Network Working Group	L. Blunk
Internet-Draft	M. Karir
Intended status: Informational	Merit Network
Expires: October 22, 2011	C. Labovitz
	Arbor Networks
	April 20, 2011

MRT routing information export format
draft-ietf-grow-mrt-14.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

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 http://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 October 22, 2011.

Copyright Notice

Copyright (c) 2011 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 (http://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 Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

Table of Contents

1. Introduction

Researchers and engineers often wish to analyze network behavior by studying routing protocol transactions and routing information base snapshots. To this end, the MRT record 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. Two examples of large-scale MRT based BGP archival projects include the University of Oregon Route Views Project and the RIPE NCC Routing Information Service (RIS).

The MRT format was initially defined in the MRT Programmer's Guide *[MRT PROG GUIDE]*. Subsequent extensions were made in the the GNU Zebra software routing suite and the Sprint Advanced Technology Labs Python Routing Toolkit (PyRT). Further extensions may be introduced at a later date through additional definitions of the MRT Type field and Subtype fields.

A number of MRT record types listed in the MRT Programmer's Guide *[MRT PROG GUIDE]* are not known to have been implemented and, in some cases, were incompletely specified. Further, several types were employed in early MRT implementations, but saw limited use and were updated by improved versions. These types are considered to be deprecated and are documented in the Deprecated MRT Types *[app-deprecated]* section at the end of this document. The deprecated types consist of codes 0 through 10 inclusive. Some of the deprecated types may be of interest to researchers examining historical MRT format 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.

1.1. Specification of Requirements

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].

2. MRT Common Header

All MRT format records have a Common Header which consists of 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)
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Header Field Descriptions:

* Timestamp:

  -A 4-octet field whose integer value is the number of seconds, excluding leap seconds, elapsed since midnight proleptic Coordinated Universal Time (UTC). This representation of time is sometimes called "UNIX time" POSIX *[IEEE.P1003.1-1990]*. This time format cannot represent time values prior to January 1, 1970. The latest UTC time value that can be represented by a four-octet integer value is 03:14:07 on January 19, 2038, which is represented by the hexadecimal value 7FFFFFFF. Implementations which wish to create MRT records after this date will need to provide an alternate EPOCH time base for the Timestamp field. Mechanisms for indicating this alternate EPOCH are currently outside the scope of this document.

* 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 record 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.

3. Extended Timestamp MRT Header

Several MRT format record types support a variant type with an extended timestamp field. The purpose of this field is to support measurements at sub-second resolutions. This field, Microsecond Timestamp, contains an unsigned 32BIT offset value in microseconds which is added to the Timestamp field value. The Timestamp field remains as defined in the MRT Common Header. The Microsecond Timestamp immediately follows the length field in the MRT Common Header and precedes all other fields in the message. The Microsecond Timestamp is included in the computation of the length field value. The Extended Timestamp MRT 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                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Microsecond Timestamp                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Message... (variable)
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

4. MRT Types

The following MRT Types are currently defined for the MRT format. The MRT Types which contain the "_ET" suffix in their names identify those types which use an Extended Timestamp MRT Header. The subtype and message fields in these types remain as defined for the MRT Types of the same name without the "_ET" suffix.

```
11    OSPFv2
12    TABLE_DUMP
13    TABLE_DUMP_V2
16    BGP4MP
17    BGP4MP_ET
32    ISIS
33    ISIS_ET
48    OSPFv3
49    OSPFv3_ET
```

4.1. OSPFv2 Type

This Type supports the OSPFv2 Protocol as defined in RFC 2328 *[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 OSPFv2 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)
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The Remote IP address field contains source IPv4 *[RFC0791]* address from the IP header of the OSPF message. The Local IP address contains the destination IPv4 address from the IP header. The OSPF Message Contents field contains the complete contents of the OSPF packet following the IP header.

4.2. TABLE_DUMP Type

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. It is RECOMMENDED that new MRT encoding implementations use the TABLE_DUMP_V2 Type (see below) instead of the TABLE_DUMP Type due to limitations in this type. However, due to the significant volume of historical data encoded with this type, MRT decoding applications MAY wish to support this type.
The Subtype field is used to encode whether the RIB entry contains IPv4 or IPv6 *[RFC2460]* 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). 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 record. The AFI_IPv4 Subtype value specifies an Address Family *[IANA-AF]* Identifier (AFI) type of IPv4. It specifies a prefix field length of 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 Autonomous System (AS) number of the peer.

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. The AS_PATH attribute MUST only consist of 2-Byte AS numbers. The TABLE_DUMP_V2 supports 4-Byte AS numbers in the AS_PATH attribute.

4.3. TABLE_DUMP_V2 Type

The TABLE_DUMP_V2 Type updates the TABLE_DUMP Type to include 4-Byte Autonomous System Number (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 Network Layer Reachability Information (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
```

4.3.1. PEER_INDEX_TABLE Subtype

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 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 *[RFC3629]*.

```
 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)
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

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)                     |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

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 identified by the A and I bits, respectively, below.

```
    0 1 2 3 4 5 6 7
   +-+-+-+-+-+-+-+-+
   | | | | | | |A|I|
   +-+-+-+-+-+-+-+-+

   Bit 6: Peer AS number size:  0 = 16 bits, 1 = 32 bits
   Bit 7: Peer IP Address family:  0 = IPv4,  1 = IPv6
```

The MRT records which follow the PEER_INDEX_TABLE MRT record consist of the subtypes listed below and contain the actual RIB table entries. They include a header which specifies a sequence number, a NLRI field, and a count of the number of RIB entries contained within the record.

4.3.2. AFI/SAFI specific RIB Subtypes

The AFI/SAFI specific RIB Subtypes consist of the RIB_IPV4_UNICAST, RIB_IPV4_MULTICAST, RIB_IPV6_UNICAST, and RIB_IPV6_MULTICAST Subtypes. These specific RIB table entries are given their own MRT TABLE_DUMP_V2 subtypes as they are the most common type of RIB table instances and providing specific MRT subtypes for them permits more compact encodings. These subtypes permit a single MRT record to encode multiple RIB table entries for a single prefix. 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. 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)
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

4.3.3. RIB_GENERIC Subtype

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 consists of an AFI, Subsequent AFI (SAFI) 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)
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

4.3.4. RIB Entries

The RIB Entries 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. All AS numbers in the AS_PATH attribute MUST be encoded as 4-Byte AS numbers.

```
 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 *[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.

4.4. BGP4MP Type

This Type was initially defined in the Zebra software package for the BGP protocol with multiprotocol extension support as defined by RFC 4760 *[RFC4760]*. 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
```

4.4.1. BGP4MP_STATE_CHANGE Subtype

This message is used to encode state changes in the BGP finite state machine. The BGP Finite State Machine (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            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The FSM states are defined in RFC 4271 *[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
```

4.4.2. BGP4MP_MESSAGE Subtype

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. The BGP4MP_MESSAGE Subtype does not support 4-Byte AS numbers. 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)
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

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
```

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 SHALL be encoded in the BGP4MP_MESSAGE Subtype.

4.4.3. BGP4MP_MESSAGE_AS4 Subtype

This Subtype updates the BGP4MP_MESSAGE Subtype to support 4-Byte AS 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)
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

4.4.4. BGP4MP_STATE_CHANGE_AS4 Subtype

This Subtype updates the BGP4MP_STATE_CHANGE Subtype to support 4-Byte AS 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            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

4.4.5. BGP4MP_MESSAGE_LOCAL Subtype

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 BGP message and the Peer IP and AS fields refer to the recipient of the generated BGP messages.

4.4.6. 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.

4.5. ISIS Type

This Type supports the IS-IS routing protocol as defined in RFC 1195 *[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.

4.6. OSPFv3 Type

```
     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)
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The OSPFv3 Type extends the original OSPFv2 Type to support IPv6 addresses for the OSPFv3 protocol as defined in RFC 5340 *[RFC5340]*. The format of the MRT Message field for the OSPFv3 Type is as follows:

5. IANA Considerations

This document has no IANA actions.

6. 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.

Some information contained in an MRT data structure might be considered sensitive or private. For example, a BGP peer that sends a message to an MRT-enabled router might not expect that message to be shared beyond the AS to which it is sent. The proposed geolocation extension to MRT could reveal the location of an MRT router's peers [I- D.ietf-grow-geomrt]. An organization that intends to use the MRT structure to export routing information beyond the domain where it normally accessible (e.g., publishing MRT dumps for use by researchers) should verify with any peers whose information might be included, and possibly remove sensitive fields.

7. References

7.1. Normative References

, "

[IANA-AF]	Address Family Numbers", .
[RFC0791]	Postel, J., "Internet Protocol", STD 5, RFC 791, September 1981.
[RFC1195]	Callon, R., "Use of OSI IS-IS for routing in TCP/IP and dual environments", RFC 1195, December 1990.
[RFC2119]	Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
[RFC2328]	Moy, J., "OSPF Version 2", STD 54, RFC 2328, April 1998.
[RFC2460]	Deering, S.E. and R.M. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998.
[RFC3629]	Yergeau, F., "UTF-8, a transformation format of ISO 10646", STD 63, RFC 3629, November 2003.
[RFC4271]	Rekhter, Y., Li, T. and S. Hares, "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, January 2006.
[RFC4760]	Bates, T., Chandra, R., Katz, D. and Y. Rekhter, "Multiprotocol Extensions for BGP-4", RFC 4760, January 2007.
[RFC5340]	Coltun, R., Ferguson, D., Moy, J. and A. Lindem, "OSPF for IPv6", RFC 5340, July 2008.

7.2. Informative References

[RFC2080]	Malkin, G. and R. Minnear, "RIPng for IPv6", RFC 2080, January 1997.
[RFC2453]	Malkin, G.S., "RIP Version 2", STD 56, RFC 2453, November 1998.
[MRT PROG GUIDE]	Labovitz, C, "MRT Programmer's Guide", November 1999.
[IEEE.P1003.1-1990]	Institute of Electrical and Electronics Engineers, "P1003.1, Information Technology

| | Portable Operating System Interface (POSIX) Part 1: System Application Program Interface (API) [C Language], 1990.", IEEE Standard P1003.1, . |

Appendix A. MRT Encoding Examples

This appendix, which is not a normative reference, contains a MRT encoding examples.
The following example shows the encoding for a MRT record type of BGP4MP and subtype BGP4MP_MESSAGE_AS4. The Peer AS and Local AS numbers are encoded in 4 bytes fields due to the use of the BGP4MP_MESSAGE_AS4 subtype. The encoded BGP Update is shown in hexadecimal. The AS numbers in the ASPATH in the BGP Update are encoded as 4 byte values in accord with the MRT BGP4MP_MESSAGE_AS4 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
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |    Timestamp = 1300475700 epoch sec (2011-03-18 19:15:00)     |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |          Type = 16            |         Subtype = 4           |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                           Length = 82                         |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                         Peer AS = 64496                       |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                        Local AS = 64497                       |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |     Interface Index = 0       |     Address Family  = 1       |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                Peer IP address = 192.0.2.85                   |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |               Local IP address = 198.51.100.4                 |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |  BGP Update =

            ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff
            00 3e 02 00 00 00 1f 40 01 01 02 40 02 0e 02 03
            00 00 fb f0 00 00 fb ff 00 00 fb f6 40 03 04 c6
            33 64 55 c0 08 04 fb f0 00 0e 18 cb 00 71
```

The contents of the BGP Update Message above are as follows:

```
  ORIGIN: INCOMPLETE
  ASPATH: 64496 64511 64502
  NEXT_HOP: 198.51.100.188
  COMMUNITY: 64496:14
  NLRI: 203.0.113.0/24
```

The following example displays the encoding for a MRT record type of TABLE_DUMP_V2 and subtype PEER_INDEX_TABLE. The table in this example contains 2 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
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |    Timestamp = 1300475700 epoch sec (2011-03-18 19:15:00)     |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |          Type = 13            |         Subtype = 1           |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                          Length = 34                          |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |               Collector BGP ID = 198.51.100.4                 |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |     View Name Length = 0      |       Peer Count = 2          |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |  Peer Type = 2  |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                   Peer BGP ID  = 198.51.100.5                 |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                 Peer IP address = 198.51.100.5                |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                        Peer AS = 65541                        |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |  Peer Type = 2  |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                   Peer BGP ID  = 192.0.2.33                   |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                 Peer IP address = 192.0.2.33                  |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                        Peer AS = 65542                        |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The following example displays the encoding for a MRT record type of TABLE_DUMP_V2 and subtype RIB_IPV6_UNICAST. This entry applies to the NLRI prefix of 2001:0DB8::/32. There is a single entry for this prefix. The entry applies to the peer identified by index location 15 in a preceding MRT PEER_INDEX_TABLE 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
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |    Timestamp = 1300475700 epoch sec (2011-03-18 19:15:00)     |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |          Type = 13            |         Subtype = 4           |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                             Length = 87                       |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                        Sequence number = 42                   |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      | Preflen = 32  |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                Prefix  =  2001:0DB8::/32                      |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |    Entry Count = 1            |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |    Peer Index =  15           |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |Originated Time = 1300475700 epoch sec (2011-03-18 19:15:00)   |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |   Attribute Length  =  68     |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |   BGP Path Attributes =

             40 01 01 00 50 02 00 0e 02 03 00 00 fb f0 00 00
             fb ff 00 00 fb f6 80 0e 2b 00 02 01 20 20 01 0d
             b8 00 0d 00 ff 00 00 00 00 00 00 01 87 fe 80 00
             00 00 00 00 00 02 12 f2 ff fe 9f 1b 00 00 00 20
             20 01 0d b8
```

The contents of the BGP Path Attribute field above are as follows:

ORIGIN: IGP
ASPATH: 64496 64511 64502
MP_REACH_NLRI(IPv6 Unicast)
NEXT_HOP: 2001:db8:d:ff::187
NEXT_HOP: fe80::212:f2ff:fe9f:1b00
NLRI: 2001:0DB8::/32

Appendix B. Deprecated MRT Types

This Appendix lists deprecated MRT types. These types are documented for informational purposes.

Appendix B.1. Deprecated MRT Informational Types

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

The message field MAY contain an OPTIONAL string for diagnostic purposes. The message string encoding MUST follow the UTF-8 transformation format *[RFC3629]*. The Subtype field is unused for these Types and SHOULD be set to 0.

The MRT Informational Types are defined below:

```
0    NULL
1    START
2    DIE
3    I_AM_DEAD
4    PEER_DOWN
```

Appendix B.1.1. NULL Type

The NULL Type message causes no operation.

Appendix B.1.2. START Type

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

Appendix B.1.3. DIE Type

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

Appendix B.1.4. I_AM_DEAD Type

An I_AM_DEAD MRT record indicates that a collector has shut down and has stopped generating MRT records.

Appendix B.1.5. PEER_DOWN Type

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.

Appendix B.2. Other Deprecated MRT Types

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

Appendix B.2.1. BGP Type

The BGP Type indicates the Message field contains BGP routing information. The BGP routing protocol is defined in RFC 4271 *[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
```

Appendix B.2.1.1. BGP_NULL Subtype

The BGP_NULL Subtype is a reserved Subtype.

Appendix B.2.1.2. BGP_UPDATE Subtype

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)
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

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.

Appendix B.2.1.3. BGP_PREF_UPDATE Subtype

The BGP_PREF_UPDATE Subtype is not defined.

Appendix B.2.1.4. BGP_STATE_CHANGE Subtype

The BGP_STATE_CHANGE Subtype is used to reflect changes in the BGP finite state machine. These FSM states are defined in RFC 4271 *[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           |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Appendix B.2.1.5. BGP_SYNC Subtype

The BGP_SYNC Subtype was intended to convey a system file name where BGP Table Dump messages MAY be recorded. The View # was to correspond to the View # provided in the TABLE_DUMP Type records. 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.

Appendix B.2.1.6. BGP_OPEN Subtype

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.

Appendix B.2.1.7. BGP_NOTIFY Subtype

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.

Appendix B.2.1.8. BGP_KEEPALIVE Subtype

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.

Appendix B.2.2. RIP Type

The RIP Type is used to export RIP protocol packets as defined in RFC 2453 *[RFC2453]*. 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)
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Appendix B.2.3. IDRP Type

The IDRP Type was intended to be used to export Inter-Domain-Routing Protocol (IDRP) protocol information as defined in the ISO/IEC 10747 standard. However, this Type has seen no known use and there are no details on protocol encoding for this Type.

Appendix B.2.4. RIPNG Type

The RIPNG Type is used to export RIPNG protocol packets as defined in RFC 2080 *[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)
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Appendix B.2.5. BGP4PLUS and BGP4PLUS_01 Types

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 records 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.

Appendix B.2.6. Deprecated BGP4MP Subtypes

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

```
    2    BGP4MP_ENTRY
    3    BGP4MP_SNAPSHOT
```

Appendix B.2.6.1. BGP4MP_ENTRY Subtype

This Subtype is similar to the TABLE_DUMP Type and is used to record RIB table entries. It was intended to include true multiprotocol support. However, this Subtype does not support 4-Byte AS numbers and has not been widely implemented.

```
 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)
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Appendix B.2.6.2. BGP4MP_SNAPSHOT Subtype

This Subtype was intended to convey a system file name where BGP4MP_ENTRY records MAY be recorded. It is similar to the BGP_SYNC 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)
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Appendix C. Acknowledgements

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.

Authors' Addresses

Larry Blunk Blunk Merit Network EMail: ljb@merit.edu

Manish Karir Karir Merit Network EMail: mkarir@merit.edu

Craig Labovitz Labovitz Arbor Networks EMail: labovit@arbor.net