| | |
|---|---|
| Network Working Group | L. Blunk |
| Internet-Draft | M. Karir |
| Intended status: Standards Track | Merit Network |
| Expires: March 13, 2011 | C. Labovitz |
| | Arbor Networks |
| | September 9, 2010 |


TOC

**MRT routing information export format**
**draft-ietf-grow-mrt-12.txt**

## Abstract


This document describes the MRT format for routing information export. This format was developed in concert with the Multi-threaded Routing Toolkit (MRT) from whence the format takes it name. The format can be used to export routing protocol messages, state changes, and routing information base contents.


## Status of this Memo



## Copyright Notice

---

**Table of Contents**

---

## 1. Requirements notation

TOC

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119] (Bradner, S., “Key words for use in RFCs to Indicate Requirement Levels,” March 1997.).

---

## 2. Introduction

TOC

Researchers and engineers often wish to analyze network behavior by studying routing protocol transactions and routing information base snapshots. To this end, the MRT format was developed to encapsulate, export, and archive this information in a standardized data representation. The BGP routing protocol, in particular, has been the subject of extensive study and analysis which has been significantly aided by the availability of the MRT format. The MRT format was initially defined in the MRT Programmer's Guide (Labovitz, C., “MRT Programmer's Guide,” November 1999.) [MRT PROG GUIDE].
This memo serves to document the MRT format as currently implemented in publicly available software. The format has been extended since its original introduction in the MRT toolset and these extensions are also included in this memo. Further extensions may be introduced at a later date through additional definitions of the MRT Type field and Subtype fields.
A number of MRT message types have been documented in some references but are not known to have been implemented. Further, several types were employed in early MRT implementations, but are no longer actively being used. These types are considered to be deprecated and are documented in a separate appendix at the end of this document. Some of the deprecated types may of interest to researchers examining historical MRT archives.

Fields which contain multi-octet numeric values are encoded in network octet order from most significant octet to least significant octet. Fields which contain routing message fields are encoded in the same order as they appear in the packet contents.

## 3. Basic MRT Format

TOC

All MRT format messages have a common header which includes a timestamp, Type, Subtype, and length field. The header is followed by a message field. The MRT common header is illustrated below.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           Timestamp                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|             Type              |            Subtype            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                             Length                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Message... (variable)
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

**Figure 1: Basic MRT Format**

Header Field Descriptions:

Timestamp:

Time in seconds since 1 January 1970 00:00:00 UTC

Type:

A 2-octet field that indicates the Type of information contained in the message field. Types 0 through 4 are informational messages pertaining to the state of an MRT collector, while Types 5 and higher are used to convey routing information.

Subtype:

A 2-octet field that is used to further distinguish message information within a particular message Type.

Length:

A 4-octet message length field. The length field contains the number of octets within the message. The length field does not include the length of the MRT common header.

Message:

A variable length message. The contents of this field are context dependent upon the Type and Subtype fields.

---

## 4. MRT Informational Types

TOC

The MRT format defines five Informational Type messages. These messages are intended to signal the state of an MRT data collector and do not contain routing information. These messages are OPTIONAL and were largely intended for use when MRT messages are sent over a network to a remote repository store. However, MRT message repository stores have traditionally resided on the same device as the collector and these Informational Types have seen limited implementation. Further, transport mechanisms for MRT messages are considered to be outside the scope of this document.

The START and I_AM_DEAD messages MAY be used to provide a time reference when a data collector begins and ends the collection process. The time reference is obtained from the Timestamp field in the MRT message header.

The message field MAY contain an OPTIONAL message string for diagnostic purposes. The message string encoding MUST follow the UTF-8 transformation format. The Subtype field is unused for these Types and SHOULD be set to 0.

The MRT Informational Types are defined below:

1 START
3 I_AM_DEAD

---

TOC

### 4.1. START Type

The START Type indicates a collector is about to begin generating MRT messages.

### 4.2. I_AM_DEAD Type

TOC

An I_AM_DEAD MRT message indicates that a collector has shut down and has stopped generating MRT messages.

## 5. MRT Routing Information Types

TOC

The following MRT Routing Information Types are currently defined for the MRT format:

```
11    OSPF
12    TABLE_DUMP
13    TABLE_DUMP_V2
16    BGP4MP
17    BGP4MP_ET
32    ISIS
33    ISIS_ET
48    OSPFv3
49    OSPFv3_ET
```

### 5.1. OSPF Type

TOC

This Type supports the OSPF Protocol as defined in RFC 2328 (Moy, J., “OSPF Version 2,” April 1998.) [RFC2328]. The Subtype field may contain two possible values:

```
0     OSPF_STATE_CHANGE
1     OSPF_LSA_UPDATE
```

The format of the MRT Message field for the OSPF Type is as follows:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Remote IP address                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Local IP address                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  OSPF Message Contents (variable)
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
```

**Figure 2: OSPF Type**

### 5.2. TABLE_DUMP Type

TOC

The TABLE_DUMP Type is used to encode the contents of a BGP Routing Information Base (RIB). Each RIB entry is encoded in a distinct sequential MRT record. The Subtype field is used to encode whether the RIB entry contains IPv4 or IPv6 addresses. There are two possible values for the Subtype as shown below.

```
1    AFI_IPv4
2    AFI_IPv6
```

The format of the TABLE_DUMP Type is illustrated below.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           View #              |       Sequence number         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Prefix (variable)                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Prefix Length |    Status     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Originated Time                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Peer IP address (variable)                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Peer AS             |       Attribute Length        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   BGP Attribute... (variable)
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

**Figure 3: TABLE_DUMP Type**

The View field is normally 0 and is intended for cases where an implementation may have multiple RIB views (such as a route server). In cases where multiple RIB views are present, an implementation may use the the view field to distinguish entries from each view. The Sequence field is a simple incremental counter for each RIB entry. A typical RIB dump will exceed the 16-bit bounds of this counter and implementation should simply wrap back to zero and continue incrementing the counter in such cases.

The Prefix field contains the IP address of a particular RIB entry. The size of this field is dependent on the value of the Subtype for this message. For AFI_IPv4, this field is 4 octets, for AFI_IPv6, it is 16 octets in length. The Prefix Length field indicates the length in bits of the prefix mask for the preceding Prefix field.

The Status octet is unused in the TABLE_DUMP Type and SHOULD be set to 1.

The Originated Time contains the 4-octet time at which this prefix was heard. The value represents the time in seconds since 1 January 1970 00:00:00 UTC.

The Peer IP field is the IP address of the peer which provided the update for this RIB entry. As with the Prefix field, the size of this field is dependent on the Subtype. AFI_IPv4 indicates a 4 octet field and an IPv4 address, while a Subtype of AFI_IPv6 requires a 16 octet field and an IPv6 address. The Peer AS field contains the 2 octet AS number of the peer.

Note that the TABLE_DUMP Type does not permit 4-Byte Peer AS numbers. Nor does it allow the AFI of the peer IP to differ from the AFI of the Prefix field. The TABLE_DUMP_V2 Type must be used in these situations. Attribute Length contains the length of Attribute field and is 2-octets. The BGP Attribute field contains the BGP attribute information for the RIB entry.

---

### 5.3. TABLE_DUMP_V2 Type

TOC

The TABLE_DUMP_V2 Type updates the TABLE_DUMP Type to include 4-Byte ASN support and full support for BGP Multiprotocol extensions. It also improves upon the space efficiency of the TABLE_DUMP Type by employing an index table for peers and permitting a single MRT record per NLRI entry. The following subtypes are used with the TABLE_DUMP_V2 Type.

```
1    PEER_INDEX_TABLE
2    RIB_IPV4_UNICAST
3    RIB_IPV4_MULTICAST
4    RIB_IPV6_UNICAST
5    RIB_IPV6_MULTICAST
6    RIB_GENERIC
```

An initial PEER_INDEX_TABLE MRT record provides the BGP ID of the collector, an optional view name, and a list of indexed peers. Following the PEER_INDEX_TABLE MRT record, a series of MRT records are used to encode RIB table entries. This series of MRT records use subtypes 2-6 and are separate from the PEER_INDEX_TABLE MRT record itself and include full MRT record headers. Note that the RIB entry MRT records MUST immediately follow the PEER_INDEX_TABLE MRT record.
The header of the PEER_INDEX_TABLE Subtype is shown below. The View Name is optional and, if not present, the View Name Length MUST be set to 0. The View Name encoding MUST follow the UTF-8 transformation format.

---

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Collector BGP ID                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       View Name Length        |     View Name (variable)      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Peer Count           |    Peer Entries (variable)
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

**Figure 4: PEER_INDEX_TABLE Subtype**

The format of the Peer Entries is shown below. The PEER_INDEX_TABLE record contains Peer Count number of Peer Entries.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Peer Type   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Peer BGP ID                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   Peer IP address (variable)                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Peer AS (variable)                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

**Figure 5: Peer Entries**

The Peer Type, Peer BGP ID, Peer IP, and Peer AS fields are repeated as indicated by the Peer Count field. The position of the Peer in the PEER_INDEX_TABLE is used as an index in the subsequent TABLE_DUMP_V2 MRT records. The index number begins with 0.
The Peer Type field is a bit field which encodes the type of the AS and IP address as follows:

```
Bit 0 - unset for IPv4 Peer IP address, set for IPv6
Bit 1 - unset when Peer AS is 16 bits, set when it's 32 bits
```

The MRT records which follow the PEER_INDEX_TABLE MRT record contain the RIB entries and include a header which specifies a sequence number, NLRI, and a count of the number of RIB entries which follow.
The format for the RIB_IPV4_UNICAST, RIB_IPV4_MULTICAST, RIB_IPV6_UNICAST, and RIB_IPV6_MULTICAST headers are shown below. The Prefix Length and Prefix fields are encoded in the same manner as the BGP NLRI encoding for IPV4 and IPV6 prefixes. Namely, the Prefix field contains address prefixes followed by enough trailing bits to make the end of the field fall on an octet boundary. Note that the value of trailing bits is irrelevant.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Sequence number                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Prefix Length |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Prefix (variable)                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Entry Count           |  RIB Entries (variable)
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

**Figure 6: RIB Entry Header**

The RIB_GENERIC header is shown below. It is used to cover RIB entries which do not fall under the common case entries defined above. It includes Address Family Identifier (AFI), Subsequent AFI and a single NLRI entry. The NLRI information is specific to the AFI and SAFI values. An implementation which does not recognize particular AFI and SAFI values SHOULD discard the remainder of the MRT record.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Sequence number                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Address Family Identifier  |Subsequent AFI |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Network Layer Reachability Information (variable)         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Entry Count           |  RIB Entries (variable)
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

**Figure 7: RIB_GENERIC Entry Header**

The RIB and RIB_GENERIC Entry Headers are followed by a series of RIB Entries which are repeated Entry Count times. These entries share a common format as shown below. They include a Peer Index from the PEER_INDEX_TABLE MRT record, an originated time for the RIB Entry, and the BGP path attribute length and attributes encoded as provided in a BGP Update message.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Peer Index            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Originated Time                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      Attribute Length         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    BGP Attributes... (variable)
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

**Figure 8: RIB Entries**

There is one exception to the encoding of BGP attributes for the BGP MP_REACH_NLRI attribute (BGP Type Code 14) RFC 4760 (Bates, T., Chandra, R., Katz, D., and Y. Rekhter, “Multiprotocol Extensions for BGP-4,” January 2007.) [RFC4760]. Since the AFI, SAFI, and NLRI information is already encoded in the MULTIPROTOCOL header, only the Next Hop Address Length and Next Hop Address fields are included. The Reserved field is omitted. The attribute length is also adjusted to reflect only the length of the Next Hop Address Length and Next Hop Address fields.

### 5.4. BGP4MP Type

TOC

This Type was initially defined in the Zebra software package for the BGP protocol with multiprotocol extension support as defined by RFC 4760 (Bates, T., Chandra, R., Katz, D., and Y. Rekhter, “Multiprotocol Extensions for BGP-4,” January 2007.) [RFC4760]. It supersedes the BGP, BGP4PLUS, BGP4PLUS_01 Types. The BGP4MP Type has six Subtypes which are defined as follows:

```
0    BGP4MP_STATE_CHANGE
1    BGP4MP_MESSAGE
4    BGP4MP_MESSAGE_AS4
5    BGP4MP_STATE_CHANGE_AS4
6    BGP4MP_MESSAGE_LOCAL
7    BGP4MP_MESSAGE_AS4_LOCAL
```

#### 5.4.1. BGP4MP_STATE_CHANGE Subtype

TOC

This record is used to encode state changes in the BGP finite state machine. The BGP FSM states are encoded in the Old State and New State fields to indicate the previous and current state. In some cases, the Peer AS number may be undefined. In such cases, the value of this field may be set to zero. The format is illustrated below:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Peer AS number        |        Local AS number        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        Interface Index        |        Address Family         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Peer IP address (variable)               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Local IP address (variable)              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            Old State          |          New State            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

**Figure 9: BGP4MP_STATE_CHANGE Subtype**

The FSM states are defined in RFC 4271 (Rekhter, Y., Li, T., and S. Hares, “A Border Gateway Protocol 4 (BGP-4),” January 2006.) [RFC4271], Section 8.2.2. Both the old state value and the new state value are encoded as 2-octet numbers. The state values are defined numerically as follows:

```
1    Idle
2    Connect
3    Active
4    OpenSent
5    OpenConfirm
6    Established
```

The BGP4MP_STATE_CHANGE message also includes interface index and Address Family fields. The interface index provides the interface number of the peering session. The index value is OPTIONAL and MAY be zero if unknown or unsupported. The Address Family indicates what types of addresses are in the the address fields. At present, the following AFI Types are supported:

| | |
|---|---|
| 1 | AFI_IPv4 |
| 2 | AFI_IPv6 |

### 5.4.2. BGP4MP_MESSAGE Subtype

TOC

This Subtype is used to encode BGP Messages. It can be used to encode any Type of BGP message. The entire BGP message is encapsulated in the BGP Message field, including the 16-octet marker, the 2-octet length, and the 1-octet type fields. Note that the BGP4MP_MESSAGE Subtype does not support 4-Byte AS numbers. Further, the AS_PATH contained in these messages MUST only consist of 2-Byte AS numbers. The BGP4MP_MESSAGE_AS4 Subtype updates the BGP4MP_MESSAGE Subtype in order to support 4-Byte AS numbers. The BGP4MP_MESSAGE fields are shown below:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Peer AS number        |        Local AS number        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        Interface Index        |        Address Family         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Peer IP address (variable)               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Local IP address (variable)              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    BGP Message... (variable)
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

**Figure 10: BGP4MP_MESSAGE Subtype**

The interface index provides the interface number of the peering session. The index value is OPTIONAL and MAY be zero if unknown or unsupported. The Address Family indicates what types of addresses are in the the subsequent address fields. At present, the following AFI Types are supported:

| | |
|---|---|
| 1 | AFI_IPv4 |
| 2 | AFI_IPv6 |

Note that the Address Family value only applies to the IP addresses contained in the MRT header. The BGP4MP_MESSAGE Subtype is otherwise transparent to the contents of the actual message which may contain any valid AFI/SAFI values. Only one BGP message may be encoded in the BGP4MP_MESSAGE Subtype.

### 5.4.3. BGP4MP_MESSAGE_AS4 Subtype

TOC

This Subtype updates the BGP4MP_MESSAGE Subtype to support 4-Byte Autonomous System numbers. The BGP4MP_MESSAGE_AS4 Subtype is otherwise identical to the BGP4MP_MESSAGE Subtype. The AS_PATH in these messages MUST only consist of 4-Byte AS numbers. The BGP4MP_MESSAGE_AS4 fields are shown below:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Peer AS number                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Local AS number                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        Interface Index        |        Address Family         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Peer IP address (variable)               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Local IP address (variable)              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    BGP Message... (variable)
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

**Figure 11: BGP4MP_MESSAGE_AS4 Subtype**

### 5.4.4. BGP4MP_STATE_CHANGE_AS4 Subtype

TOC

This Subtype updates the BGP4MP_STATE_CHANGE Subtype to support 4-Byte Autonomous System numbers. As with the BGP4MP_STATE_CHANGE Subtype, the BGP FSM states are encoded in the Old State and New State fields to indicate the previous and current state. Aside from the extension of the peer and local AS fields to 4-Bytes, this subtype is otherwise identical to the BGP4MP_STATE_CHANGE Subtype. The BGP4MP_STATE_CHANGE_AS4 fields are shown below:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Peer AS number                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Local AS number                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        Interface Index        |        Address Family         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Peer IP address (variable)               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Local IP address (variable)              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            Old State          |          New State            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

**Figure 12: BGP4MP_STATE_CHANGE_AS4 Subtype**

### 5.4.5. BGP4MP_MESSAGE_LOCAL Subtype

TOC

Implementations of MRT have largely focused on collecting remotely generated BGP messages in a passive route collector role. However, for active BGP implementations, it can be useful to archive locally generated BGP messages in addition to remote messages. This subtype is added to indicated a locally generated BGP message. The fields remain identical to the BGP4MP_MESSAGE type including the Peer and Local IP and AS fields. The Local fields continue to refer to the local IP and AS number of the collector which generated the message and the Peer IP and AS fields refer to the receipient of the generated BGP messages.

TOC

### 5.4.6. BGP4MP_MESSAGE_AS4_LOCAL Subtype

As with the BGP4MP_MESSAGE_LOCAL type, this type indicate locally generated messages. The fields are identical to the BGP4MP_MESSAGE_AS4 message type.

### 5.5. BGP4MP_ET Type

TOC

This type extends the MRT common header field to include a 32BIT microsecond timestamp field. The type and subtype field definitions remain as defined for the BGP4MP Type. The 32BIT microsecond timestamp immediately follows the length field in the MRT common header and precedes all other fields in the message. The 32BIT microsecond field is included in the computation of the length field value. The MRT common header modification is illustrated below.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           Timestamp                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|             Type              |            Subtype            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                             Length                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      microsecond timestamp                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Message... (variable)
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
```

**Figure 13: BGP4MP_ET Type**

### 5.6. ISIS Type

TOC

This Type supports the IS-IS routing protocol as defined in RFC 1195 (Callon, R., “Use of OSI IS-IS for routing in TCP/IP and dual environments,” December 1990.) [RFC1195]. There is no Type specific header for the ISIS Type. The Subtype code for this Type is undefined. The ISIS PDU directly follows the MRT common header fields.

### 5.7. ISIS_ET Type

TOC

The ISIS_ET Type extends the ISIS Type to support microsecond timestamps. As with the BGP4MP_ET Type, a 32BIT microsecond timestamp field is appended to the MRT common header after the length field. The ISIS_ET Type is otherwise identical to the ISIS Type.

### 5.8. OSPFv3 Type

TOC

The OSPFv3 Type extends the original OSPF Type to support IPv6 addresses for the OSPFv3 protocol as defined in RFC 5340 (Coltun, R., Ferguson, D., Moy, J., and A. Lindem, "OSPF for IPv6," July 2008.) [RFC5340]. The format of the MRT Message field for the OSPFv3 Type is as follows:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        Address Family         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                Remote IP address (variable)                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                 Local IP address (variable)                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|               OSPF Message Contents (variable)
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

**Figure 14: OSPFv3 Type**

### 5.9. OSPFv3_ET Type

TOC

The OSPFv3_ET Type extends the OSPFv3 Type to support microsecond timestamps. As with the BGP4MP_ET Type, a 32BIT microsecond timestamp field is appended to the MRT common header after the length field and its length is included in the calculation of the length field value. The OSPFv3_ET Type is otherwise identical to the OSPFv3 Type.

## 6. Acknowledgements

[TOC]

The initial MRT specification was developed by Craig Labovitz for use in the Multi-thread Routing Toolkit (MRT) project. The BGP4MP Type was introduced in the Zebra routing software project by Kunihiro Ishiguro. The BGP4MP_ET, ISIS, and ISIS_ET Types were defined in the Python Routeing Toolkit (PyRT) developed by Richard Mortier while at Sprint Advanced Technology Labs.

## 7. IANA Considerations

[TOC]

This section provides guidance to the Internet Assigned Numbers Authority (IANA) regarding registration of values related to the MRT specification, in accordance with BCP 26, RFC 5226 (Narten, T. and H. Alvestrand, “Guidelines for Writing an IANA Considerations Section in RFCs,” May 2008.) [RFC5226].
There are two name spaces in MRT that require registration: Type Codes and Subtype Codes.
MRT is not intended as a general-purpose specification for protocol information export, and allocations should not be made for purposes unrelated to routing protocol information export.
The following policies are used here with the meanings defined in BCP 26: "Specification Required", "IETF Consensus", "Experimental Use", "First Come First Served".

### 7.1. Type Codes

[TOC]

Type Codes have a range from 0 to 65535, of which 1-64 have been allocated. New Type Codes MUST be allocated starting at 65. Type Codes 65 - 511 are to be assigned by IETF Review. Type Codes 512 - 2047 are assigned based on Specification Required. Type Codes 2048 - 64511 are available on a First Come First Served policy. Type Codes 64512 - 65534 are available for Experimental Use. The Type Code Values of 0 and 65535 are reserved.

[TOC]

### 7.2. Subtype Codes

Subtype Codes have a range from 0 to 65535. Subtype definitions are specific to a particular Type Code definition. New Subtype Code definition must reference an existing Type Code to which the Subtype belongs. Subtype assignmnents to Type Codes 0 - 511 are to be assigned by IETF Review. Subtype assignments for the remaning Type Codes follow the assignment rules for the Type Codes to which they belong.

## 8. Security Considerations

TOC

The MRT Format utilizes a structure which can store routing protocol information data. The fields defined in the MRT specification are of a descriptive nature and provide information that is useful to facilitate the analysis of routing data. As such, the fields currently defined in the MRT specification do not in themselves create additional security risks, since the fields are not used to induce any particular behavior by the recipient application.

## 9. References

TOC

### 9.1. Normative References

TOC


| | |
|---|---|
| [RFC1195] | Callon, R., "Use of OSI IS-IS for routing in TCP/IP and dual environments," RFC 1195, December 1990 (TXT, PS). |
| [RFC1723] | Malkin, G., "RIP Version 2 - Carrying Additional Information," STD 56, RFC 1723, November 1994 (TXT). |
| [RFC2080] | Malkin, G. and R. Minnear, "RIPng for IPv6," RFC 2080, January 1997 (TXT). |
| [RFC2119] | Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels," BCP 14, RFC 2119, March 1997 (TXT, HTML, XML). |
| [RFC2328] | Moy, J., "OSPF Version 2," STD 54, RFC 2328, April 1998 (TXT, HTML, XML). |
| [RFC4271] | Rekhter, Y., Li, T., and S. Hares, "A Border Gateway Protocol 4 (BGP-4)," RFC 4271, January 2006 (TXT). |
| [RFC4760] | Bates, T., Chandra, R., Katz, D., and Y. Rekhter, "Multiprotocol Extensions for BGP-4," RFC 4760, January 2007 (TXT). |
| [RFC5226] | |

| | |
|---|---|
| | Narten, T. and H. Alvestrand, “Guidelines for Writing an IANA Considerations Section in RFCs,” BCP 26, RFC 5226, May 2008 (TXT). |
| [RFC5340] | Coltun, R., Ferguson, D., Moy, J., and A. Lindem, “OSPF for IPv6,” RFC 5340, July 2008 (TXT). |


### 9.2. Informative References

TOC


| | |
|---|---|
| [MRT PROG GUIDE] | Labovitz, C., “MRT Programmer's Guide,” November 1999 (HTML). |


## Appendix A. Deprecated MRT types

TOC

This Appendix lists deprecated MRT types. These types are documented for informational purposes only. While documented in some references, they are not known to have been generally implemented.

### A.1. Deprecated MRT Informational Types

TOC

The deprecated MRT Informational Types are defined below:

```
0    NULL
2    DIE
4    PEER_DOWN
```

#### A.1.1. NULL Type

TOC

The NULL Type message causes no operation.

#### A.1.2. DIE Type

TOC

The DIE Type signals a remote MRT repository it should stop accepting messages.

### A.1.3. PEER_DOWN Type

TOC

The PEER_DOWN message was intended to indicate that a collector had lost association with a BGP peer. However, the MRT format provides BGP state change message types which duplicate this functionality.

---

## A.2. Deprecated MRT Routing Information Types

TOC

```
5     BGP
6     RIP
7     IDRP
8     RIPNG
9     BGP4PLUS
10    BGP4PLUS_01
```

---

### A.2.1. BGP Type

TOC

The BGP Type indicates the Message field contains BGP routing information. The BGP routing protocol is defined in RFC 4271 (Rekhter, Y., Li, T., and S. Hares, “A Border Gateway Protocol 4 (BGP-4),” January 2006.) [RFC4271]. The information in the message is dependent on the Subtype value. The BGP Type and all associated Subtypes below are considered to be deprecated by the BGP4MP Type.
The following BGP Subtypes are defined for the MRT BGP Type. As with the BGP Type itself, they are all considered to be deprecated.

```
0     BGP_NULL
1     BGP_UPDATE
2     BGP_PREF_UPDATE
3     BGP_STATE_CHANGE
4     BGP_SYNC
5     BGP_OPEN
6     BGP_NOTIFY
7     BGP_KEEPALIVE
```

---

#### A.2.1.1. BGP_NULL Subtype

TOC

The BGP_NULL Subtype is a reserved Subtype.

### A.2.1.2. BGP_UPDATE Subtype

TOC

The BGP_UPDATE Subtype is used to encode BGP UPDATE messages. The format of the MRT Message field for this Subtype is as follows:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Peer AS number        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Peer IP address                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        Local AS number        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Local IP address                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   BGP UPDATE Contents (variable)
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

**Figure 15: BGP_UPDATE Subtype**

The BGP UPDATE Contents include the entire BGP UPDATE message which follows the BGP Message Header. The BGP Message Header itself is not included. The Peer AS number and IP address fields contain the AS number and IP address of the remote system which are generating the BGP UPDATE messages. The Local AS number and IP address fields contain the AS number and IP address of the local collector system which is archiving the messages.

### A.2.1.3. BGP_PREF_UPDATE Subtype

TOC

The BGP_PREF_UPDATE Subtype is not defined.

TOC

#### A.2.1.4. BGP_STATE_CHANGE Subtype

The BGP_STATE_CHANGE Subtype is used to record changes in the BGP finite state machine. These FSM states are defined in RFC 4271 (Rekhter, Y., Li, T., and S. Hares, “A Border Gateway Protocol 4 (BGP-4),” January 2006.) [RFC4271], Section 8.2.2. Both the old state value and the new state value are encoded as 2-octet numbers. The state values are defined numerically as follows:

```
1    Idle
2    Connect
3    Active
4    OpenSent
5    OpenConfirm
6    Established
```

The format of the BGP_STATE_CHANGE Subtype MRT Message field is as follows:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Peer AS number        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Peer IP address                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            Old State          |          New State            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

**Figure 16: BGP_STATE_CHANGE Subtype**

#### A.2.1.5. BGP_SYNC Subtype

TOC

The BGP_SYNC Subtype was intended to convey a system file name where BGP Table Dump messages should be recorded. The View # was to correspond to the View # provided in the TABLE_DUMP Type messages. There are no known implementations of this subtype and it SHOULD be ignored. The following format applies to this Subtype:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        View #               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            File Name... (variable)
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

**Figure 17: BGP_SYNC Subtype**

The File Name is terminated with a NULL (0) character.

### A.2.1.6. BGP_OPEN Subtype

TOC

The BGP_OPEN Subtype is used to encode BGP OPEN messages. The format of the MRT Message field for this Subtype is the same as the BGP_UPDATE, however, the last field contains the contents of the BGP OPEN message.

### A.2.1.7. BGP_NOTIFY Subtype

TOC

The BGP_NOTIFY Subtype is used to encode BGP NOTIFICATION messages. The format of the MRT Message field for this Subtype is the same as the BGP_UPDATE, however, the last field contains the contents of the BGP NOTIFICATION message.

### A.2.1.8. BGP_KEEPALIVE Subtype

TOC

The BGP_KEEPALIVE Subtype is used to encode BGP KEEPALIVE messages. The format of the MRT Message field for this Subtype is the same as the BGP_UPDATE, however, the last field contains no information.

### A.2.2. RIP Type

TOC

The RIP Type is used to export RIP protocol packets as defined in RFC 1723 (Malkin, G., “RIP Version 2 - Carrying Additional Information,”

November 1994.) [RFC1723]. The Subtype field is currently reserved for this Type and SHOULD be set to 0.
The format of the MRT Message field for the RIP Type is as follows:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Peer IP address                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Local IP address                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   RIP Message Contents (variable)
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

**Figure 18: RIP Type**

### A.2.3.  IDRP Type

TOC

The IDRP Type is used to export Inter-Domain-Routing Protocol (IDRP) protocol information as defined in the ISO/IEC 10747 standard. The Subtype field is unused. This Type is deprecated due to lack of deployment of IDRP.

### A.2.4.  RIPNG Type

TOC

The RIPNG Type is used to export RIPNG protocol packets as defined in RFC 2080 (Malkin, G. and R. Minnear, “RIPng for IPv6,” January 1997.) [RFC2080]. The RIPNG protocol updates the RIP protocol to support IPv6. The Subtype field is currently reserved for this Type and SHOULD be set to 0.
The format of the MRT Message field for the RIPNG Type is as follows:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
~                        Peer IPv6 address                      ~
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
~                        Local IPv6 address                     ~
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  RIPNG Message Contents (variable)
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

**Figure 19: RIPNG Type**

### A.2.5. BGP4PLUS and BGP4PLUS_01 Types

TOC

The BGP4PLUS and BGP4PLUS_01 Types were defined to support IPv6 BGP routing information. The BGP4PLUS Type was specified based on the initial Internet Draft for Multiprotocol Extensions to BGP-4. The BGP4PLUS_01 Type was specified to correspond to the -01 revision of this Internet Draft. The two Types share the same definitions in terms of their MRT format specifications.
The Subtype field definitions are shared with the BGP Type, however, the address fields in the BGP_UPDATE, BGP_OPEN, BGP_NOTIFY, BGP_KEEPALIVE, and BGP_STATE_CHANGE Subtype messages are extended to 16 octets for IPv6 addresses. As with the BGP Type, the BGP4PLUS and BGP4PLUS_01 Types are deprecated as they superseded by the BGP4MP Type.

### A.2.6. Deprecated BGP4MP Subtypes

TOC

The following two subtypes of the BGP4MP Type are considered to be deprecated.

```
2    BGP4MP_ENTRY
3    BGP4MP_SNAPSHOT
```

### A.2.6.1. BGP4MP_ENTRY Subtype

TOC

This Subtype is similar to the TABLE_DUMP Type and is used to record RIB table entries. It extends the TABLE_DUMP Type to include true multiprotocol support. However, this Type does not support 4-Byte AS numbers and has not been widely implemented. This Type is deprecated in favor of the TABLE_DUMP_V2 which includes 4-Byte AS number support and a more compact format.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Peer AS number        |        Local AS number        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        Interface Index        |        Address Family         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Peer IP address (variable)               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Local IP address (variable)              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           View #              |             Status            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Time last change                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        Address Family         |    SAFI       | Next-Hop-Len  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     Next Hop Address (variable)               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Prefix Length |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     Address Prefix (variable)                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       Attribute Length        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    BGP Attribute... (variable)
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

**Figure 20: BGP4MP_ENTRY Subtype**

TOC

#### A.2.6.2. BGP4MP_SNAPSHOT Subtype

This Subtype was intended to convey a system file name where BGP4MP_ENTRY messages should be recorded. It is similar to the BGP_SYNC message Subtype and is deprecated.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        View #                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            File Name... (variable)
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

**Figure 21: BGP4MP_SNAPSHOT Subtype**

### Authors' Addresses

TOC


| | |
|---|---|
| | Larry Blunk |
| | Merit Network |
| Email: | ljb@merit.edu |
| | |
| | Manish Karir |
| | Merit Network |
| Email: | mkarir@merit.edu |
| | |
| | Craig Labovitz |
| | Arbor Networks |
| Email: | labovit@arbor.net |