rodoni
Swisscom
Binzring 17
CH-8045 Zurich
Switzerland
Email: leonardo.rodoni@swisscom.com

Maxence Younsi
INSA-Lyon
Lyon
France
Email: maxence.younsi@insa-lyon.fr