

GROW  
Internet-Draft  
Intended status: Standards Track  
Expires: 23 April 2026

C. Cardona  
P. Lucente  
NTT  
T. Graf  
Swisscom  
B. Claise  
Everything OPS  
D. Patki  
P. Narasimha  
Cisco  
20 October 2025

BMP YANG Module  
draft-ietf-grow-bmp-yang-07

Abstract

This document proposes a YANG module for the configuration and monitoring of the BGP Monitoring Protocol (BMP).

Status of This Memo

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

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

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

This Internet-Draft will expire on 23 April 2026.

Copyright Notice

Copyright (c) 2025 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components

extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

## Table of Contents

1. Introduction . . . . .	3
2. Terminology . . . . .	3
3. Model description . . . . .	3
3.1. IP Connectivity . . . . .	3
3.1.1. Active connection . . . . .	4
3.1.2. Passive connection . . . . .	4
3.2. TCP Options . . . . .	5
3.3. Other BMP connectivity options . . . . .	8
3.4. BMP data . . . . .	9
3.4.1. BMP route monitoring . . . . .	10
3.4.1.1. Network instances . . . . .	13
3.4.1.2. RIB Type . . . . .	17
3.4.1.3. Address families . . . . .	18
3.4.1.4. Peers . . . . .	22
3.4.1.5. Filtering route-monitoring messages . . . . .	23
3.4.1.6. Full examples of Route monitoring configurations . . . . .	23
3.5. Session stats . . . . .	30
3.6. Session reset action . . . . .	30
4. Implementation guidelines . . . . .	30
5. BMP YANG module . . . . .	30
5.1. TCP dependencies for BMP YANG Module . . . . .	30
6. Security Considerations . . . . .	33
6.1. Security Considerations for ietf-bmp module . . . . .	33
6.2. Security Considerations for ietf-bmp-tcp-dependencies module . . . . .	34
7. IANA Considerations . . . . .	35
8. Open Issues . . . . .	35
9. References . . . . .	35
9.1. Normative References . . . . .	35
9.2. Informative References . . . . .	37
Appendix A. BMP YANG model tree . . . . .	38
Appendix B. Base BMP YANG Module . . . . .	57
Appendix C. Examples . . . . .	80
C.1. Example one . . . . .	80
C.2. Example two . . . . .	81
Acknowledgements . . . . .	84
Authors' Addresses . . . . .	84

## 1. Introduction

The BGP Monitoring Protocol (BMP) provides a mechanism for monitoring BGP sessions. This document defines a YANG data model for BMP, enabling operators to configure and manage BMP sessions; control the data exported to monitoring stations; and collect statistics for each BMP session. The model is designed to accommodate both simple and advanced deployment scenarios, supporting granular control over network instances, RIBs, address families, and peers.

## 2. Terminology

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

Routing Information Bases, peers, monitoring stations, and initiation messages are defined in [RFC7854].

## 3. Model description

This document specifies a YANG module for configuring and monitoring the BGP Monitoring Protocol (BMP) [RFC7854] on the monitored router. The model provides parameters for configuring the session to BMP monitoring stations; configuration of BMP messages; provides operational metrics and enables resetting of BMP monitoring sessions.

The model is included in Section 5. In this section, we provide details and examples of each of its parts.

The BMP yang model is placed at the root of the YANG tree. At its upper level, the BMP model lists each monitoring station. Every monitoring station is identified by an ID, which is a string provided by the operator.

### 3.1. IP Connectivity

BMP allows for active and passive connections between the router and the BMP monitoring station as described in Section 3.2 of [RFC7854]. In an active connection, the router establishes the TCP connection to the monitoring station, while in a passive one, it is the monitoring station which initiates the connection. The BMP YANG module provides options for both types of connection using a choice.

We describe each type of connection option next, and provide examples of their configuration.

### 3.1.1.1. Active connection

For an active connection, the IP address and port of the monitoring station, together with the local endpoint must be provided. The local endpoint can be a local IP address or a source interface. One can optionally provide the local port for establishing the connection. If the monitoring station is connected over a network-instance instead of the global one, this one must also be specified. An example of configuration is included in Figure 1.

===== NOTE: '\ ' line wrapping per RFC 8792 =====

```
<?xml version="1.0"?>
<bmp xmlns="urn:ietf:params:xml:ns:yang:ietf-bmp">
  <monitoring-stations>
    <monitoring-station>
      <id>1</id>
      <connection>
        <active>
          <station-address>192.0.2.1</station-address>
          <station-port>57992</station-port>
          <monitored-router-address>192.0.2.2</monitored-router-addr\
ess>
        </active>
      </connection>
    </monitoring-station>
  </monitoring-stations>
</bmp>
```

Figure 1: Active connection example

Note in the example from Figure 1 that there is no network instance defined, so the connection is using the global network instance.

### 3.1.1.2. Passive connection

In a passive connection, the IP of the monitoring station, the local endpoint, and the local port for the incoming connection must be specified. If the port of the monitoring station is provided, it must match the incoming connection. If the monitoring station is connected through a network-instance instead of the global one, this one must also be specified.

An incoming connection not matching a valid entry MUST be ignored by the router.

Figure 2 includes an example of configuring a passive connection. In this example, we specify the network-instance where we want to receive the connection.

===== NOTE: '\ ' line wrapping per RFC 8792 =====

```
<?xml version="1.0"?>
<bmp xmlns="urn:ietf:params:xml:ns:yang:ietf-bmp">
  <monitoring-stations>
    <monitoring-station>
      <id>monitoring_station_two</id>
      <connection>
        <passive>
          <network-instance>test</network-instance>
          <station-address>192.0.2.1</station-address>
          <monitored-router-address>192.0.2.2</monitored-router-addr\
ess>
          <monitored-router-port>57993</monitored-router-port>
        </passive>
      </connection>
    </monitoring-station>
  </monitoring-stations>
</bmp>
```

Figure 2: Passive connection example

### 3.2. TCP Options

The BMP module allows tuning various parameters of the TCP connection supporting the BMP session:

- \* The maximum segment size of the TCP connection. See Section 3.7.1 of [RFC9293].
- \* Enabling MTU discovery for the path. See Section 3.7.2 of [RFC9293].
- \* For configuring TCP keepalives, the connection container uses the tcp-common-grouping from [RFC9643]. Note that the implementation must support module ietf-bmp-tcp-dependencies in addition to the main module to support these elements. Please see Section 2.1.3.1 of [RFC9643] for the explanation of each of its parameters. The device must have the feature "tcp-client-keepalives" enabled. See also Section 3.8.4 of [RFC9293]
- \* Session security. Provides options for authentication using AO and MD5. This part of the model was taken from the BGP YANG model [I-D.ietf-idr-bgp-model]. This feature also requires support of the ietf-bmp-tcp-dependencies module.

Figures 3 and 4 include examples configuring the previous TCP parameters in the model.

===== NOTE: '\ ' line wrapping per RFC 8792 =====

```
<?xml version="1.0"?>
<bmp xmlns="urn:ietf:params:xml:ns:yang:ietf-bmp">
  <monitoring-stations>
    <monitoring-station>
      <id>1</id>
      <connection>
        <active>
          <station-address>192.0.2.1</station-address>
          <station-port>57992</station-port>
          <monitored-router-address>192.0.2.2</monitored-router-addr\
ess>
        </active>
        <tcp-options>
          <keepalives xmlns="urn:ietf:params:xml:ns:yang:ietf-bmp-tc\
p-dependencies">
            <idle-time>15</idle-time>
            <max-probes>3</max-probes>
            <probe-interval>30</probe-interval>
          </keepalives>
            <maximum-segment-size>1500</maximum-segment-size>
            <mtu-discovery>true</mtu-discovery>
          </tcp-options>
        </connection>
      </monitoring-station>
    </monitoring-stations>
  </bmp>
```

Figure 3: Example of configuring basic TCP parameters

===== NOTE: '\ ' line wrapping per RFC 8792 =====

```
<key-chains xmlns="urn:ietf:params:xml:ns:yang:ietf-key-chain">
  <key-chain>
    <name>bmp-key-chain</name>
    <description>An example of TCP-AO configuration for BMP</descr\
ption>
    <key>
      <key-id>55</key-id>
      <crypto-algorithm>aes-cmac-prf-128</crypto-algorithm>
      <lifetime>
        <send-lifetime>
          <start-date-time>2023-01-01T00:00:00+00:00</start-date-t\
ime>
```

```

        <end-date-time>2023-02-01T00:00:00+00:00</end-date-time>
      </send-lifetime>
      <accept-lifetime>
        <start-date-time>2023-01-01T00:00:00+00:00</start-date-t\
ime>
        <end-date-time>2023-02-01T00:00:00+00:00</end-date-time>
      </accept-lifetime>
    </lifetime>
    <key-string>
      <keystring>teststring</keystring>
    </key-string>
    <authentication xmlns="urn:ietf:params:xml:ns:yang:ietf-tcp">
      <keychain>bmp-key-chain</keychain>
      <ao>
        <send-id>65</send-id>
        <recv-id>87</recv-id>
      </ao>
    </authentication>
  </key>
  <key>
    <key-id>56</key-id>
    <crypto-algorithm>aes-cmac-prf-128</crypto-algorithm>
    <lifetime>
      <send-lifetime>
        <start-date-time>2023-01-01T00:00:00+00:00</start-date-t\
ime>
        <end-date-time>2023-02-01T00:00:00+00:00</end-date-time>
      </send-lifetime>
      <accept-lifetime>
        <start-date-time>2023-01-01T00:00:00+00:00</start-date-t\
ime>
        <end-date-time>2023-02-01T00:00:00+00:00</end-date-time>
      </accept-lifetime>
    </lifetime>
    <authentication xmlns="urn:ietf:params:xml:ns:yang:ietf-tcp">
      <keychain>bmp-key-chain</keychain>
      <ao>
        <send-id>65</send-id>
        <recv-id>87</recv-id>
      </ao>
    </authentication>
  </key>
</key-chain>
</key-chains>
<bmp xmlns="urn:ietf:params:xml:ns:yang:ietf-bmp">
  <monitoring-stations>
    <monitoring-station>
      <id>monitoring_station_one</id>

```

```
<connection>
  <active>
    <station-address>192.0.2.1</station-address>
    <station-port>57992</station-port>
    <monitored-router-address>192.0.2.2</monitored-router-ad\
dress>
  </active>
  <tcp-options>
    <secure-session xmlns="urn:ietf:params:xml:ns:yang:ietf-\
bmp-tcp-dependencies">
      <ao-keychain>bmp-key-chain</ao-keychain>
    </secure-session>
  </tcp-options>
</connection>
</monitoring-station>
</monitoring-stations>
</bmp>
```

Figure 4: Example of the Configuration of TCP session security.

### 3.3. Other BMP connectivity options

The model also includes the following options to configure the connection to the BMP monitoring station:

- \* Initial-delay: a value in seconds that the device must wait before starting the connection to the station. An operator can use this delay to let BGP converge before starting the BMP session.
- \* Backoff time: Configuration of the backoff time strategy after failing to connect to the monitoring station. The model includes a basic exponential backoff with a default initial backoff of 30 seconds and a maximum of 720 seconds, as suggested in Section 3.2 of [RFC7854].

In the example in Figure 5, we configure an initial-delay of 10, an initial-backoff of 50 seconds and 600 of maximum-backoff.



===== NOTE: '\ ' line wrapping per RFC 8792 =====

```
<?xml version="1.0"?>
<bmp xmlns="urn:ietf:params:xml:ns:yang:ietf-bmp">
  <monitoring-stations>
    <monitoring-station>
      <id>1</id>
      <connection>
        <active>
          <station-address>192.0.2.1</station-address>
          <station-port>57992</station-port>
          <monitored-router-address>192.0.2.2</monitored-router-addr\
ess>
        </active>
        <initial-delay>10</initial-delay>
        <backoff>
          <simple-exponential>
            <initial-backoff>50</initial-backoff>
            <maximum-backoff>600</maximum-backoff>
          </simple-exponential>
        </backoff>
      </connection>
    </monitoring-station>
  </monitoring-stations>
</bmp>
```

Figure 5: Example of the initial-delay and simple exponential backoff.

### 3.4. BMP data

The `bmp-data` container defines the configuration parameters for the data that the device sends to the monitoring station using BMP messages. See Section 4 of [RFC7854].

The BMP model defines options for the initiation message, the statistics report, the route mirroring, and the route monitoring. The first three have simple configuration options and are described next. Route monitoring is the most complex and is detailed in Section 3.4.1.

- \* Initiation-message: Content for an information TLV type-0 for identification of the device. See 4.3 and Section 4.4 of [RFC7854]

- \* Statistics-interval: The statistics report is enabled by the presence of the statistics-report container. The statistics-interval is mandatory if the statistics-report container exists and defines the interval of the statistics report. See Section 4.8 of [RFC7854].
- \* Route-mirroring: Route Mirroring messages serve as an exact replica of the messages received by the device. See Section 6 of [RFC7854]. Enabling route mirroring messages towards a particular BMP monitoring station only requires the presence of the container "route-mirroring" within the monitoring station container.

An example of configuring the previous options is included in Figure 6

===== NOTE: '\' line wrapping per RFC 8792 =====

```
<?xml version="1.0"?>
<bmp xmlns="urn:ietf:params:xml:ns:yang:ietf-bmp">
  <monitoring-stations>
    <monitoring-station>
      <id>monitoring_station_one</id>
      <connection>
        <active>
          <station-address>192.0.2.1</station-address>
          <station-port>57992</station-port>
          <monitored-router-address>192.0.2.2</monitored-router-addr\
ess>
        </active>
      </connection>
      <bmp-data>
        <initiation-message>BMP device supporting the BMP yang modul\
e</initiation-message>
        <statistics-report>
          <statistics-interval>600</statistics-interval>
        </statistics-report>
      </bmp-data>
    </monitoring-station>
  </monitoring-stations>
</bmp>
```

Figure 6: Example of configuration of initiation-message and statistics report interval.

### 3.4.1. BMP route monitoring

Route monitoring messages are used for synchronization of RIBs to the monitoring station. See Section 5 of [RFC7854].

The next 3 requirements were defined before designing this part of the model.

- \* Operators might not want to receive all routes from all RIBs in a network device. For instance, some devices contain a considerable amount of data that might overwhelm the monitoring station. In these cases, operators might want to only collect information from an arbitrary subset of RIBs, address families, peers.
- \* Operators might want to configure the route monitoring messages for different network instances differently. For example, they might want to receive different address families from the global network instance than in L3 VPN network instances.
- \* In contrast to the previous points, some operators might want a simple configuration that covers multiple cases (e.g. same config for all peers, or same config for all network instances). This would not only make configurations look smaller and concise, but will reduce the need for reconfiguring devices when you add a new peer or add a new network instance (which happens frequently on some type of networks).

Based on the previous points, the BMP yang model is designed to flexibly control the data sent through the BMP route monitoring packets, yet it provides options to facilitate configurations for simple cases, such as when the operator wants to receive all routes from a RIB.

The route monitoring configuration is divided in a four-part hierarchy:

- \* Network Instance
- \* RIB Type (e.g. Adj-RIB-IN pre/post, local RIB)
- \* Address Family
- \* Peers

Absence of the route monitoring container will disable the route monitoring messages to the monitoring station.

We'll offer an introduction to these hierarchies before going over them with detail.

The number of RIB types (e.g. Adj-RIB-IN/OUT, local RIB, etc) and Address families is low, and their configuration should not change frequently. Therefore, they are configured explicitly in the model. That is, the model does not provide a way of providing a default configuration for these or configuring them in groups.

On the other hand, Network instances and peers require greater flexibility.

For network instances, the model should configure not only the "global" network instance, but also other network instances. Also, network instances can change frequently in networks with customer connecting to Virtual Private Networks. To not force operators to change configuration at every change, the model provides methods for defining a "default" configuration for network instances. However, to provide control over the configuration, each network instance can be configured independently, if needed.

A situation is similar with peers for the Adj-RIB-IN and Adj-RIB-OUT RIBs. The model includes a way of configuring a default for all peers for simple cases, but one can provide configuration for type of peers, peergroups, or each peer individually.

We summarize the requirements stated on the previous two paragraphs next:

For network instances:

- \* The configuration should be simple for cases where only the "global" routing instance is enabled.
- \* The model should provide ways of configuring all Network instances (kind of a default config for any Network instance that is configured in the device).
- \* The model should provide a way of configuring network instances individually.

For peers:

- \* The model should provide ways of configuring all peers, kind of a default. This would be the most common case.
- \* The model should provide ways of configuring peergroups.
- \* The model should provide ways of configuring type of peers. For instance, only send routes from eBGP peers.
- \* The model should provide ways of configuring individual peers. For instance, an operator might apply a route-policy to filter certain prefixes for a specific peer, or disable route monitoring messages for a peer that is noisy yet not important.

To further control the route monitoring data, the peer container includes a route-policy option in which the operator can further filter the data send to the BMP monitoring station.

We'll describe each of the 4 hierarchies, and provide examples for each, in the next sections.

#### 3.4.1.1. Network instances

The route monitoring configuration starts with the configuration of network instances. A network instance can be configured individually, or it can be configured if it matches any of the selectors from the "bmp-ni-types" identity. We explain each next.

The model currently defines three bmp-ni-types identities: "all-ni" which selects all network instances, "non-global-ni" which selects all network instances except the global one, and "global-ni", which configures the global network instance when the device does not offer an explicit name for it. The former can be used as a "default" configuration for simple cases.

Network-instances are configured under the container "bmp/route-monitoring/network-instance".

An empty configuration disables route monitoring messages for the selected network-instances. Operators can also use the "enable" leaf to disable explicitly the routing messages for the network instance.

The route-monitoring data for a network instance can be configured by at most one element under the network-instance-configuration container. There SHOULD be clear rules for which element to apply to a network instance in case multiple elements can select it. We provide rules and examples in the next part of the section.

The rules for selecting which element configures a network instance are presented next. Each point is evaluated only if the previous points do not hold.

- \* If the name of the network instance is referenced in "network-instance-configuration/network-instances/network-instance", the network instance SHOULD be configured using this element.
- \* If the selector 'global-ni' under the "network-instance-configuration/network-instance-selectors" exist, the global network instance SHOULD be configured using this element. Note that if a vendor has a name for the global network instance, the previous step (i.e, network instance name) will take priority over using the global-ni selector.
- \* If the selector 'non-global-ni' under the "network-instance-configuration/network-instance-selectors" exist, any non-global network instance should be configured using its content.
- \* Any Network instance not referenced by any rule above SHOULD be configured using the all-ni if one exists. If it does not exist, then the network instance is not configured (and therefore no route monitoring messages from the network instance are sent to the monitoring station).

Any extension of the bmp-ni-types SHOULD provide explanations of how to deal with case in which multiple elements select the same network instance.

We provide examples of configuring the network instance level next. For now, we will focus on the configuration using the BMP container. To focus on the network instance configuration, we mask the configuration under each instance using "Configuration X".

===== NOTE: '\ ' line wrapping per RFC 8792 =====

```
<?xml version="1.0"?>
<bmp xmlns="urn:ietf:params:xml:ns:yang:ietf-bmp">
  <monitoring-stations>
    <monitoring-station>
      <id>monitoring_station_one</id>
      <connection>
        <active>
          <station-address>192.0.2.1</station-address>
          <station-port>57992</station-port>
          <monitored-router-address>192.0.2.2</monitored-router-addr\
ess>
        </active>
      </connection>
    </monitoring-station>
  </monitoring-stations>
  <bmp-data>
    <route-monitoring>
      <network-instance-configuration>
        <network-instance-selectors>
          <network-instance-selector>
            <id>all-ni</id>
            <!-- Configuration A -->
          </network-instance-selector>
          <network-instance-selector>
            <id>global-ni</id>
            <!-- Configuration B -->
          </network-instance-selector>
        </network-instance-selectors>
        <network-instances>
          <network-instance>
            <id>network-instance-two</id>
            <!-- Configuration C -->
          </network-instance>
          <network-instance>
            <id>network-instance-one</id>
            <enabled>false</enabled>
          </network-instance>
        </network-instances>
      </network-instance-configuration>
    </route-monitoring>
  </bmp-data>
</bmp>
```

Figure 7: Examples of configuring the network instance level for Route Monitoring.

In example from Figure 7, we have a "default" configuration (Configuration A) applied to any network instance without any explicit configuration. The global network instance and network-instance-two get Configuration B and Configuration C, respectively. The network-instance-one instance container disables the route monitoring messages for that network instance.

===== NOTE: '\\' line wrapping per RFC 8792 =====

```
<?xml version="1.0"?>
<bmp xmlns="urn:ietf:params:xml:ns:yang:ietf-bmp">
  <monitoring-stations>
    <monitoring-station>
      <id>monitoring_station_one</id>
      <connection>
        <active>
          <station-address>192.0.2.1</station-address>
          <station-port>57992</station-port>
          <monitored-router-address>192.0.2.2</monitored-router-addr\
ess>
        </active>
      </connection>
    </monitoring-station>
  </monitoring-stations>
  <bmp-data>
    <route-monitoring>
      <network-instance-configuration>
        <network-instance-selectors>
          <network-instance-selector>
            <id>all-ni</id>
            <!-- Configuration D -->
          </network-instance-selector>
        </network-instance-selectors>
      </network-instance-configuration>
    </route-monitoring>
  </bmp-data>
</bmp>
```

Figure 8: Example of configuring all network instances.

The example in Figure 8 shows a "simple" configuration. In this case, all network instances would get "Configuration D". Note that 'all-ni' would also cover the global instance.

Another simple configuration would just involve configuring the global network instance. In this case, information of non-global network instances would not be sent to the monitoring station. This is depicted in Figure 9



```

===== NOTE: '\ ' line wrapping per RFC 8792 =====

<?xml version="1.0"?>
<bmp xmlns="urn:ietf:params:xml:ns:yang:ietf-bmp">
  <monitoring-stations>
    <monitoring-station>
      <id>monitoring_station_one</id>
      <connection>
        <active>
          <station-address>192.0.2.1</station-address>
          <station-port>57992</station-port>
          <monitored-router-address>192.0.2.2</monitored-router-addr\
ess>
        </active>
      </connection>
    </monitoring-station>
  </monitoring-stations>
  <bmp-data>
    <route-monitoring>
      <network-instance-configuration>
        <network-instance-selectors>
          <network-instance-selector>
            <id>global-ni</id>
            <!-- configuration E -->
          </network-instance-selector>
        </network-instance-selectors>
      </network-instance-configuration>
    </route-monitoring>
  </bmp-data>
</bmp>

```

Figure 9: Example of configuring only the global network instance.

#### 3.4.1.2. RIB Type

Each RIB type is configured explicitly in the model through a container. The model currently provides containers for adj-rib-out-pre, adj-rib-out-post, adj-rib-in-post, adj-rib-in-pre and local-rib.

An empty configuration or absence of a RIB-type container disables route-messages for it. Operators can also disable the route monitoring messages for each RIB explicitly by marking the "enabled" leaf as False.

We provide an example of this, together with address families, in the next section

### 3.4.1.3. Address families

Address families are configured explicitly within each RIB type using a list. The key is of type 'ietf-bgp-types:afi-safi-type' without any further constraint.

An empty configuration or absence of an address family disables route-messages for it. Operators can also disable the address-family route monitoring messages by marking the "enabled" leaf as False.

We show a few examples of configuring RIB-Types and Address families next. We will mask further configurations of address families with "Configuration X" to focus on the covered parts.

===== NOTE: '\ ' line wrapping per RFC 8792 =====

```
<?xml version="1.0"?>
<bmp xmlns="urn:ietf:params:xml:ns:yang:ietf-bmp">
  <monitoring-stations>
    <monitoring-station>
      <id>monitoring_station_one</id>
      <connection>
        <active>
          <station-address>192.0.2.1</station-address>
          <station-port>57992</station-port>
          <monitored-router-address>192.0.2.2</monitored-router-addr\
ess>
        </active>
      </connection>
    </monitoring-station>
  </monitoring-stations>
  <bmp-data>
    <route-monitoring>
      <network-instance-configuration>
        <network-instance-selectors>
          <network-instance-selector>
            <id>all-ni</id>
            <adj-rib-in-pre>
              <address-families>
                <address-family>
                  <id>ipv6-unicast</id>
                  <!-- Configuration F -->
                </address-family>
                <address-family>
                  <id>ipv4-unicast</id>
                  <!-- Configuration G -->
                </address-family>
              </address-families>
            </adj-rib-in-pre>
          </network-instance-selector>
        </network-instance-selectors>
      </network-instance-configuration>
    </route-monitoring>
  </bmp-data>
</bmp>
```

```
<network-instance-selector>
  <id>global-ni</id>
  <adj-rib-in-pre>
    <address-families>
      <address-family>
        <id>ipv6-unicast</id>
        <!-- Configuration H -->
      </address-family>
      <address-family>
        <id>ipv4-unicast</id>
        <!-- Configuration I -->
      </address-family>
    </address-families>
  </adj-rib-in-pre>
  <adj-rib-in-post>
    <address-families>
      <address-family>
        <id>ipv6-unicast</id>
        <!-- Configuration H -->
      </address-family>
      <address-family>
        <id>ipv4-unicast</id>
        <!-- Configuration I -->
      </address-family>
    </address-families>
  </adj-rib-in-post>
</network-instance-selector>
</network-instance-selectors>
<network-instances>
  <network-instance>
    <id>network-instance-one</id>
    <enabled>false</enabled>
  </network-instance>
  <network-instance>
    <id>network-instance-two</id>
    <adj-rib-in-post>
      <address-families>
        <address-family>
          <id>ipv4-unicast</id>
          <!-- Configuration L -->
        </address-family>
      </address-families>
    </adj-rib-in-post>
  </network-instance>
</network-instances>
</network-instance-configuration>
</route-monitoring>
</bmp-data>
```

```
    </monitoring-station>
  </monitoring-stations>
</bmp>
```

Figure 10: Example of configuring RIBs and address families.

In Figure 10, we expand previous sections examples with RIB-Type and address families configurations. The expected result of the previous configuration would be:

- \* For the global network instance, adj-rib-in-pre and adj-rib-in-post RIBs are enabled. In each of them IPv4 and IPv6 address families are configured. The configuration can be the same or not, depending on the requirements of the operators. Any other RIB and address families are disabled.
- \* Network instance "network-instance-one" is disabled, meaning that route monitoring messages are disabled for that network instance.
- \* Network instance "network-instance-two" has adj-rib-out-post enabled, but only address family ipv4-unicast is configured. The ipv6-unicast will not be configured for this instance.
- \* For all other network instances, adj-rib-in-pre with IPv4 and IPv6 address families are configured, thanks to the configuration of all-ni

If an operator only wants to configure the IPv4/IPv6 of adj-rib-pre-in for the global instance, the configuration in Figure 11 plus the peer configuration (coming in next section) will be enough. We note again that even if the configuration of both address families is the same, they must be explicitly configured for each of them.

===== NOTE: '\ ' line wrapping per RFC 8792 =====

```
<?xml version="1.0"?>
<bmp xmlns="urn:ietf:params:xml:ns:yang:ietf-bmp">
  <monitoring-stations>
    <monitoring-station>
      <id>monitoring_station_one</id>
      <connection>
        <active>
          <station-address>192.0.2.1</station-address>
          <station-port>57992</station-port>
          <monitored-router-address>192.0.2.2</monitored-router-addr\
ess>
        </active>
      </connection>
    <bmp-data>
      <route-monitoring>
        <network-instance-configuration>
          <network-instance-selectors>
            <network-instance-selector>
              <id>global-ni</id>
              <adj-rib-in-pre>
                <address-families>
                  <address-family>
                    <id>ipv6-unicast</id>
                    <!-- Configuration for ipv6-unicast -->
                  </address-family>
                  <address-family>
                    <id>ipv4-unicast</id>
                    <!-- Configuration for ipv4-unicast -->
                  </address-family>
                </address-families>
              </adj-rib-in-pre>
            </network-instance-selector>
          </network-instance-selectors>
        </network-instance-configuration>
      </route-monitoring>
    </bmp-data>
  </monitoring-station>
</monitoring-stations>
</bmp>
```

Figure 11: Example of configuring RIBs and address families.

#### 3.4.1.4. Peers

For adj-RIB-in and adj-RIB-out, both pre and post, the model requires the selection of peer RIBs that will be transmitted to the monitoring station. The local-rib does not include this container.

Peers can be configured using different "selectors", which can be one of the following:

- \* An individual peer, using a remote address. For the configuration under the "/bmp" tree, the model currently does not check if the remote address exists, that would be a responsibility of the device.
- \* Peergroups.
- \* A group of peers matching a BGP type. i.e. eBGP peers.
- \* One or more peers defined by an 'bmp-peer-types' identity. The BMP model currently provides the 'all-peers' identity which select all peers. For simple cases, this is the value that would normally be considered.

Peers MUST be selected (configured) by at most a single instance of the peers list. For the included keys in the BMP model, the process to select which instance to use is as follows:

- \* If there is a peer address matching the peer, it should be configured using that instance.
- \* If the peer matches a peergroup, it should be configured using the peer-group configuration.
- \* If the peer is of any BGP type listed in the peer list, it should be configured using this instance.
- \* If there is a peer instance identified with the 'all-peers', it would be configured using this instance.
- \* Finally, if no instance covers the peer, route monitoring messages from this peer should not be transmitted to the monitoring station.

An empty configuration of a peer type disables route-messages for it. Operators can also disable the address-family route monitoring messages by marking the "enabled" leaf as False.

Note that if an operator only wants the information of a few peers, it can enable them individually using their id. If no other configuration exists, only the messages from those enabled peers will be transmitted to the monitoring station.

Any additional bmp-peer-types identity created SHOULD describe how to unambiguously select a peer when there are conflicting options (multiple options covering the peer).

We'll provide examples of the peers configuration after describing the filter containers.

#### 3.4.1.5. Filtering route-monitoring messages

The local rib, and the peer containers within the rest of rib types, include a filter container. This container includes mechanisms to filter route-monitoring messages for the specific RIB.

The policy-filter can include a routing policy that, if existing, should be applied to the outgoing updates to the monitoring station, and would serve as a granular way of filtering the messages that the monitoring station receives.

Note that the policy-filter contains an 'accept-route' default export policy. An operator can change it to a reject-route, if required.

The policies created with the routing-policy can perform a large variety of actions on routes, and can filter them based on multiple characteristics. For the consistency of the data in the monitoring station, the route policies actions MUST be restricted to accepting or rejecting routes. Furthermore, the conditions SHOULD only match prefix sets.

We present examples of full configurations next.

#### 3.4.1.6. Full examples of Route monitoring configurations

##### 3.4.1.6.1. Example one - simple configuration

In the example configuration from Figure 12, address families IPv6 and IPv4 are configured to send all peers from the global network instance. This is an example of a simple configuration

===== NOTE: '\ ' line wrapping per RFC 8792 =====

```
<?xml version="1.0"?>
<bmp xmlns="urn:ietf:params:xml:ns:yang:ietf-bmp">
  <monitoring-stations>
    <monitoring-station>
      <id>monitoring_station_one</id>
      <connection>
        <active>
          <station-address>192.0.2.1</station-address>
          <station-port>57992</station-port>
          <monitored-router-address>192.0.2.2</monitored-router-addr\
ess>
        </active>
```

```
</connection>
<bmp-data>
  <route-monitoring>
    <network-instance-configuration>
      <network-instance-selectors>
        <network-instance-selector>
          <id>global-ni</id>
          <adj-rib-in-pre>
            <address-families>
              <address-family>
                <id>ipv6-unicast</id>
                <peers-configurations>
                  <peer-selectors>
                    <peer-selector>
                      <id>all-peers</id>
                    </peer-selector>
                  </peer-selectors>
                </peers-configurations>
              </address-family>
              <address-family>
                <id>ipv4-unicast</id>
                <peers-configurations>
                  <peer-selectors>
                    <peer-selector>
                      <id>all-peers</id>
                    </peer-selector>
                  </peer-selectors>
                </peers-configurations>
              </address-family>
            </address-families>
          </adj-rib-in-pre>
        </network-instance-selector>
      </network-instance-selectors>
    </network-instance-configuration>
  </route-monitoring>
</bmp-data>
</monitoring-station>
</monitoring-stations>
</bmp>
```

Figure 12: Enabling Route monitoring for all peers in the global network instance; IPv4/IPv6 Address families, in the adj-rib-in-pre RIB.



## 3.4.1.6.2. Example two - policy list example

In the example in Figure 13, the global network instance enables the adj-rib-in-pre. In this RIB, the IPv4 unicast address family is configured for all external peers. We assume peer 198.51.100.1 is external, but its BGP configuration is not shown in the snippet. Peer 198.51.100.1, however, has a specific configuration: it announces everything but prefixes matching the test\_policy list. Note that there is a default accept-route default policy in the model.

===== NOTE: '\ ' line wrapping per RFC 8792 =====

```
<?xml version="1.0"?>
<routing-policy xmlns="urn:ietf:params:xml:ns:yang:ietf-routing-policy">
  <policy-definitions>
    <policy-definition>
      <name>test_policy</name>
      <!-- Policy definition -->
    </policy-definition>
  </policy-definitions>
</routing-policy>
<bmp xmlns="urn:ietf:params:xml:ns:yang:ietf-bmp">
  <monitoring-stations>
    <monitoring-station>
      <id>monitoring_station_one</id>
      <connection>
        <active>
          <station-address>192.0.2.1</station-address>
          <station-port>57992</station-port>
          <monitored-router-address>192.0.2.2</monitored-router-address>
        </active>
      </connection>
    </monitoring-station>
  </monitoring-stations>
  <bmp-data>
    <route-monitoring>
      <network-instance-configuration>
        <network-instance-selectors>
          <network-instance-selector>
            <id>global-ni</id>
            <adj-rib-in-pre>
              <address-families>
                <address-family>
                  <id>ipv6-unicast</id>
                  <peers-configurations>
                    <peer-selectors>
                      <peer-selector>
```

```

        <id>all-peers</id>
      </peer-selector>
    </peer-selectors>
  </peers-configurations>
</address-family>
<address-family>
  <id>ipv4-unicast</id>
  <peers-configurations>
    <peer-types>
      <peer-type>
        <id>external</id>
      </peer-type>
    </peer-types>
    <peers>
      <peer>
        <id>198.51.100.1</id>
        <filters>
          <policy-filter>
            <export-policy>test_policy</export-policy>
          </policy-filter>
        </filters>
      </peer>
    </peers>
  </peers-configurations>
</address-family>
</address-families>
</adj-rib-in-pre>
</network-instance-selector>
</network-instance-selectors>
</network-instance-configuration>
</route-monitoring>
</bmp-data>
</monitoring-station>
</monitoring-stations>
</bmp>
y>

```

Figure 13: Configuring address families differently for the global network instance

#### 3.4.1.6.3. Example three - specific network instance configuration

In the example from Figure 14, all network instances have adj-rib-in-pre with IPv6 and IPv4 configured receiving all peers. network-instance-one is disabled, and network-instance-two is announcing only the local-rib/IPv4 unicast routes.

===== NOTE: '\ ' line wrapping per RFC 8792 =====

```
<?xml version="1.0"?>
<bmp xmlns="urn:ietf:params:xml:ns:yang:ietf-bmp">
  <monitoring-stations>
    <monitoring-station>
      <id>monitoring_station_one</id>
      <connection>
        <active>
          <station-address>192.0.2.1</station-address>
          <station-port>57992</station-port>
          <monitored-router-address>192.0.2.2</monitored-router-addr\
ess>
        </active>
      </connection>
    </monitoring-station>
  </monitoring-stations>
  <bmp-data>
    <route-monitoring>
      <network-instance-configuration>
        <network-instance-selectors>
          <network-instance-selector>
            <id>all-ni</id>
            <adj-rib-in-pre>
              <address-families>
                <address-family>
                  <id>ipv6-unicast</id>
                  <peers-configurations>
                    <peer-selectors>
                      <peer-selector>
                        <id>all-peers</id>
                      </peer-selector>
                    </peer-selectors>
                  </peers-configurations>
                </address-family>
                <address-family>
                  <id>ipv4-unicast</id>
                  <peers-configurations>
                    <peer-selectors>
                      <peer-selector>
                        <id>all-peers</id>
                      </peer-selector>
                    </peer-selectors>
                  </peers-configurations>
                </address-family>
              </address-families>
            </adj-rib-in-pre>
          </network-instance-selector>
        </network-instance-selectors>
      </network-instance-configuration>
    </route-monitoring>
  </bmp-data>
</bmp>
```

```

    <network-instance>
      <id>network-instance-one</id>
      <enabled>>false</enabled>
    </network-instance>
    <network-instance>
      <id>network-instance-two</id>
      <local-rib>
        <address-families>
          <address-family>
            <id>ipv4-unicast</id>
          </address-family>
        </address-families>
      </local-rib>
    </network-instance>
  </network-instances>
</network-instance-configuration>
</route-monitoring>
</bmp-data>
</monitoring-station>
</monitoring-stations>
</bmp>

```

Figure 14: Applying a general configuration to all network instances, except of two, which are configured specifically.

#### 3.4.1.6.4. Example four - Enabling just a few peers in global

In the example from Figure 15, we configure the device to only send to the `monitoring_station_one` monitoring station the Route monitoring messages for ipv4 and ipv6 from peers "198.51.100.1" and "198.51.100.2" in the `adj-rib-in-pre` policy.

===== NOTE: '\ ' line wrapping per RFC 8792 =====

```

<?xml version="1.0"?>
<bmp xmlns="urn:ietf:params:xml:ns:yang:ietf-bmp">
  <monitoring-stations>
    <monitoring-station>
      <id>monitoring_station_one</id>
      <connection>
        <active>
          <station-address>192.0.2.1</station-address>
          <station-port>57992</station-port>
          <monitored-router-address>192.0.2.2</monitored-router-addr\
ess>
        </active>
      </connection>
    </monitoring-station>
  </monitoring-stations>
</bmp>

```

```
<route-monitoring>
  <network-instance-configuration>
    <network-instance-selectors>
      <network-instance-selector>
        <id>global-ni</id>
        <adj-rib-in-pre>
          <address-families>
            <address-family>
              <id>ipv6-unicast</id>
              <peers-configurations>
                <peers>
                  <peer>
                    <id>198.51.100.1</id>
                  </peer>
                  <peer>
                    <id>198.51.100.2</id>
                  </peer>
                </peers>
              </peers-configurations>
            </address-family>
            <address-family>
              <id>ipv4-unicast</id>
              <peers-configurations>
                <peers>
                  <peer>
                    <id>198.51.100.1</id>
                  </peer>
                  <peer>
                    <id>198.51.100.2</id>
                  </peer>
                </peers>
              </peers-configurations>
            </address-family>
          </address-families>
        </adj-rib-in-pre>
      </network-instance-selector>
    </network-instance-selectors>
  </network-instance-configuration>
</route-monitoring>
</bmp-data>
</monitoring-station>
</monitoring-stations>
</bmp>
```

Figure 15: Sending just BGP messages from adj-rib-in-pre from 2 peers

### 3.5. Session stats

The non-configurable container "session-stats" includes various metrics for the session with the monitoring station.

### 3.6. Session reset action

The "session-reset" action resets a session with a monitoring station.

## 4. Implementation guidelines

To facilitate implementation, the model is divided into two distinct parts. The core model (ietf-bmp) is designed to configure BMP without deep dependencies to other modules. This is an attempt to facilitate the implementation by vendors.

Additionally, a supplementary module, ietf-bmp-tcp-dependencies.yang, enhances the model's functionality but rely on the IETF TCP models. This module can be adopted by implementations that support the dependency.

## 5. BMP YANG module

### 5.1. TCP dependencies for BMP YANG Module

```
<CODE BEGINS> file "ietf-bmp-tcp-dependencies.yang@2022-01-27.yang"
module ietf-bmp-tcp-dependencies {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-bmp-tcp-dependencies";
  prefix bmp-tcp;

  import ietf-bmp {
    prefix bmp;
  }
  import ietf-tcp-common {
    prefix tcpcmn;
    reference
      "RFC 9643: YANG Groupings for TCP
       Clients and TCP Servers.";
  }
  import ietf-key-chain {
    prefix key-chain;
    reference
      "RFC 8177: YANG Key Chain.";
  }

  organization
```

```
"IETF GROW Working Group";
contact
  "WG Web:    <https://datatracker.ietf.org/wg/grow/>
  WG List:    <mailto:grow@ietf.org>

  Author:     Camilo Cardona
              <mailto:camilo@ntt.net>

  Author:     Paolo Lucente
              <mailto:cpaolo@ntt.net>

  Author:     Thomas Graf
              <mailto:thomas.graf@swisscom.com>

  Author:     Benoit Claise
              <mailto:benoit.claise@huawei.com>";
description
  "This module specifies a structure for BMP
  (BGP Monitoring Protocol) configuration and monitoring.

  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) 2025 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.

  ";

revision 2025-01-02 {
  description
    "initial version";
  reference
    "RFC YYYY: BMP YANG Module
    RFC-EDITOR: please update YYYY with this RFC ID";
```

```
}

augment "/bmp:bmp/bmp:monitoring-stations/"
+ "bmp:monitoring-station/"
+ "bmp:connection/bmp:tcp-options" {
  description
    "Augment the tcp options of the BMP model";
  uses tcpcmn:tcp-common-grouping;
  container secure-session {
    presence "Means the session should be secure. ";
    description
      "Container for describing how a particular BMP session
      is to be secured.";
    choice authentication {
      mandatory true;
      description
        "Choice of TCP authentication.";
      case ao {
        description
          "Uses TCP-AO to secure the session. Parameters for
          those are defined as a grouping in the TCP YANG
          model.";
        reference
          "RFC 5925 - The TCP Authentication Option.";
        leaf ao-keychain {
          type key-chain:key-chain-ref;
          description
            "Reference to the key chain that will be used by
            this model. Applicable for TCP-AO and TCP-MD5
            only";
          reference
            "RFC 8177: YANG Key Chain.";
        }
      }
      case md5 {
        description
          "Uses TCP-MD5 to secure the session. Parameters for
          those are defined as a grouping in the TCP YANG
          model.";
        reference
          "RFC 5925: The TCP Authentication Option.";
        leaf md5-keychain {
          type key-chain:key-chain-ref;
          description
            "Reference to the key chain that will be used by
            this model. Applicable for TCP-AO and TCP-MD5
            only";
          reference
```





Some of the RPC or action operations in this YANG module may be considered sensitive or vulnerable in some network environments. It is thus important to control access to these operations. Specifically, the following operations have particular sensitivities/vulnerabilities: The session-reset action can demand a considerable amount of resources from network elements. It SHOULD thus be protected from unauthorized access.

## 6.2. Security Considerations for ietf-bmp-tcp-dependencies module

This section is modeled after the template described in Section 3.7.1 of [I-D.ietf-netmod-rfc8407bis].

The ietf-bmp-tcp-dependencies YANG module defines a data model that is designed to be accessed via YANG-based management protocols, such as NETCONF [RFC6241] and RESTCONF [RFC8040]. These protocols have to use a secure transport layer (e.g., SSH [RFC6242], 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. 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. There are no particularly sensitive readable data nodes.

This YANG module uses groupings from other YANG modules that define nodes that may be considered sensitive or vulnerable in network environments. Refer to the Security Considerations of [RFC9643] for information as to which nodes may be considered sensitive or vulnerable in network environments.

## 7. IANA Considerations

IANA is requested to register the following URI in the "ns" registry within the "IETF XML Registry" group [RFC3688]:

URI: urn:ietf:params:xml:ns:yang:ietf-bmp  
Registrant Contact: The IESG.  
XML: N/A; the requested URI is an XML namespace.

URI: urn:ietf:params:xml:ns:yang:ietf-bmp-tcp-dependencies  
Registrant Contact: The IESG.  
XML: N/A; the requested URI is an XML namespace.

IANA is requested to register the following YANG module in the "YANG Module Names" registry [RFC6020] within the "YANG Parameters" registry group.

Name: ietf-bmp  
Maintained by IANA? N  
Namespace: urn:ietf:params:xml:ns:yang:ietf-bmp  
Prefix: bmp  
Reference: RFC-XXXX

Name: ietf-bmp-tcp-dependencies  
Maintained by IANA? N  
Namespace: urn:ietf:params:xml:ns:yang:ietf-bmp-tcp-dependencies  
Prefix: bmp-tcp  
Reference: RFC-XXXX

## 8. Open Issues

The security considerations section will have to be aligned with <https://trac.ietf.org/trac/ops/wiki/yang-security-guidelines>

## 9. References

### 9.1. Normative References

- [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>>.

- [RFC1191] Mogul, J. and S. Deering, "Path MTU discovery", RFC 1191, DOI 10.17487/RFC1191, November 1990, <<https://www.rfc-editor.org/info/rfc1191>>.
- [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>>.
- [RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <<https://www.rfc-editor.org/info/rfc6020>>.
- [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <<https://www.rfc-editor.org/info/rfc6241>>.
- [RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, DOI 10.17487/RFC6242, June 2011, <<https://www.rfc-editor.org/info/rfc6242>>.
- [RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types", RFC 6991, DOI 10.17487/RFC6991, July 2013, <<https://www.rfc-editor.org/info/rfc6991>>.
- [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>>.
- [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>>.
- [RFC8177] Lindem, A., Ed., Qu, Y., Yeung, D., Chen, I., and J. Zhang, "YANG Data Model for Key Chains", RFC 8177, DOI 10.17487/RFC8177, June 2017, <<https://www.rfc-editor.org/info/rfc8177>>.

- [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>>.
- [RFC8349] Lhotka, L., Lindem, A., and Y. Qu, "A YANG Data Model for Routing Management (NMDA Version)", RFC 8349, DOI 10.17487/RFC8349, March 2018, <<https://www.rfc-editor.org/info/rfc8349>>.
- [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>>.
- [RFC8529] Berger, L., Hopps, C., Lindem, A., Bogdanovic, D., and X. Liu, "YANG Data Model for Network Instances", RFC 8529, DOI 10.17487/RFC8529, March 2019, <<https://www.rfc-editor.org/info/rfc8529>>.
- [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>>.
- [RFC9293] Eddy, W., Ed., "Transmission Control Protocol (TCP)", STD 7, RFC 9293, DOI 10.17487/RFC9293, August 2022, <<https://www.rfc-editor.org/info/rfc9293>>.
- [RFC9643] Watsen, K. and M. Scharf, "YANG Groupings for TCP Clients and TCP Servers", RFC 9643, DOI 10.17487/RFC9643, October 2024, <<https://www.rfc-editor.org/info/rfc9643>>.

## 9.2. 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>>.

## Appendix A. BMP YANG model tree

The following tree diagram provides an overview of the base ietf-bmp.yang data model. It also includes the trees for the module ietf-bmp-tcp-dependencies.yang that adds some options to the models in case the implementation supports this model (and their requisites)

===== NOTE: '\ ' line wrapping per RFC 8792 =====

module: ietf-bmp

```

+--rw bmp
  +--rw monitoring-stations
    +--rw monitoring-station* [id]
      +--rw id string
      +--rw description? string
      +--rw connection
        +--rw (passive-or-active)
          +--:(active)
            +--rw active
              +--rw network-instance?
                | leafref
              +--rw station-address
                | inet:ip-address
              +--rw station-port
                | inet:port-number
              +--rw (local-endpoint)
                +--:(monitored-router-address)
                  | +--rw monitored-router-address?
                    | inet:ip-address
                +--:(monitored-router-interface)
                  | +--rw monitored-router-interface?
                    | if:interface-ref
              +--rw monitored-router-port?
                | inet:port-number
            +--:(passive)
              +--rw passive
                +--rw network-instance?
                  | leafref
                +--rw station-address
                  | inet:ip-address
                +--rw station-port?

```

```

|         inet:port-number
|         +--rw (local-endpoint)
|         |         +--:(monitored-router-address)
|         |         |         +--rw monitored-router-address?
|         |         |         |         inet:ip-address
|         |         +--:(monitored-router-interface)
|         |         |         +--rw monitored-router-interface?
|         |         |         |         if:interface-ref
|         +--rw monitored-router-port
|         |         inet:port-number
+--rw dscp?          inet:dscp
+--rw tcp-options
|   +--rw maximum-segment-size?      uint16
|   +--rw mtu-discovery?              boolean
|   +--rw bmp-tcp:keepalives!
|   |   {keepalives-supported}?
|   |   +--rw bmp-tcp:idle-time?      uint16
|   |   +--rw bmp-tcp:max-probes?     uint16
|   |   +--rw bmp-tcp:probe-interval? uint16
|   +--rw bmp-tcp:secure-session!
|   |   +--rw (bmp-tcp:authentication)
|   |   |   +--:(bmp-tcp:ao)
|   |   |   |   +--rw bmp-tcp:ao-keychain?
|   |   |   |   |   key-chain:key-chain-ref
|   |   |   +--:(bmp-tcp:md5)
|   |   |   |   +--rw bmp-tcp:md5-keychain?
|   |   |   |   |   key-chain:key-chain-ref
+--rw initial-delay?  uint32
+--rw backoff
|   +--rw (backoff-options)?
|   |   +--:(simple-exponential)
|   |   |   +--rw simple-exponential
|   |   |   |   +--rw initial-backoff?  uint32
|   |   |   |   +--rw maximum-backoff?  uint32
+--rw bmp-data
|   +--rw initiation-message?  string
|   +--rw statistics-report!
|   |   +--rw statistics-interval  uint32
+--rw route-monitoring
|   +--rw network-instance-configuration
|   |   +--rw network-instances
|   |   |   +--rw network-instance* [id]
|   |   |   |   +--rw id              leafref
|   |   |   |   +--rw enabled?        boolean
|   |   |   +--rw adj-rib-in-pre
|   |   |   |   +--rw address-families
|   |   |   |   |   +--rw address-family* [id]
|   |   |   |   |   |   +--rw id

```

						identityref
						+++rw enabled?
						boolean
						+++rw peers-configurations
						+++rw peers
						+++rw peer* [id]
						+++rw id
						string
						+++rw enabled?
						boolean
						+++rw filters
						+++rw policy-filter
						{bmp-filter-b\
ased-on-route-policy}?						
y*						+++rw export-polic\
						leafref
rt-policy?						+++rw default-expo\
ault-policy-type						rt-pol:def\
						+++rw peer-groups
						+++rw peer-group* [id]
						+++rw id
						string
						+++rw enabled?
						boolean
						+++rw filters
						+++rw policy-filter
						{bmp-filter-b\
ased-on-route-policy}?						
y*						+++rw export-polic\
						leafref
rt-policy?						+++rw default-expo\
ault-policy-type						rt-pol:def\
						+++rw peer-selectors
						+++rw peer-selector*
						[id]
						+++rw id
						identityref
						+++rw enabled?
						boolean
						+++rw filters
						+++rw policy-filter
						{bmp-filter-b\



```

ased-on-route-policy}}?
y*
rt-policy?
ault-policy-type
    +--rw peer-types
    +--rw peer-type* [id]
    +--rw id
    |   peer-type
    +--rw enabled?
    |   boolean
    +--rw filters
    +--rw policy-filter
    {bmp-filter-b\
ased-on-route-policy}}?
y*
rt-policy?
ault-policy-type
    +--rw adj-rib-in-post
    +--rw address-families
    +--rw address-family* [id]
    +--rw id
    |   identityref
    +--rw enabled?
    |   boolean
    +--rw peers-configurations
    +--rw peers
    +--rw peer* [id]
    +--rw id
    |   string
    +--rw enabled?
    |   boolean
    +--rw filters
    +--rw policy-filter
    {bmp-filter-b\
ased-on-route-policy}}?
y*
rt-policy?

```

					rt-pol:def\
a	u	l	t	-	p
o	l	i	c	y	-
t	y	p	e		
					+
					---
					rw
					peer-groups
					+
					---
					rw
					peer-group* [id]
					+
					---
					rw
					id
					string
					+
					---
					rw
					enabled?
					boolean
					+
					---
					rw
					filters
					+
					---
					rw
					policy-filter
					{bmp-filter-b\
a	s	e	d	-	o
n	-	r	o	u	t
e					-
					p
					o
					l
					y* }
					+
					---
					rw
					export-polic\
r	t	-	p	o	l
i	c	y	?		
					leafref
					+
					---
					rw
					default-expo\
a	u	l	t	-	p
o	l	i	c	y	-
t	y	p	e		
					rt-pol:def\
					+
					---
					rw
					peer-selectors
					+
					---
					rw
					peer-selector*
					[id]
					+
					---
					rw
					id
					identityref
					+
					---
					rw
					enabled?
					boolean
					+
					---
					rw
					filters
					+
					---
					rw
					policy-filter
					{bmp-filter-b\
a	s	e	d	-	o
n	-	r	o	u	t
e					-
					p
					o
					l
					y* }
					+
					---
					rw
					export-polic\
r	t	-	p	o	l
i	c	y	?		
					leafref
					+
					---
					rw
					default-expo\
a	u	l	t	-	p
o	l	i	c	y	-
t	y	p	e		
					rt-pol:def\
					+
					---
					rw
					peer-types
					+
					---
					rw
					peer-type* [id]
					+
					---
					rw
					id
					peer-type
					+
					---
					rw
					enabled?
					boolean
					+
					---
					rw
					filters
					+
					---
					rw
					policy-filter
					{bmp-filter-b\
a	s	e	d	-	o
n	-	r	o	u	t
e					-
					p
					o
					l
					y* }
					+
					---
					rw
					export-polic\

y*							leafref +--rw default-expo\
rt-policy?							rt-pol:def\
aunt-policy-type							
				+--rw local-rib			
				+--rw address-families			
				+--rw address-family* [id]			
				+--rw id		identityref	
				+--rw filters			
				+--rw policy-filter		{bmp-filter-based-on-r\	
oute-policy}?							
						+--rw export-policy*	
							leafref
						+--rw default-export-policy?	
						rt-pol:default-poli\	
cy-type							
				+--rw adj-rib-out-pre			
				+--rw address-families			
				+--rw address-family* [id]			
				+--rw id		identityref	
				+--rw enabled?		boolean	
				+--rw peers-configurations			
				+--rw peers			
					+--rw peer* [id]		
					+--rw id		string
					+--rw enabled?		boolean
					+--rw filters		
					+--rw policy-filter		{bmp-filter-b\
ased-on-route-policy}?							
						+--rw export-polic\	
y*							
							leafref
rt-policy?						+--rw default-expo\	
							rt-pol:def\
aunt-policy-type							
				+--rw peer-groups			
				+--rw peer-group* [id]			
				+--rw id			string

						+--rw enabled?
						boolean
						+--rw filters
						+--rw policy-filter
						{bmp-filter-b\
ased-on-route-policy}}?						
y*						+--rw export-polic\
						leafref
rt-policy?						+--rw default-expo\
aault-policy-type						rt-pol:def\
						+--rw peer-selectors
						+--rw peer-selector*
						[id]
						+--rw id
						identityref
						+--rw enabled?
						boolean
						+--rw filters
						+--rw policy-filter
						{bmp-filter-b\
ased-on-route-policy}}?						
y*						+--rw export-polic\
						leafref
rt-policy?						+--rw default-expo\
aault-policy-type						rt-pol:def\
						+--rw peer-types
						+--rw peer-type* [id]
						+--rw id
						peer-type
						+--rw enabled?
						boolean
						+--rw filters
						+--rw policy-filter
						{bmp-filter-b\
ased-on-route-policy}}?						
y*						+--rw export-polic\
						leafref
rt-policy?						+--rw default-expo\
aault-policy-type						rt-pol:def\

				+--rw adj-rib-out-post
				+--rw address-families
				+--rw address-family* [id]
				+--rw id
				identityref
				+--rw enabled?
				boolean
				+--rw peers-configurations
				+--rw peers
				+--rw peer* [id]
				+--rw id
				string
				+--rw enabled?
				boolean
				+--rw filters
				+--rw policy-filter
				{bmp-filter-b\
ased-on-route-policy}?				
y*				+--rw export-police\
				leafref
rt-policy?				+--rw default-expo\
aunt-policy-type				rt-pol:def\
				+--rw peer-groups
				+--rw peer-group* [id]
				+--rw id
				string
				+--rw enabled?
				boolean
				+--rw filters
				+--rw policy-filter
				{bmp-filter-b\
ased-on-route-policy}?				
y*				+--rw export-police\
				leafref
rt-policy?				+--rw default-expo\
aunt-policy-type				rt-pol:def\
				+--rw peer-selectors
				+--rw peer-selector*
				[id]
				+--rw id
				identityref
				+--rw enabled?

						boolean
						+--rw filters
						+--rw policy-filter
						{bmp-filter-b\
ased-on-route-policy}}?						
y*						+--rw export-polic\
						leafref
rt-policy?						+--rw default-expo\
ault-policy-type						rt-pol:def\
						+--rw peer-types
						+--rw peer-type* [id]
						+--rw id
						peer-type
						+--rw enabled?
						boolean
						+--rw filters
						+--rw policy-filter
						{bmp-filter-b\
ased-on-route-policy}}?						
y*						+--rw export-polic\
						leafref
rt-policy?						+--rw default-expo\
ault-policy-type						rt-pol:def\
						+--rw network-instance-selectors
						+--rw network-instance-selector* [id]
						+--rw id identityref
						+--rw enabled? boolean
						+--rw adj-rib-in-pre
						+--rw address-families
						+--rw address-family* [id]
						+--rw id
						identityref
						+--rw enabled?
						boolean
						+--rw peers-configurations
						+--rw peers
						+--rw peer* [id]
						+--rw id
						string
						+--rw enabled?
						boolean
						+--rw filters

					+--rw policy-filter {bmp-filter-b\
ased-on-route-policy}?					
y*					+--rw export-polic\
					leafref
rt-policy?					+--rw default-expo\
ault-policy-type					rt-pol:def\
					+--rw peer-groups
					+--rw peer-group* [id]
					+--rw id
					string
					+--rw enabled?
					boolean
					+--rw filters
					+--rw policy-filter
ased-on-route-policy}?					{bmp-filter-b\
y*					+--rw export-polic\
					leafref
rt-policy?					+--rw default-expo\
ault-policy-type					rt-pol:def\
					+--rw peer-selectors
					+--rw peer-selector*
					[id]
					+--rw id
					identityref
					+--rw enabled?
					boolean
					+--rw filters
					+--rw policy-filter
ased-on-route-policy}?					{bmp-filter-b\
y*					+--rw export-polic\
					leafref
rt-policy?					+--rw default-expo\
ault-policy-type					rt-pol:def\
					+--rw peer-types
					+--rw peer-type* [id]
					+--rw id

					peer-type
					+++rw enabled?
					boolean
					+++rw filters
					+++rw policy-filter
					{bmp-filter-b\
ased-on-route-policy}?					
y*					+++rw export-polic\
					leafref
rt-policy?					+++rw default-expo\
aault-policy-type					rt-pol:def\
					+++rw adj-rib-in-post
					+++rw address-families
					+++rw address-family* [id]
					+++rw id
					identityref
					+++rw enabled?
					boolean
					+++rw peers-configurations
					+++rw peers
					+++rw peer* [id]
					+++rw id
					string
					+++rw enabled?
					boolean
					+++rw filters
					+++rw policy-filter
					{bmp-filter-b\
ased-on-route-policy}?					
y*					+++rw export-polic\
					leafref
rt-policy?					+++rw default-expo\
aault-policy-type					rt-pol:def\
					+++rw peer-groups
					+++rw peer-group* [id]
					+++rw id
					string
					+++rw enabled?
					boolean
					+++rw filters
					+++rw policy-filter
					{bmp-filter-b\



```

ased-on-route-policy}}?
y*
rt-policy?
ault-policy-type
    +--rw peer-selectors
    |   +--rw peer-selector*
    |   |   [id]
    |   |   +--rw id
    |   |   |   identityref
    |   |   +--rw enabled?
    |   |   |   boolean
    |   |   +--rw filters
    |   |   |   +--rw policy-filter
    |   |   |   |   {bmp-filter-b\
ased-on-route-policy}}?
y*
rt-policy?
ault-policy-type
    +--rw peer-types
    |   +--rw peer-type* [id]
    |   |   +--rw id
    |   |   |   peer-type
    |   |   +--rw enabled?
    |   |   |   boolean
    |   |   +--rw filters
    |   |   |   +--rw policy-filter
    |   |   |   |   {bmp-filter-b\
ased-on-route-policy}}?
y*
rt-policy?
ault-policy-type
    +--rw local-rib
    |   +--rw address-families
    |   |   +--rw address-family* [id]
    |   |   |   +--rw id
    |   |   |   |   identityref
    |   |   |   +--rw filters

```

				<pre>       +---rw policy-filter               {bmp-filter-based-on-r\ </pre>
oute-policy}}?				<pre>       +---rw export-policy*               leafref       +---rw default-export-policy?               rt-pol:default-poli\ </pre>
cy-type				<pre> +---rw adj-rib-out-pre +---rw address-families +---rw address-family* [id] +---rw id         identityref +---rw enabled?         boolean +---rw peers-configurations +---rw peers         +---rw peer* [id]                 +---rw id                         string                 +---rw enabled?                         boolean         +---rw filters                 +---rw policy-filter                         {bmp-filter-b\ </pre>
ased-on-route-policy}}?				<pre>       +---rw export-polic\ </pre>
y*				<pre>               leafref       +---rw default-expo\ </pre>
rt-policy?				<pre>       rt-pol:def\ </pre>
ault-policy-type				<pre> +---rw peer-groups +---rw peer-group* [id] +---rw id         string +---rw enabled?         boolean +---rw filters +---rw policy-filter         {bmp-filter-b\ </pre>
ased-on-route-policy}}?				<pre>       +---rw export-polic\ </pre>
y*				<pre>               leafref       +---rw default-expo\ </pre>
rt-policy?				<pre> </pre>

				rt-pol:def\
a- ult-policy-type				
				+--rw peer-selectors
				+--rw peer-selector*
				[id]
				+--rw id
				identityref
				+--rw enabled?
				boolean
				+--rw filters
				+--rw policy-filter
				{bmp-filter-b\
ased-on-route-policy}?				
y*				+--rw export-polic\
				leafref
rt-policy?				+--rw default-expo\
a- ult-policy-type				rt-pol:def\
				+--rw peer-types
				+--rw peer-type* [id]
				+--rw id
				peer-type
				+--rw enabled?
				boolean
				+--rw filters
				+--rw policy-filter
				{bmp-filter-b\
ased-on-route-policy}?				
y*				+--rw export-polic\
				leafref
rt-policy?				+--rw default-expo\
a- ult-policy-type				rt-pol:def\
				+--rw adj-rib-out-post
				+--rw address-families
				+--rw address-family* [id]
				+--rw id
				identityref
				+--rw enabled?
				boolean
				+--rw peers-configurations
				+--rw peers
				+--rw peer* [id]
				+--rw id

						string
						+++rw enabled?
						boolean
						+++rw filters
						+++rw policy-filter
						{bmp-filter-b\
ased-on-route-policy}?						
y*						+++rw export-polic\
rt-policy?						leafref
						+++rw default-expo\
aault-policy-type						rt-pol:def\
						+++rw peer-groups
						+++rw peer-group* [id]
						+++rw id
						string
						+++rw enabled?
						boolean
						+++rw filters
						+++rw policy-filter
						{bmp-filter-b\
ased-on-route-policy}?						
y*						+++rw export-polic\
rt-policy?						leafref
						+++rw default-expo\
aault-policy-type						rt-pol:def\
						+++rw peer-selectors
						+++rw peer-selector*
						[id]
						+++rw id
						identityref
						+++rw enabled?
						boolean
						+++rw filters
						+++rw policy-filter
						{bmp-filter-b\
ased-on-route-policy}?						
y*						+++rw export-polic\
rt-policy?						leafref
						+++rw default-expo\
						rt-pol:def\

[illegible]

		<pre> +--ro address-family* [id]   +--ro id       identityref   +--ro enabled?       boolean   +--ro total-route-monitoring-upda\ </pre>
ted-prefixes-per-af?		
draw-prefixes-per-af?		<pre>     uint64 +--ro total-route-monitoring-with\ </pre>
		<pre>     uint64 +--ro peers-stats   +--ro peer* [id]     +--ro id         string     +--ro enabled?         boolean     +--ro total-route-mnt-updat\ </pre>
ed-prefixes-per-peer?		
raw-prefixes-per-peer?		<pre>     uint64 +--ro total-route-mnt-withd\ </pre>
		<pre>     uint64 +--ro adj-rib-in-post   +--ro enabled?       boolean   +--ro total-route-mirroring-messages-pe\ </pre>
r-rib?		
		<pre>     uint64 +--ro address-families   +--ro address-family* [id]     +--ro id         identityref     +--ro enabled?         boolean     +--ro total-route-monitoring-upda\ </pre>
ted-prefixes-per-af?		
draw-prefixes-per-af?		<pre>     uint64 +--ro total-route-monitoring-with\ </pre>
		<pre>     uint64 +--ro peers-stats   +--ro peer* [id]     +--ro id         string     +--ro enabled?         boolean     +--ro total-route-mnt-updat\ </pre>

ed-prefixes-per-peer?				uint64
				+++ro total-route-mnt-withd\
raw-prefixes-per-peer?				uint64
				+++ro local-rib
				+++ro enabled?
				boolean
r-rib?				+++ro total-route-mirroring-messages-pe\
				uint64
				+++ro address-families
				+++ro address-family* [id]
				+++ro id
				identityref
				+++ro enabled?
				boolean
ted-prefixes-per-af?				+++ro total-route-monitoring-upda\
				uint64
draw-prefixes-per-af?				+++ro total-route-monitoring-with\
				uint64
				+++ro adj-rib-out-pre
				+++ro enabled?
				boolean
r-rib?				+++ro total-route-mirroring-messages-pe\
				uint64
				+++ro address-families
				+++ro address-family* [id]
				+++ro id
				identityref
				+++ro enabled?
				boolean
ted-prefixes-per-af?				+++ro total-route-monitoring-upda\
				uint64
draw-prefixes-per-af?				+++ro total-route-monitoring-with\
				uint64
				+++ro peers-stats
				+++ro peer* [id]
				+++ro id
				string
				+++ro enabled?
				boolean
				+++ro total-route-mnt-updat\

```

ed-prefixes-per-peer?
|
|                               |         uint64
|                               |--ro total-route-mnt-withd\
raw-prefixes-per-peer?
|
|                               |         uint64
|                               +--ro adj-rib-out-post
|                               |         boolean
|                               +--ro total-route-mirroring-messages-pe\
r-rib?
|
|                               |         uint64
|                               +--ro address-families
|                               |         +--ro address-family* [id]
|                               |         |         ++ro id
|                               |         |         |         identityref
|                               |         |         +--ro enabled?
|                               |         |         |         boolean
|                               |         +--ro total-route-monitoring-upda\
ted-prefixes-per-af?
|
|                               |         uint64
|                               +--ro total-route-monitoring-with\
draw-prefixes-per-af?
|
|                               |         uint64
|                               +--ro peers-stats
|                               |         +--ro peer* [id]
|                               |         |         ++ro id
|                               |         |         |         string
|                               |         |         +--ro enabled?
|                               |         |         |         boolean
|                               |         +--ro total-route-mnt-updat\
ed-prefixes-per-peer?
|
|                               |         uint64
|                               +--ro total-route-mnt-withd\
raw-prefixes-per-peer?
|
|                               |         uint64
|                               +--rw actions
|               +---x session-reset
|               |       +--ro output
|               |       |       +--ro (outcome)?
|               |       |       |       +--:(success)
|               |       |       |       |       +--ro success?     empty
|               |       |       |       +--:(failure)
|               |       |       |       |       +--ro failure?      string
|               +---x session-counter-reset
|               |       +--ro output
|               |       |       +--ro (outcome)?
|               |       |       |       +--:(success)
|               |       |       |       |       +--ro success?      empty

```



```
    +--:(failure)
      +--ro failure?   string
```

## Appendix B. Base BMP YANG Module

```
<CODE BEGINS> file "ietf-bmp@2022-01-27.yang"
module ietf-bmp {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-bmp";
  prefix bmp;

  import ietf-yang-types {
    prefix yang;
  }
  import ietf-inet-types {
    prefix inet;
    reference
      "RFC 6991: Common YANG Data Types";
  }
  import ietf-netconf-acm {
    prefix nacm;
    reference
      "RFC 8341: Network Configuration Access Control Model";
  }
  import ietf-routing-policy {
    prefix rt-pol;
    description
      "This module is only needed if the feature
       bmp-filter-based-on-route-policy is set";
    reference
      "RFC 9067: A YANG Data Model for Routing Policy";
  }
  import ietf-network-instance {
    prefix ni;
    reference
      "RFC 8529: YANG Data Model for Network Instances";
  }
  import ietf-interfaces {
    prefix if;
    reference
      "RFC 8343: A YANG Data Model for Interface Management";
  }

  organization
    "IETF GROW Working Group";
  contact
    "WG Web:   <https://datatracker.ietf.org/wg/grow/>
    WG List:  <mailto:grow@ietf.org>
```

Author: Camilo Cardona  
<mailto:camilo@ntt.net>

Author: Paolo Lucente  
<mailto:cpaolo@ntt.net>

Author: Thomas Graf  
<mailto:thomas.graf@swisscom.com>

Author: Benoit Claise  
<mailto:benoit.claise@huawei.com>

Author: Dhananjay Patki  
<mailto:dhpatki@cisco.com>

Author: Prasad S. Narasimha  
<mailto:snprasad@cisco.com>";

description

"This module defines a YANG data model for configuration and monitoring of the BGP Monitoring Protocol (BMP).

Copyright (c) 2025 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 2025-01-02 {

description

"initial version";

reference

"RFC YYYY: BMP YANG Module

RFC-EDITOR: please update YYYY with this RFC ID";

```
}

feature bmp-filter-based-on-route-policy {
  description
    "This feature means that the device
    is capable of filtering prefixes in BMP monitoring sessions";
}

/* The next enums are temporary here until we resolve how to deal
 * with them. Taken from draft-ietf-idr-bgp-model */
/* BGP AFI-SAFI Type Identities. */

identity afi-safi-type {
  description
    "Base identity type for AFI,SAFI tuples for BGP-4";
  reference
    "RFC4760: Multiprotocol Extensions for BGP-4.";
}

identity ipv4-unicast {
  base afi-safi-type;
  description
    "IPv4 unicast (AFI,SAFI = 1,1)";
  reference
    "RFC4760: Multiprotocol Extensions for BGP-4.";
}

identity ipv6-unicast {
  base afi-safi-type;
  description
    "IPv6 unicast (AFI,SAFI = 2,1)";
  reference
    "RFC4760: Multiprotocol Extensions for BGP-4.";
}

/* End of temporal objects */

identity bmp-peer-selectors {
  description
    "Generic Identity for selecting peers";
}

identity all-peers {
  base bmp-peer-selectors;
  description
    "This identity selects all peers under the RIB.
    When used, it acts as a default configuration.";
}
```

```
identity bmp-ni-types {
  description
    "Identities for selecting one or more network instances for
    configuration";
}

identity all-ni {
  base bmp-ni-types;
  description
    "This identity is an explicit way
    to select all network instances.";
}

identity global-ni {
  base bmp-ni-types;
  description
    "Identity for Selecting the global or main network instance";
}

identity non-global-ni {
  base bmp-ni-types;
  description
    "This identity is an explicit way
    to select all network instances except the global one.";
}

/* The next types are temporary here until we resolve how to deal
 * with them. Taken from draft-ietf-idr-bgp-model */
/* BGP Peer-Types */

typedef peer-type {
  type enumeration {
    enum internal {
      description
        "Internal (IBGP) peer";
    }
    enum external {
      description
        "External (EBGP) peer";
    }
    enum confederation-internal {
      description
        "Confederation Internal (IBGP) peer.";
    }
    enum confederation-external {
      description
        "Confederation External (EBGP) peer.";
    }
  }
}
```

```
    }
    description
      "Labels a peer or peer group as explicitly internal,
       external, or the related confederation type.";
    reference
      "RFC 4271: A Border Gateway Protocol 4 (BGP-4), Sec 1.1.
       RFC 5065: Autonomous System Configuration for BGP.";
  }

/* End of temporal objects */

grouping bmp-ip-connection-grouping {
  description
    "Common parameters for establishing connectivity
     to a BMP monitoring station.";
  choice passive-or-active {
    mandatory true;
    description
      "Selects whether the device initiates
       (active) or waits for (passive)
       the connection to the monitoring station,
       as described in RFC 7854 Section 3.2.";
    case active {
      description
        "Device initiates the connection to
         the monitoring station.";
      container active {
        description
          "The device starts the connection to
           the monitoring station";
        leaf network-instance {
          type leafref {
            path "/ni:network-instances/ni:network-instance/"
              + "ni:name";
          }
          description
            "Network instance used to reach the monitoring
             station.
             Defaults to the global network instance
             if not specified.";
        }
        leaf station-address {
          type inet:ip-address;
          mandatory true;
          description
            "IP address of the monitoring station.";
        }
        leaf station-port {
```

```
    type inet:port-number;
    mandatory true;
    description
      "Port number of the monitoring station.";
  }
  choice local-endpoint {
    mandatory true;
    description
      "Local endpoint for the connection.";
    case monitored-router-address {
      leaf monitored-router-address {
        type inet:ip-address;
        description
          "Local IP address to source the connection.";
      }
    }
    case monitored-router-interface {
      leaf monitored-router-interface {
        type if:interface-ref;
        description
          "Local interface to source the connection.";
      }
    }
  }
  leaf monitored-router-port {
    type inet:port-number;
    description
      "Optional local port for the active connection.";
  }
}

case passive {
  description
    "Device waits for incoming connection at a local
    endpoint.";
  container passive {
    description
      "Parameters for passively accepting
      a connection from the monitoring station.";
    leaf network-instance {
      type leafref {
        path "/ni:network-instances/ni:network-instance/"
          + "ni:name";
      }
    }
    description
      "Network instance used for the passive connection.
      Defaults to the global network instance if not
      specified.";
  }
}
```

```
    }
    leaf station-address {
      type inet:ip-address;
      mandatory true;
      description
        "IP address of the monitoring station.";
    }
    leaf station-port {
      type inet:port-number;
      description
        "Optional value identifying the origin port of the
        connection. If provided, it MUST match the receiving
        connection.";
    }
    choice local-endpoint {
      mandatory true;
      description
        "Local endpoint for the connection.";
      case monitored-router-address {
        leaf monitored-router-address {
          type inet:ip-address;
          description
            "Local IP address to accept the connection.";
        }
      }
      case monitored-router-interface {
        leaf monitored-router-interface {
          type if:interface-ref;
          description
            "Local interface to accept the connection.";
        }
      }
    }
    leaf monitored-router-port {
      type inet:port-number;
      mandatory true;
      description
        "Local port to accept the connection.";
    }
  }
}

leaf dscp {
  type inet:dscp;
  description
    "DSCP value for marking traffic to the monitoring station.";
  reference
    "RFC 6991: Common YANG Data Types";
}
```

```
    }  
  }  
  
  grouping route-monitoring-peer-grouping {  
    description  
      "General configuration options for route monitoring  
      of a peer.";  
    container filters {  
      description  
        "Filters for selecting which routes to export to  
        the monitoring station.";  
      container policy-filter {  
        if-feature "bmp-filter-based-on-route-policy";  
        description  
          "Filter routes using a routing policy from the  
          rt-pol module.  
          The policy should only contain accept/reject actions  
          and match prefix sets.";  
        leaf-list export-policy {  
          type leafref {  
            path "/rt-pol:routing-policy/"  
              + "rt-pol:policy-definitions/"  
              + "rt-pol:policy-definition/rt-pol:name";  
            require-instance true;  
          }  
          ordered-by user;  
          description  
            "Ordered list of policy names used to select  
            routes for export.";  
        }  
        leaf default-export-policy {  
          type rt-pol:default-policy-type;  
          default "accept-route";  
          description  
            "Default action if no export policy matches.";  
        }  
      }  
    }  
  }  
}  
  
grouping bmp-peer-ribs-filter-grouping {  
  description  
    "Configuration containers for RIBs under the main  
    BMP container.";  
  container address-families {  
    description  
      "List of address families for route monitoring.";  
    list address-family {
```



```
key "id";
description
  "Address family, as defined in the BGP model.";
leaf id {
  type identityref {
    base afi-safi-type;
  }
  description
    "Address family identifier.";
}
leaf enabled {
  type boolean;
  default "true";
  description
    "Enables route monitoring messages for this address
    family.";
}
container peers-configurations {
  description
    "Each peer under this BGP instance can be configured by
    at most one of the following containers.
    If the peer is not covered by any,
    then no BMP route monitoring message
    should include information from/to that peer.
    If the peer is covered by more than one, then the
    priority is:
    1. peer
    2. peer-groups
    3. peer-type
    4. peer-selectors

    New child containers or new bmp-peer-selectors
    instances SHOULD provide a way of unambiguously
    selecting which configuration container should
    be selected
    for a peer in case of multiple matches.

    Note that if the implementation supports module
    ietf-bmp-bgp-dependencies, the peer configurations
    under the BGP container have priority over the
    configurations under this container.";
  container peers {
    description
      "Configuration for individual peers.";
    list peer {
      key "id";
      description
        "Peer identifier.";
```

```
    leaf id {
      type string;
      description
        "Identifier of the peer.";
    }
    leaf enabled {
      type boolean;
      default "true";
      description
        "Enables route monitoring messages for this peer.";
    }
    uses route-monitoring-peer-grouping;
  }
}
container peer-groups {
  description
    "Configuration for peer groups.";
  list peer-group {
    key "id";
    description
      "Peer group identifier.";
    leaf id {
      type string;
      description
        "Identifier of the peer group.";
    }
    leaf enabled {
      type boolean;
      default "true";
      description
        "Enables route monitoring messages for
        this peer group.";
    }
    uses route-monitoring-peer-grouping;
  }
}
container peer-selectors {
  description
    "Configuration for peers selected by BMP peer
    selectors.";
  list peer-selector {
    key "id";
    description
      "Identification of peers
      for which we send BMP data to the collector
      using a peer type defined using a
      bmp-peer-selectors identity.
      For instance, to create a default for all
```

```

        peers use all-peers";
    leaf id {
        type identityref {
            base bmp-peer-selectors;
        }
        description
            "BMP peer selector identity.";
    }
    leaf enabled {
        type boolean;
        default "true";
        description
            "Enables route monitoring messages
             for the peer(s).";
    }
    uses route-monitoring-peer-grouping;
}

container peer-types {
    description
        "Generic identification of peers to configure.";
    list peer-type {
        key "id";
        description
            "Identification of peers
             for which we send BMP data to the collector
             using BGP peer-type (e.g. internal, external)
            ";
        leaf id {
            type peer-type;
            description
                "BGP peer type.";
        }
        leaf enabled {
            type boolean;
            default "true";
            description
                "Enables route monitoring messages
                 for the peer(s).";
        }
        uses route-monitoring-peer-grouping;
    }
}
}
}
}
}

```

```
grouping generic-network-instance-grouping {
  description
    "Generic configuration of a network instance.";
  leaf enabled {
    type boolean;
    default "true";
    description
      "Enables route monitoring
       messages for the network instance.";
  }
  container adj-rib-in-pre {
    description
      "Configuration for the adj-rib-in pre-policy.";
    reference
      "RFC7854: BGP Monitoring Protocol (BMP), Section 2.";
    uses bmp-peer-ribs-filter-grouping;
  }
  container adj-rib-in-post {
    description
      "Configuration for the adj-rib-in post-policy";
    reference
      "RFC7854: BGP Monitoring Protocol (BMP), Section 2.";
    uses bmp-peer-ribs-filter-grouping;
  }
  container local-rib {
    description
      "Configuration for the local-rib.";
    reference
      "RFC9069: Support for Local RIB in the BGP Monitoring
       Protocol (BMP), Section 3.";
    container address-families {
      description
        "List of address families to enable for local-rib.";
      list address-family {
        key "id";
        description
          "Address family to enable for local-rib";
        leaf id {
          type identityref {
            base afi-safi-type;
          }
          description
            "Address family id to enable for local-rib";
        }
        uses route-monitoring-peer-grouping;
      }
    }
  }
}
```

```
container adj-rib-out-pre {
  description
    "Configuration for the adj-rib-out pre-policy";
  reference
    "RFC8671: Support for Adj-RIB-Out in the BGP Monitoring
      Protocol (BMP) , Section 3.";
  uses bmp-peer-ribs-filter-grouping;
}
container adj-rib-out-post {
  description
    "Configuration for the adj-rib-out post-policy";
  reference
    "RFC8671: Support for Adj-RIB-Out in the BGP Monitoring
      Protocol (BMP) , Section 3.";
  uses bmp-peer-ribs-filter-grouping;
}
}

grouping route-monitoring-sources {
  description
    "Configuration of route monitoring sources.";
  reference
    "RFC7854: BGP Monitoring Protocol, Section 5.";
  container network-instance-configuration {
    description
      "This container offers options for configuring BMP
        route-monitoring messages for each network instance either
        selecting it through its name or through a
        network-instance-selectors.

        Network-instance-selectors are instances of bmp-ni-types
        that select one or more network instances for configuration.
        For instance, all-ni to configure all network
        instances (serving as a default).

        Network-instance can be at most configured by one of the
        containers. If the network instance is not covered by any,
        then no BMP route monitoring message should include that
        network instance. If more than one container matches
        the network instance, the priority for selecting the
        container to use for configuration is:

        1. For any named network instance, the configuration
           under the element listed with its name under the
           network-instance container.
        2. If the global-ni network-instance type exists,
           it SHOULD be used for the global-ni.
           However, if the global-ni
```

- has an explicit name, and it is configured, then from the previous rule, the explicit network instance name configuration SHOULD be used.
3. The configuration under network-instance-groups of type non-global-ni if existing and not the global network instance.
  4. the configuration under network-instance-groups under the element all-ni.

If the implementation has a name for the global network instance (e.g. 'main') it can be configure directly under the network-instances container.

New identities under bmp-ni-types or augmentations of this container in the future SHOULD provide a clear way of selecting the configuration container for a network-instance without ambiguity.";

```

container network-instances {
  description
    "Configuration for specific network instances";
  list network-instance {
    key "id";
    description
      "Network instance to monitor using BMP.";
    leaf id {
      type leafref {
        path "/ni:network-instances/ni:network-instance/"
          + "ni:name";
      }
    }
    description
      "Name of the network instance.";
  }
  uses generic-network-instance-grouping;
}

container network-instance-selectors {
  description
    "Configuration of network instances. Uses
    bmp-ni-types to identify one or a group of
    network instances to configure.";
  list network-instance-selector {
    key "id";
    description
      "Network instance(s) to monitor using BMP.";
    leaf id {
      type identityref {
        base bmp-ni-types;
      }
    }
  }
}

```

```

        description
            "Configures one or multiple network instances selected
            based on a bmp-ni-types identity (e.g.
            all-ni for all of them).";
    }
    uses generic-network-instance-grouping;
}
}
}
}

container bmp {
    description
        "Top-level container for BMP configuration.";
    container monitoring-stations {
        description
            "List of BMP monitoring stations.";
        list monitoring-station {
            key "id";
            description
                "Configuration for a BMP monitoring station.";
            leaf id {
                type string;
                description
                    "Unique identifier for the monitoring station.";
            }
            leaf description {
                type string;
                description
                    "Description of the BMP monitoring station.";
            }
        }
        container connection {
            description
                "Connection parameters for the monitoring station.";
            uses bmp-ip-connection-grouping;
            container tcp-options {
                description
                    "TCP options for the connection to the monitoring
                    station.";
                leaf maximum-segment-size {
                    type uint16;
                    description
                        "Maximum segment size for the TCP connections.
                        In the absence of this container, the system
                        will select the maximum segment size for this
                        connection.";
                }
            }
        }
    }
}
// Taken from the bgp yang module

```

```
leaf mtu-discovery {
  type boolean;
  default "true";
  description
    "Enables path MTU discovery for the TCP sessions
    (true) or disables it (false).";
  reference
    "RFC 1191: Path MTU discovery.";
}
}
leaf initial-delay {
  type uint32;
  units "seconds";
  default "0";
  description
    "Initial delay before connecting to the monitoring
    station.
    Useful for allowing BGP sessions to stabilize
    before starting BMP.";
}
container backoff {
  description
    "Configures the backoff strategy after a connection
    retry";
  reference
    "RFC7854 Section 3.2";
  choice backoff-options {
    description
      "Options for backoff strategies";
    reference
      "RFC7854 Section 3.2";
    case simple-exponential {
      description
        "Simple exponential backoff with limits.";
      container simple-exponential {
        description
          "Simple exponential backoff with limits.
          Starts with the initial backoff and doubles
          the backoff after every retry until reaching the
          maximum backoff.";
        leaf initial-backoff {
          type uint32;
          units "seconds";
          default "30";
          description
            "Initial backoff time";
        }
        leaf maximum-backoff {
```



```

        type uint32;
        units "seconds";
        default "720";
        description
            "Maximum backoff time";
    }
}
}
}
}
}
container bmp-data {
    description
        "Configuration of BMP data sent to the monitoring
        station.";
    leaf initiation-message {
        type string;
        description
            "User-defined message to append to the
            initiation message.";
        reference
            "RFC7854: BGP Monitoring Protocol,
            Section 4.3 and 4.4";
    }
    container statistics-report {
        presence "Enable BMP statistics report.";
        description
            "Configuration for periodic statistics reports.";
        reference
            "RFC7854: BGP Monitoring Protocol,
            Section 4.8";
        leaf statistics-interval {
            type uint32;
            units "seconds";
            mandatory true;
            description
                "Interval between statistics report messages.";
        }
    }
}
container route-monitoring {
    description
        "Configuration of the data sources for
        route-monitoring messages";
    uses route-monitoring-sources;
}
container route-mirroring {
    presence "Enable BMP route mirroring to the monitoring
    station.";

```

```
        description
          "Configuration for route mirroring to the
           monitoring station.";
      }
    }
  container session-stats {
    config false;
    description
      "Operational statistics for the monitoring station.
       Counters are reset after each successful
       connection or reset.";
    grouping bmp-af-stats-with-peers-grouping {
      description
        "Generic statistics for an address family that can be
         disaggregated by peers";
      container peers-stats {
        description
          "Peer stats";
        list peer {
          key "id";
          description
            "List of peers";
          leaf id {
            type string;
            description
              "Peer id";
          }
          leaf enabled {
            type boolean;
            description
              "Indicates if route monitoring messages are
               currently enabled for the peer under this
               network instance, address family, and RIB.";
          }
          leaf total-route-mnt-updated-prefixes-per-peer {
            type uint64;
            description
              "Number of prefixes updated for this peer.";
          }
          leaf total-route-mnt-withdraw-prefixes-per-peer {
            type uint64;
            description
              "Number of prefixes withdrawn for this peer.";
          }
        }
      }
    }
  }
}
```

```
grouping bmp-af-stats-grouping {
  description
    "Group for statistics for an address family.";
  leaf enabled {
    type boolean;
    description
      "Indicates if any route monitoring messages
       are currently enabled for the address family
       within the RIB.";
  }
  leaf total-route-monitoring-updated-prefixes-per-af {
    type uint64;
    description
      "Number of prefixes updated for this address
       family.";
  }
  leaf total-route-monitoring-withdraw-prefixes-per-af {
    type uint64;
    description
      "Number of prefixes withdrawn for this address
       family.";
  }
}

grouping bmp-rib-with-peers-stats-grouping {
  description
    "Generic statistics for a RIB with peers.";
  container address-families {
    description
      "List of address families to list stats.";
    list address-family {
      key "id";
      description
        "Address family to enable for local-rib";
      leaf id {
        type identityref {
          base afi-safi-type;
        }
        description
          "Address family ID for local-rib.";
      }
      uses bmp-af-stats-grouping;
      uses bmp-af-stats-with-peers-grouping;
    }
  }
}

grouping bmp-rib-stats-grouping {
```

```
description
  "Generic statistics per RIB.";
leaf enabled {
  type boolean;
  description
    "Indicates if any Route Monitoring messages are
    currently enabled for the RIB.";
}
leaf total-route-mirroring-messages-per-rib {
  type uint64;
  description
    "Number of route-mirroring messages sent for
    this RIB.";
}
}

leaf discontinuity-time {
  type yang:date-and-time;
  mandatory true;
  description
    "The time on the most recent occasion at which any
    one or more of this station's counters suffered a
    discontinuity. If no such discontinuities have
    occurred since the last re-initialization of the
    local management subsystem, then this node contains
    the time the local management subsystem
    re-initialized itself.";
}
leaf established-session {
  type boolean;
  description
    "Indicates if the session is currently
    established.";
}
leaf total-route-monitoring-messages {
  type uint64;
  description
    "Number of route-monitoring messages sent.";
}
leaf total-statistics-messages {
  type uint64;
  description
    "Number of statistics messages sent.";
}
leaf total-peer-down-messages {
  type uint64;
  description
    "Number of peer-down messages sent.";
```

```
}
leaf total-peer-up-messages {
  type uint64;
  description
    "Number of peer-up messages sent.";
}
leaf total-initiation-messages {
  type uint64;
  description
    "Number of initiation messages sent";
}
leaf total-route-mirroring-messages {
  type uint64;
  description
    "Number of route-mirroring messages sent.";
}
leaf total-termination-messages {
  type uint64;
  description
    "Number of termination messages sent.";
}
container route-monitoring-stats {
  description
    "Statistics of route monitoring messages disaggregated
    by RIB and peers where applicable.";
  container network-instances-stats {
    description
      "Stats per network-instance";
    list network-instance {
      key "network-instance-name";
      description
        "Network instance stats list";
      leaf network-instance-name {
        type leafref {
          path "/ni:network-instances/ni:network-instance/"
            + "ni:name";
        }
        description
          "Name of the network instance.";
      }
      leaf enabled {
        type boolean;
        description
          "Indicates if route monitoring messages are
          currently enabled for the network instance.";
      }
      leaf total-route-mirroring-messages-per-ni {
        type uint64;
      }
    }
  }
}
```

```
description
  "Number of route-mirroring messages sent for
  this network instance.";
}
container ribs-stats {
  description
    "Statistics for the different RIBs.";
  container adj-rib-in-pre {
    description
      "Statistics for adj-rib-in-pre.";
    uses bmp-rib-stats-grouping;
    uses bmp-rib-with-peers-stats-grouping;
  }
  container adj-rib-in-post {
    description
      "Statistics for adj-rib-in-post";
    uses bmp-rib-stats-grouping;
    uses bmp-rib-with-peers-stats-grouping;
  }
  container local-rib {
    description
      "Statistics for local-rib";
    uses bmp-rib-stats-grouping;
    container address-families {
      description
        "List of address families to for stats.";
      list address-family {
        key "id";
        description
          "Address family to enable for local-rib";
        leaf id {
          type identityref {
            base afi-safi-type;
          }
          description
            "Address family ID for local-rib";
        }
        uses bmp-af-stats-grouping;
      }
    }
  }
}
container adj-rib-out-pre {
  description
    "Statistics for adj-rib-out-pre";
  uses bmp-rib-stats-grouping;
  uses bmp-rib-with-peers-stats-grouping;
}
container adj-rib-out-post {
```







```
<network-instance-configuration>
  <network-instance-selectors>
    <network-instance-selector>
      <id>all-ni</id>
      <adj-rib-in-pre>
        <address-families>
          <address-family>
            <id>ipv6-unicast</id>
            <peers-configurations>
              <peer-types>
                <peer-type>
                  <id>external</id>
                </peer-type>
              </peer-types>
            </peers-configurations>
          </address-family>
          <address-family>
            <id>ipv4-unicast</id>
            <peers-configurations>
              <peer-types>
                <peer-type>
                  <id>external</id>
                </peer-type>
              </peer-types>
            </peers-configurations>
          </address-family>
        </address-families>
      </adj-rib-in-pre>
    </network-instance-selector>
  </network-instance-selectors>
</network-instance-configuration>
</route-monitoring>
</bmp-data>
</monitoring-station>
</monitoring-stations>
</bmp>
```

Figure 16

## C.2. Example two

In the next example, the device connects to a monitoring station using a passive connection, over the network-instance monitoring. The configuration of route monitoring messages is more complex than in the previous example. It shows how to combine the configuration of general identities of network instances and peers (e.g. all-ni for NI, external for peers), and individual configurations to support a more complex requirement. This is what the example expects to

configure:

- \* For the global network instance, the device sends updates for adj-rib-in-pre, address families IPv4 and IPv6. It sends updates for all external peers except peer 198.51.100.11, which is disabled.
- \* Network instance monitoring is disabled for route monitoring messages.
- \* For the rest of network instances, we are enabling messages from adj-rib-in-pre, address families IPv4/IPv6, and for all peers.

===== NOTE: '\ ' line wrapping per RFC 8792 =====

```
<bmp xmlns="urn:ietf:params:xml:ns:yang:ietf-bmp">
  <monitoring-stations>
    <monitoring-station>
      <id>2</id>
      <connection>
        <passive>
          <network-instance>monitoring</network-instance>
          <station-address>192.0.2.1</station-address>
          <monitored-router-address>192.0.2.2</monitored-router-ad\
dress>
          <monitored-router-port>57993</monitored-router-port>
        </passive>
      </connection>
    <bmp-data>
      <route-monitoring>
        <network-instance-configuration>
          <network-instance-selectors>
            <network-instance-selector>
              <id>all-ni</id>
              <adj-rib-in-pre>
                <address-families>
                  <address-family>
                    <id>ipv6-unicast</id>
                    <peers-configurations>
                      <peer-selectors>
                        <peer-selector>
                          <id>all-peers</id>
                        </peer-selector>
                      </peer-selectors>
                    </peers-configurations>
                  </address-family>
                  <address-family>
                    <id>ipv4-unicast</id>
                    <peers-configurations>
```

```
<peer-selectors>
  <peer-selector>
    <id>all-peers</id>
  </peer-selector>
</peer-selectors>
</peers-configurations>
</address-family>
</address-families>
</adj-rib-in-pre>
</network-instance-selector>
<network-instance-selector>
  <id>global-ni</id>
  <adj-rib-in-pre>
    <address-families>
      <address-family>
        <id>ipv6-unicast</id>
        <peers-configurations>
          <peers>
            <peer>
              <id>198.51.100.11</id>
              <enabled>false</enabled>
            </peer>
          </peers>
          <peer-types>
            <peer-type>
              <id>external</id>
            </peer-type>
          </peer-types>
        </peers-configurations>
      </address-family>
      <address-family>
        <id>ipv4-unicast</id>
        <peers-configurations>
          <peers>
            <peer>
              <id>198.51.100.11</id>
              <enabled>false</enabled>
            </peer>
          </peers>
          <peer-types>
            <peer-type>
              <id>external</id>
            </peer-type>
          </peer-types>
        </peers-configurations>
      </address-family>
    </address-families>
  </adj-rib-in-pre>
```

```
        </network-instance-selector>
    </network-instance-selectors>
    <network-instances>
    <network-instance>
        <id>monitoring</id>
        <enabled>false</enabled>
    </network-instance>
    </network-instances>
    </network-instance-configuration>
    </route-monitoring>
</bmp-data>
</monitoring-station>
</monitoring-stations>
</bmp>
```

Figure 17

#### Acknowledgements

The authors would like to thank Yimin Shen, Jeff Haas, Pierre Vander Vorst, and Tom Petch for their review and feedback.

#### Authors' Addresses

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

Paolo Lucente  
NTT  
Siriusdreef 70-72  
2132 Hoofddorp  
Netherlands  
Email: paolo@ntt.net

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

Benoit Claise  
Everything OPS  
Email: Benoit@everything-ops.net

Dhananjay Patki  
Cisco  
Cessna Business Park SEZ, Kadubeesanahalli  
Bangalore  
India  
Email: dhpatki@cisco.com

Prasad S. Narasimha  
Cisco  
Cessna Business Park SEZ, Kadubeesanahalli  
Bangalore  
India  
Email: snprasad@cisco.com