

Network Working Group	L. Blunk	
Internet-Draft	M. Karir	
Intended status: Standards Track	Merit Network	
Expires: August 28, 2008	C. Labovitz	
	Arbor Networks	
	February 25, 2008	

[TOC](#)

MRT routing information export format draft-ietf-grow-mrt-07.txt

Status of this Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with Section 6 of BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/1id-abstracts.txt>.

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>.

This Internet-Draft will expire on August 28, 2008.

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 its name. The format can be used to export routing protocol messages, state changes, and routing information base contents.

Table of Contents

- [1.](#) Requirements notation
- [2.](#) Introduction

- [3.](#) Basic MRT Format
 - [4.](#) MRT Informational Types
 - [4.1.](#) NULL Type
 - [4.2.](#) START Type
 - [4.3.](#) DIE Type
 - [4.4.](#) I_AM_DEAD Type
 - [4.5.](#) PEER_DOWN Type
 - [5.](#) MRT Routing Information Types
 - [5.1.](#) BGP Type
 - [5.1.1.](#) BGP_NULL Subtype
 - [5.1.2.](#) BGP_UPDATE Subtype
 - [5.1.3.](#) BGP_PREF_UPDATE Subtype
 - [5.1.4.](#) BGP_STATE_CHANGE Subtype
 - [5.1.5.](#) BGP_SYNC Subtype
 - [5.1.6.](#) BGP_OPEN Subtype
 - [5.1.7.](#) BGP_NOTIFY Subtype
 - [5.1.8.](#) BGP_KEEPALIVE Subtype
 - [5.2.](#) RIP Type
 - [5.3.](#) IDRP Type
 - [5.4.](#) RIPNG Type
 - [5.5.](#) BGP4PLUS and BGP4PLUS_01 Types
 - [5.6.](#) OSPF Type
 - [5.7.](#) TABLE_DUMP Type
 - [5.8.](#) TABLE_DUMP_V2 Type
 - [5.9.](#) BGP4MP Type
 - [5.9.1.](#) BGP4MP_STATE_CHANGE Subtype
 - [5.9.2.](#) BGP4MP_MESSAGE Subtype
 - [5.9.3.](#) BGP4MP_ENTRY Subtype
 - [5.9.4.](#) BGP4MP_SNAPSHOT Subtype
 - [5.9.5.](#) BGP4MP_STATE_CHANGE_AS4 Subtype
 - [5.9.6.](#) BGP4MP_MESSAGE_AS4 Subtype
 - [5.10.](#) BGP4MP_ET Type
 - [5.11.](#) ISIS Type
 - [5.12.](#) ISIS_ET Type
 - [5.13.](#) OSPFv3 Type
 - [5.14.](#) OSPFv3_ET Type
 - [6.](#) IANA Considerations
 - [6.1.](#) Type Codes
 - [6.2.](#) Subtype Codes
 - [7.](#) Security Considerations
 - [8.](#) References
 - [8.1.](#) Normative References
 - [8.2.](#) Informative References
 - [§](#) Authors' Addresses
 - [§](#) Intellectual Property and Copyright Statements
-

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 it's 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.

Fields which contain multi-byte numeric values are encoded in network byte order from most significant byte to least significant byte. 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.

4. MRT Informational Types

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.

Two of these messages are considered potentially useful in implementations with a local repository. In particular, 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 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:

0	NULL	*DEPRECATED*
1	START	
2	DIE	*DEPRECATED*
3	I_AM_DEAD	
4	PEER_DOWN	*DEPRECATED*

4.1. NULL Type

[TOC](#)

The NULL Type message causes no operation and is deprecated.

4.2. START Type

[TOC](#)

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

4.3. DIE Type

[TOC](#)

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

4.4. I_AM_DEAD Type

[TOC](#)

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

4.5. 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. This Type is deprecated.

5. MRT Routing Information Types

[TOC](#)

The following Types are currently defined for the MRT format. Types 5-12 were defined in the MRT Toolkit package. The BGP4MP Type, number 16, was initially defined in the Zebra routing software package. The BGP4MP_ET, ISIS, and ISIS_ET Types were initially defined in the Sprint Labs Python Routing Toolkit (PyRT).

5	BGP	*DEPRECATED*
6	RIP	
7	IDRP	*DEPRECATED*
8	RIPNG	
9	BGP4PLUS	*DEPRECATED*
10	BGP4PLUS_01	*DEPRECATED*
11	OSPF	
12	TABLE_DUMP	
13	TABLE_DUMP_V2	
16	BGP4MP	
17	BGP4MP_ET	
32	ISIS	
33	ISIS_ET	
48	OSPFv3	
49	OSPFv3_ET	

[TOC](#)

5.1. BGP Type

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 are considered to be DEPRECATED by the BGP4MP Type.

The following BGP Subtypes are defined for the MRT BGP Type.

- ```
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
```

### 5.1.1. BGP\_NULL Subtype

## TOC

The BGP\_NULL Subtype is a reserved Subtype.

### 5.1.2. 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:

[illegible]

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.

---

### 5.1.3. BGP\_PREF\_UPDATE Subtype

[TOC](#)

The BGP\_PREF\_UPDATE Subtype is not defined.

---

### 5.1.4. BGP\_STATE\_CHANGE Subtype

[TOC](#)

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 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 |
+--+

```

---

[TOC](#)

#### 5.1.5. BGP\_SYNC Subtype

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)
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

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

---

#### 5.1.6. 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.

---

#### 5.1.7. 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.

---

#### 5.1.8. BGP\_KEEPAIVE Subtype

[TOC](#)

The BGP\_KEEPAIVE Subtype is used to encode BGP KEEPAIVE 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.

---

[TOC](#)

## 5.2. RIP Type

The RIP Type is used to export RIP protocol packets as defined in [RFC 1058 \(Hedrick, C., "Routing Information Protocol," June 1988.\)](#) [RFC1058]. 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) |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
```

---

## 5.3. 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.

---

## 5.4. 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 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 IP address |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| Local IP address |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| RIPNG Message Contents (variable) |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
```

---

## 5.5. 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\_KEEPAALIVE, 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.

---

## 5.6. 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) |
+--+
```

---

## 5.7. 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) |
+--+

```

The View field is normally 0 and is intended for cases where an implementation may have multiple RIB views (such as a route server). 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 not used 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 AS number of the peer.

Attribute length is the length of Attribute field and is 2-octets. The Attribute field contains the attribute information for the RIB entry.

## 5.8. TABLE\_DUMP\_V2 Type

[TOC](#)

The TABLE\_DUMP\_V2 Type updates the TABLE\_DUMP Type to include 32BIT 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. 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 |
+--+

```

The format of the peer entries is shown below. The PEER\_INDEX\_TABLE record contains Peer Count 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) |
+--+

```

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 field is 16 bits, set when it's 32 bits

```

The records which follow the PEER\_INDEX\_TABLE record constitute 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 |
+--+

```

The RIB\_GENERIC header is shown below. 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 |
+--+

```

The RIB 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)
+--+

```

There is one exception to the encoding of BGP attributes for the BGP MP\_REACH\_NLRI attribute (BGP Type Code 14) [RFC 4760]. 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.

## 5.9. 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          |              |
| 2 | BGP4MP_ENTRY            | *DEPRECATED* |
| 3 | BGP4MP_SNAPSHOT         | *DEPRECATED* |
| 4 | BGP4MP_STATE_CHANGE_AS4 |              |
| 5 | BGP4MP_MESSAGE_AS4      |              |

### 5.9.1. BGP4MP\_STATE\_CHANGE Subtype

TOC

This record is used to encode state changes in the BGP finite state machine. As with the BGP\_STATE\_CHANGE Subtype, the BGP FSM states are encoded in the Old State and New State fields to indicate the previous and current state. 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 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| Peer AS number              |   |   |   |   |   |   |   |   |   | Local AS number |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| Interface Index             |   |   |   |   |   |   |   |   |   | Address Family  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| Peer IP address (variable)  |   |   |   |   |   |   |   |   |   |                 |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| Local IP address (variable) |   |   |   |   |   |   |   |   |   |                 |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| Old State                   |   |   |   |   |   |   |   |   |   | New State       |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |

While BGP4MP\_STATE\_CHANGE message is similar to the BGP\_STATE\_CHANGE message, it also includes interface index and Address Family fields. As with the BGP\_STATE\_CHANGE message, 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 with the numeric values for these states defined in the BGP\_STATE\_CHANGE section. 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 address fields. At present, the following AFI Types are supported:

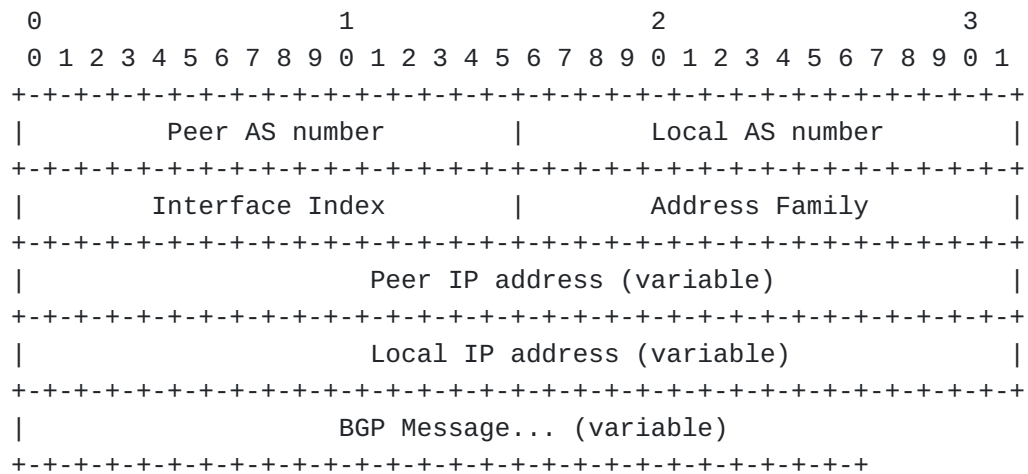
```
1 AFI_IPv4
2 AFI IPv6
```

---

### 5.9.2. BGP4MP\_MESSAGE Subtype

[TOC](#)

This Subtype is used to encode BGP Messages. It is similar to the BGP\_UPDATE Subtype, except that it can be used to encode any Type of message (not simply BGP UPDATES). In order to determine the BGP message Type, the entire BGP message is included in the BGP Message field. This includes 16-octet marker, 2-octet length, and 1-octet type fields. Note that the BGP4MP\_MESSAGE Subtype does not support 32BIT AS numbers nor IPv6 addresses. The BGP4MP\_MESSAGE\_AS4 Subtype updates the BGP4MP\_MESSAGE Subtype in order to support these. The BGP4MP\_MESSAGE fields are shown below:



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 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.

---

[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 32BIT AS numbers and has not been widely implemented. This Type is deprecated in favor of the TABLE\_DUMP\_V2 which includes 32BIT 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)      |   |   |   |   |   |   |   |   |   |              |   |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |  |  |

TOC

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.

This Subtype updates the BGP4MP\_MESSAGE Subtype to support 32BIT Autonomous System numbers. The BGP4MP\_MESSAGE\_AS4 Subtype is otherwise identical to the BGP4MP\_MESSAGE Subtype. The BGP4MP\_MESSAGE\_AS4 fields are shown below:



### 5.11. ISIS Type

This Type was initially defined in the Sprint Labs Python Routing and 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.

---

### 5.12. 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.

---

### 5.13. OSPFv3 Type

[TOC](#)

The OSPFv3 Type extends the original OSPF Type to support IPv6 addresses for the OSPFv3 protocol as defined in [RFC 2740 \(Coltun, R., Ferguson, D., and J. Moy, "OSPF for IPv6," December 1999.\)](#) [RFC2740]. 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) |
+--+
```

### 5.14. 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. 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 2434 \(Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs," October 1998.\)](#) [RFC2434].

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".

---

### 6.1. Type Codes

[TOC](#)

Type Codes have a range from 0 to 65535, of which 0-64 have been allocated. New Type Codes MUST be allocated starting at 65. Type Codes 65 - 32767 are to be assigned by IETF Consensus. Type Codes 32768 - 65535 are assigned based on Specification Required.

---

### 6.2. Subtype Codes

[TOC](#)

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. As Subtype Codes are specific to Type Codes, new numbers must be unique for the particular Type Code to which the Subtype applies. Subtype Codes specific to the Type Codes 0 - 32767 are assigned by IETF Consensus. Subtype Codes specific to Type Codes 32768 - 65535 are assigned based on Specification Required.

---

[TOC](#)

## 7. Security Considerations

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.

---

## 8. References

[TOC](#)

### 8.1. Normative References

[TOC](#)

|           |                                                                                                                                                                                                                                                        |
|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| [RFC1058] | Hedrick, C., " <a href="#">Routing Information Protocol</a> ," RFC 1058, June 1988 ( <a href="#">TXT</a> ).                                                                                                                                            |
| [RFC1195] | Callon, R., " <a href="#">Use of OSI IS-IS for routing in TCP/IP and dual environments</a> ," RFC 1195, December 1990 ( <a href="#">TXT</a> , <a href="#">PS</a> ).                                                                                    |
| [RFC2080] | <a href="#">Malkin, G.</a> and <a href="#">R. Minnear</a> , " <a href="#">RIPng for IPv6</a> ," RFC 2080, January 1997 ( <a href="#">TXT</a> ).                                                                                                        |
| [RFC2119] | <a href="#">Bradner, S.</a> , " <a href="#">Key words for use in RFCs to Indicate Requirement Levels</a> ," BCP 14, RFC 2119, March 1997 ( <a href="#">TXT</a> , <a href="#">HTML</a> , <a href="#">XML</a> ).                                         |
| [RFC2328] | <a href="#">Moy, J.</a> , " <a href="#">OSPF Version 2</a> ," STD 54, RFC 2328, April 1998 ( <a href="#">TXT</a> , <a href="#">HTML</a> , <a href="#">XML</a> ).                                                                                       |
| [RFC2434] | <a href="#">Narten, T.</a> and <a href="#">H. Alvestrand</a> , " <a href="#">Guidelines for Writing an IANA Considerations Section in RFCs</a> ," BCP 26, RFC 2434, October 1998 ( <a href="#">TXT</a> , <a href="#">HTML</a> , <a href="#">XML</a> ). |
| [RFC2740] | <a href="#">Coltun, R.</a> , <a href="#">Ferguson, D.</a> , and <a href="#">J. Moy</a> , " <a href="#">OSPF for IPv6</a> ," RFC 2740, December 1999 ( <a href="#">TXT</a> ).                                                                           |
| [RFC4271] | <a href="#">Rekhter, Y.</a> , <a href="#">Li, T.</a> , and <a href="#">S. Hares</a> , " <a href="#">A Border Gateway Protocol 4 (BGP-4)</a> ," RFC 4271, January 2006 ( <a href="#">TXT</a> ).                                                         |
| [RFC4760] | <a href="#">Bates, T.</a> , <a href="#">Chandra, R.</a> , <a href="#">Katz, D.</a> , and <a href="#">Y. Rekhter</a> , " <a href="#">Multiprotocol Extensions for BGP-4</a> ," RFC 4760, January 2007 ( <a href="#">TXT</a> ).                          |

### 8.2. Informative References

[TOC](#)

|                  |                                                                                                                    |
|------------------|--------------------------------------------------------------------------------------------------------------------|
| [MRT PROG GUIDE] | <a href="#">Labovitz, C.</a> , " <a href="#">MRT Programmer's Guide</a> ," November 1999 ( <a href="#">HTML</a> ). |
|------------------|--------------------------------------------------------------------------------------------------------------------|

---

## Authors' Addresses

[TOC](#)

|        |                                                          |
|--------|----------------------------------------------------------|
|        | Larry Blunk                                              |
|        | Merit Network                                            |
| Email: | <a href="mailto:ljb@merit.edu">ljb@merit.edu</a>         |
|        |                                                          |
|        | Manish Karir                                             |
|        | Merit Network                                            |
| Email: | <a href="mailto:mkarir@merit.edu">mkarir@merit.edu</a>   |
|        |                                                          |
|        | Craig Labovitz                                           |
|        | Arbor Networks                                           |
| Email: | <a href="mailto:labovit@arbor.net">labovit@arbor.net</a> |

---

## Full Copyright Statement

[TOC](#)

Copyright © The IETF Trust (2008).

This document is subject to the rights, licenses and restrictions contained in BCP 78, and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY, THE IETF TRUST AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

## Intellectual Property

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at <http://www.ietf.org/ipr>.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights

that may cover technology that may be required to implement this standard. Please address the information to the IETF at [ietf-ipr@ietf.org](mailto:ietf-ipr@ietf.org).