     Multi-Threaded Routing Toolkit (MRT) Routing Information Export Format

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 is an Internet Standards Track document.

   This document is a product of the Internet Engineering Task Force
   (IETF). It represents the consensus of the IETF community. It has
   received public review and has been approved for publication by the
   Internet Engineering Steering Group (IESG). Further information on
   Internet Standards is available in Section 2 of RFC 5741.

   Information about the current status of this document, any errata,
   and how to provide feedback on it may be obtained at
   http://www.rfc-editor.org/info/rfc6396.

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 have 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 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
      (Appendix B) 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 that 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 that 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)                    
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
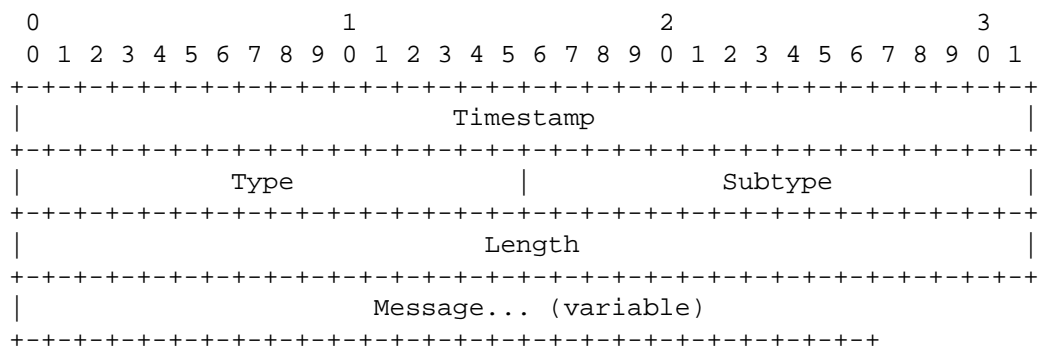
Figure 1: MRT Common Header

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]. This time format
      cannot represent time values prior to January 1, 1970. The
      latest UTC time value that can be represented by a 4-octet
      integer value is 03:14:07 on January 19, 2038, which is
      represented by the hexadecimal value 7FFFFFFF. Implementations
      that 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)                     |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
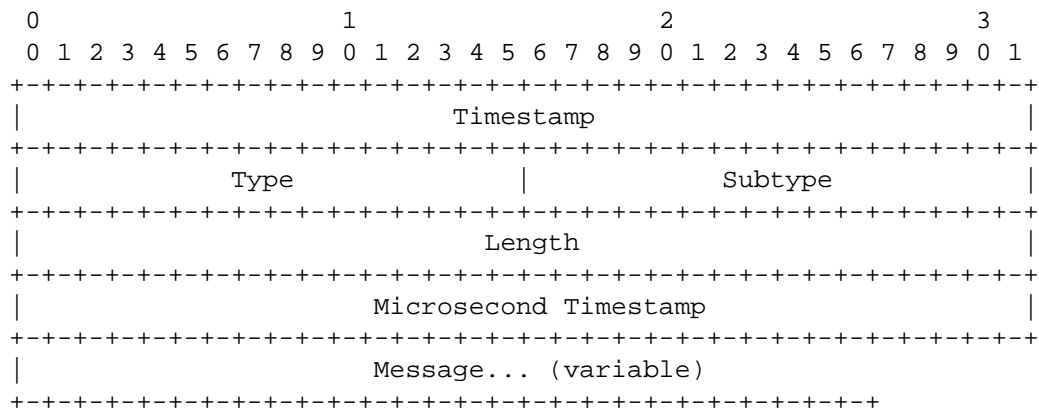
                Figure 2: Extended Timestamp MRT Header

4. MRT Types

   The following MRT Types are currently defined for the MRT format.
   The MRT Types that contain the "_ET" suffix in their names identify
   those types that 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]. It is used to encode the exchange of OSPF protocol
   packets.

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)
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
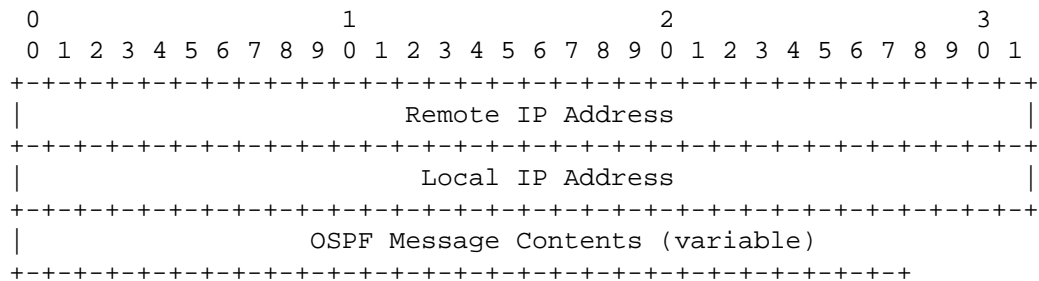
Figure 3: OSPFv2 Type

The Remote IP Address field contains the 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 Number          |        Sequence Number        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Prefix (variable)                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Prefix Length |    Status     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Originated Time                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   Peer IP Address (variable)                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Peer AS             |        Attribute Length       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     BGP Attribute... (variable)               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
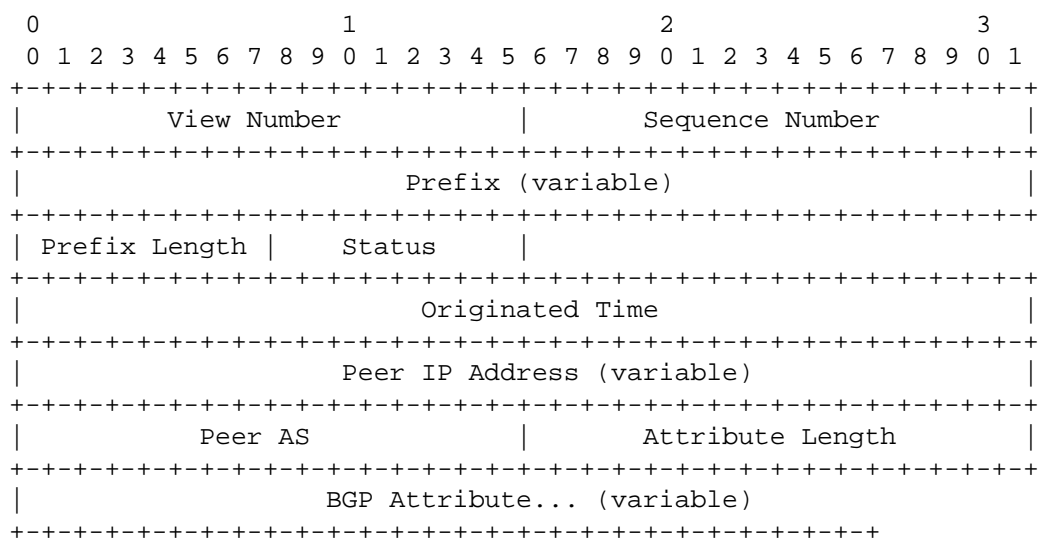
Figure 4: TABLE_DUMP Type

The View Number 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 View Number field to distinguish entries
from each view. The Sequence Number field is a simple incremental
counter for each RIB entry. A typical RIB dump will exceed the
16-bit bounds of this counter, and an 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
Identifier (AFI) type of IPv4 [IANA-AF]. 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 Address field is the IP address of the peer that 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 the 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 is
used to encode RIB table entries. This series of MRT records uses
subtypes 2-6 and is separate from the PEER_INDEX_TABLE MRT record
itself and includes 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)
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
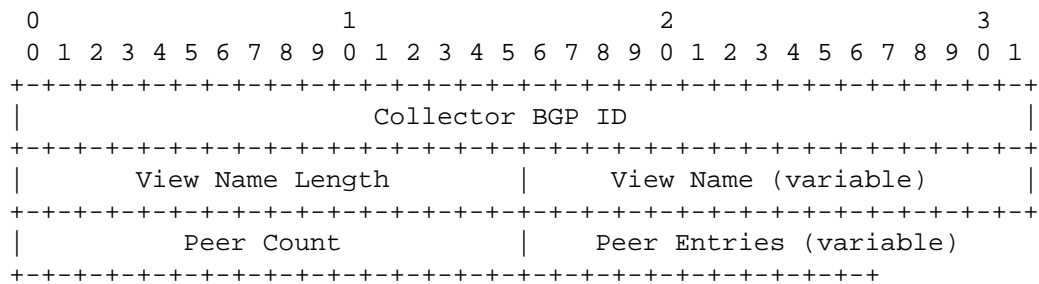
                  Figure 5: PEER_INDEX_TABLE Subtype

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

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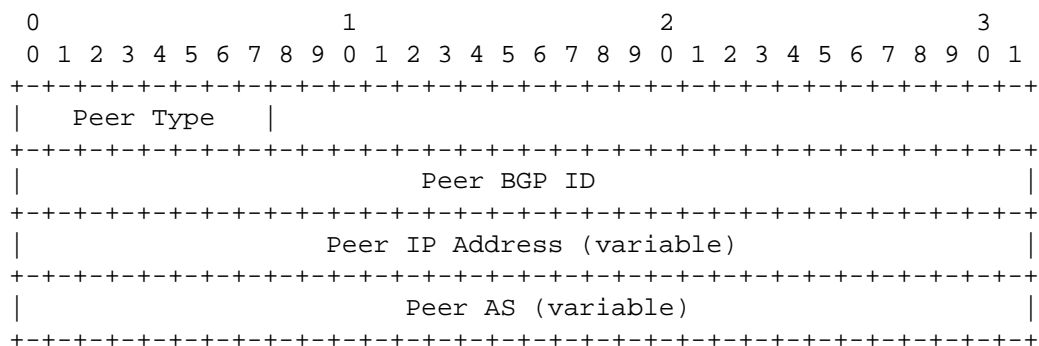
                       Figure 6: Peer Entries

The Peer Type, Peer BGP ID, Peer IP Address, 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 that 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|
 +-+-+-+-+-+-+-+-+
```
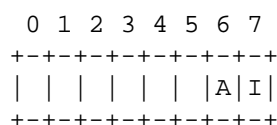
    Bit 6: Peer AS number size: 0 = 16 bits, 1 = 32 bits
    Bit 7: Peer IP Address family: 0 = IPv4, 1 = IPv6

                      Figure 7: Peer Type Field

The MRT records that follow the PEER_INDEX_TABLE MRT record consist
of the subtypes listed below and contain the actual RIB table
entries. They include a header that specifies a sequence number, an
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)
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
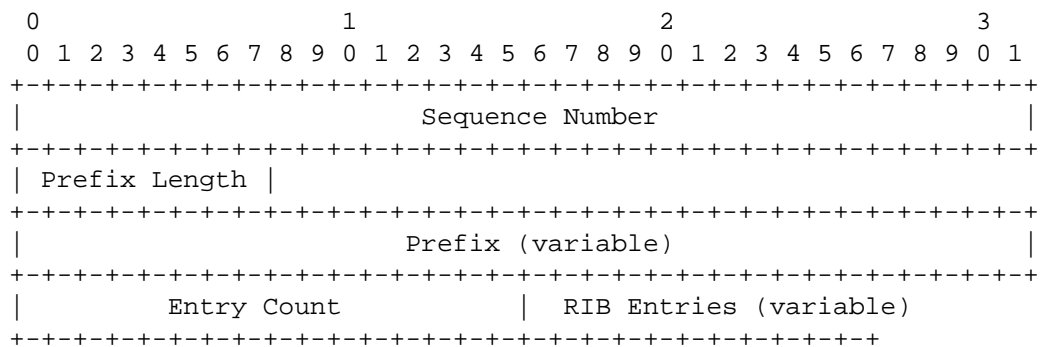
                      Figure 8: RIB Entry Header

4.3.3. RIB_GENERIC Subtype

The RIB_GENERIC header is shown below. It is used to cover RIB
entries that 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 that does not recognize particular AFI and SAFI
values SHOULD discard the remainder of the MRT record.

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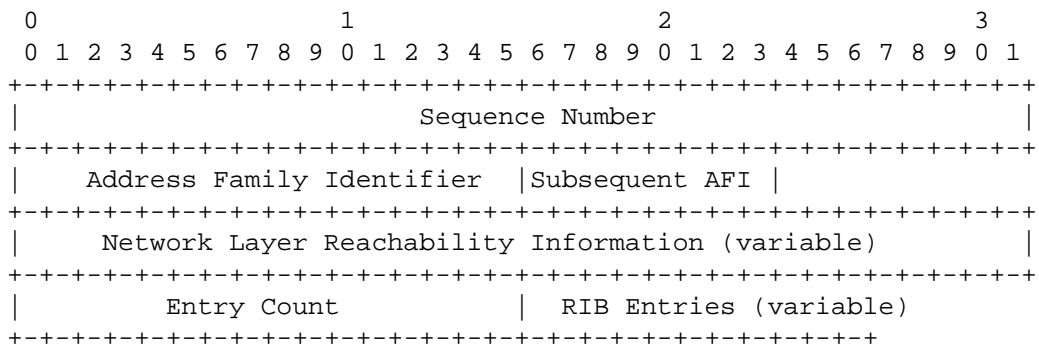
Figure 9: RIB_GENERIC Entry Header

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)
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
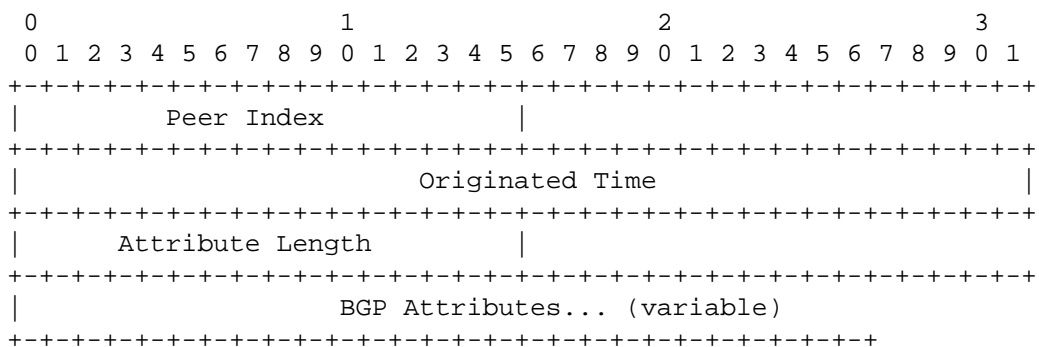
Figure 10: RIB Entries

   There is one exception to the encoding of BGP attributes for the BGP
   MP_REACH_NLRI attribute (BGP Type Code 14) [RFC4760]. Since the AFI,
   SAFI, and NLRI information is already encoded in the RIB Entry Header
   or RIB_GENERIC Entry 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 (FSM). The BGP FSM states are encoded in the Old State and
   New State fields to indicate the previous and current state. In some
   cases, the Peer AS Number may be undefined. In such cases, the value
   of this field MAY be set to zero. The format is illustrated below:

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

              Figure 11: BGP4MP_STATE_CHANGE Subtype

   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 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)                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
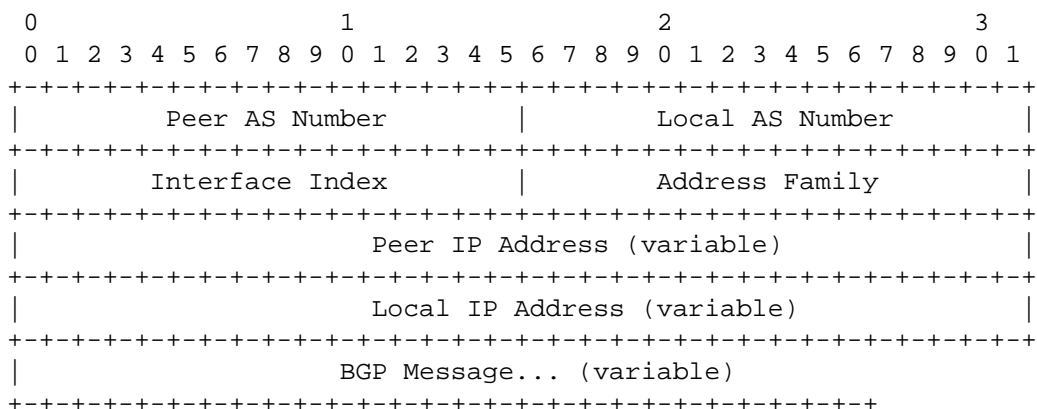
                   Figure 12: BGP4MP_MESSAGE Subtype

The Interface Index provides the interface number of the peering
session. The index value is OPTIONAL and MAY be zero if unknown or
unsupported. The Address Family indicates what types of addresses
are in the 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 that 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)                |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

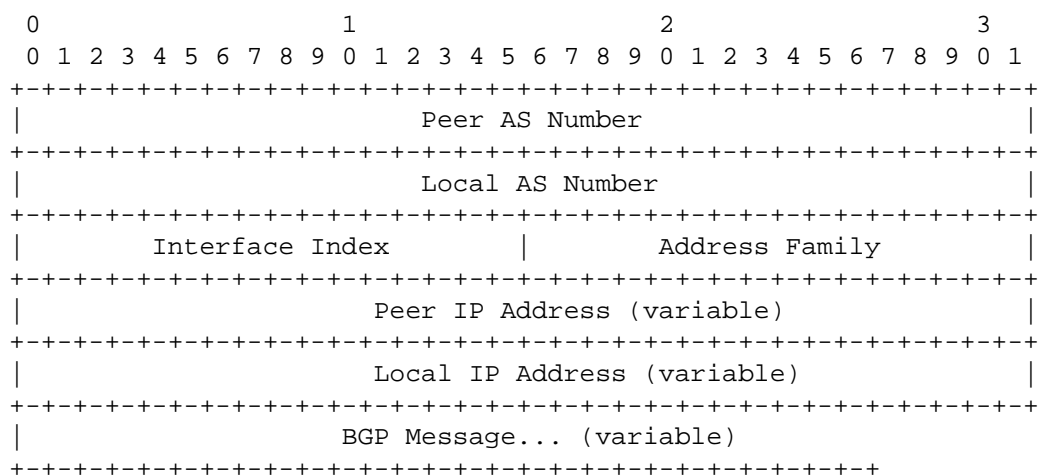Figure 13: BGP4MP_MESSAGE_AS4 Subtype

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 Number fields to 4 bytes, this subtype is

otherwise identical to the BGP4MP_STATE_CHANGE Subtype. The
BGP4MP_STATE_CHANGE_AS4 fields are shown below:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Peer AS Number                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Local AS Number                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        Interface Index        |         Address Family        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   Peer IP Address (variable)                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   Local IP Address (variable)                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Old State            |          New State            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
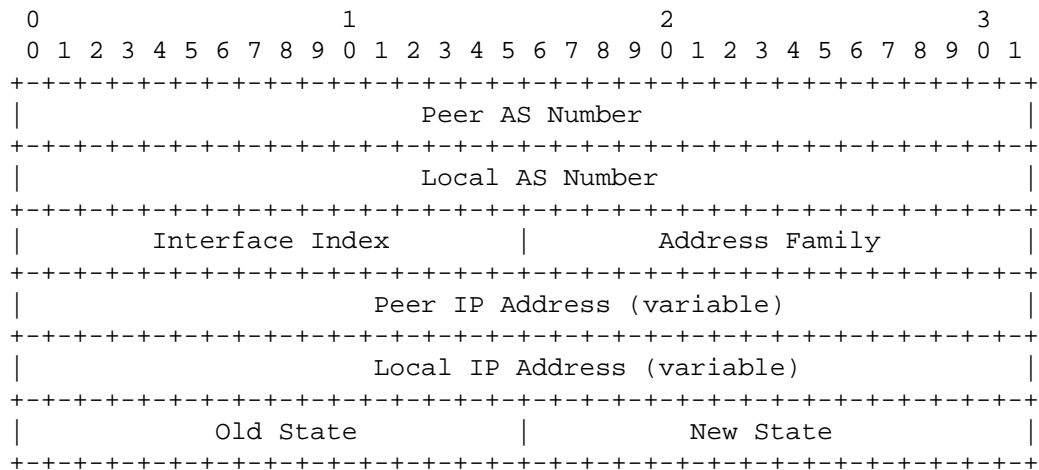
Figure 14: BGP4MP_STATE_CHANGE_AS4 Subtype

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 indicate 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 that 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 indicates 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

   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:

```
      0                   1                   2                   3
      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |         Address Family        |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                   Remote IP Address (variable)                |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                    Local IP Address (variable)                |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                   OSPF Message Contents (variable)            |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
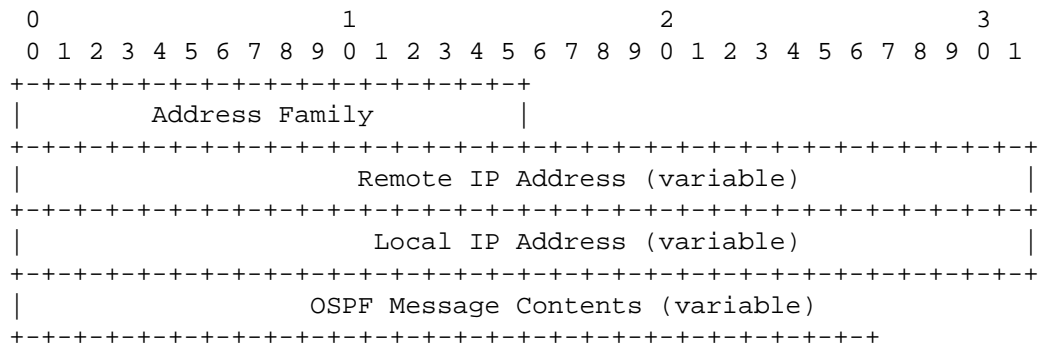
                       Figure 15: OSPFv3 Type

5. IANA Considerations

   This section provides guidance to the Internet Assigned Numbers
   Authority (IANA) regarding registration of values related to the MRT
   specification, in accordance with BCP 26, RFC 5226 [RFC5226].

   There are two name spaces in MRT that have been registered: Type
   Codes and Subtype Codes. Type Codes and Subtype Codes are each 16
   bits in length.

   MRT is not intended as a general-purpose specification for protocol
   information export, and allocations should not be made for purposes
   unrelated to routing protocol information export.

   The following policies are used here with the meanings defined in BCP
   26: "Specification Required", "IETF Consensus", "Experimental Use",
   "First Come First Served". Assignments consist of a name and the
   value.

5.1. Type Codes

   Type Codes have a range from 0 to 65535, of which 0-64 are reserved.
   New Type Codes MUST be allocated starting at 65. Type Codes 65-511
   are assigned by IETF Review. Type Codes 512-2047 are assigned based
   on Specification Required. Type Codes 2048-64511 are available on a

First Come First Served policy. Type Codes 64512 - 65534 are
available for Experimental Use. The Type Code Value 65535 is
reserved.

5.2. Subtype Codes

Subtype Codes have a range from 0 to 65535. Subtype definitions are
specific to a particular Type Code definition. New Subtype Code
definitions must reference an existing Type Code to which the Subtype
belongs. Subtype assignments follow the assignment rules for the
Type Codes to which they belong.

5.3. Defined Type Codes

This document defines the following message Type Codes:

```
        Name               Value      Definition
        ----               -----      ----------
        NULL               0          See Appendix B.1.1
        START              1          See Appendix B.1.2
        DIE                2          See Appendix B.1.3
        I_AM_DEAD          3          See Appendix B.1.4
        PEER_DOWN          4          See Appendix B.1.5
        BGP                5          See Appendix B.2.1
        RIP                6          See Appendix B.2.2
        IDRP               7          See Appendix B.2.3
        RIPNG              8          See Appendix B.2.4
        BGP4PLUS           9          See Appendix B.2.5
        BGP4PLUS_01        10         See Appendix B.2.5
        OSPFv2             11         See Section 4.1
        TABLE_DUMP         12         See Section 4.2
        TABLE_DUMP_V2      13         See Section 4.3
        BGP4MP             16         See Section 4.4
        BGP4MP_ET          17         See Section 4.4
        ISIS               32         See Section 4.5
        ISIS_ET            33         See Section 4.5
        OSPFv3             48         See Section 4.6
        OSPFv3_ET          49         See Section 4.6
```

5.4. Defined BGP, BGP4PLUS, and BGP4PLUS_01 Subtype Codes

   This document defines the following message Subtype Codes for the
   BGP, BGP4PLUS, and BGP4PLUS_01 Types:

            Name   Value   Definition
            ----   -----   ----------
            BGP_NULL   0   See Appendix B.2.1
            BGP_UPDATE   1   See Appendix B.2.1
            BGP_PREF_UPDATE   2   See Appendix B.2.1
            BGP_STATE_CHANGE   3   See Appendix B.2.1
            BGP_SYNC   4   See Appendix B.2.1
            BGP_OPEN   5   See Appendix B.2.1
            BGP_NOTIFY   6   See Appendix B.2.1
            BGP_KEEPALIVE   7   See Appendix B.2.1

5.5. Defined TABLE_DUMP Subtype Codes

   This document defines the following message Subtype Codes for the
   TABLE_DUMP Type:

            Name   Value   Definition
            ----   -----   ----------
            AFI_IPv4   1   See Section 4.2
            AFI_IPv6   2   See Section 4.2

5.6. Defined TABLE_DUMP_V2 Subtype Codes

   This document defines the following message Subtype Codes for the
   TABLE_DUMP_V2 Type:

            Name   Value   Definition
            ----   -----   ----------
            PEER_INDEX_TABLE   1   See Section 4.3
            RIB_IPV4_UNICAST   2   See Section 4.3
            RIB_IPV4_MULTICAST   3   See Section 4.3
            RIB_IPV6_UNICAST   4   See Section 4.3
            RIB_IPV6_MULTICAST   5   See Section 4.3
            RIB_GENERIC   6   See Section 4.3

5.7. Defined BGP4MP and BGP4MP_ET Subtype Codes

   This document defines the following message Subtype Codes for the
   BGP4MP Type:

            Name   Value   Definition
            ----   -----   ----------
            BGP4MP_STATE_CHANGE   0   See Section 4.4
            BGP4MP_MESSAGE   1   See Section 4.4
            BGP4MP_ENTRY   2   See Section 4.4
            BGP4MP_SNAPSHOT   3   See Section 4.4
            BGP4MP_MESSAGE_AS4   4   See Section 4.4
            BGP4MP_STATE_CHANGE_AS4   5   See Section 4.4
            BGP4MP_MESSAGE_LOCAL   6   See Section 4.4
            BGP4MP_MESSAGE_AS4_LOCAL 7   See Section 4.4

6. Security Considerations

   The MRT Format utilizes a structure that 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.

   Information that could be considered sensitive includes BGP peer IP
   addresses, BGP Next Hop IP addresses, and BGP Path Attributes. Such
   information could be useful to mount attacks against the BGP protocol
   and routing infrastructure. RFC 4272 [RFC4272] examines a number of
   weaknesses in the BGP protocol that could potentially be exploited.

   An organization that intends to use the MRT structure to export
   routing information beyond the domain where it is 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.

   The proposed geolocation extension to MRT could reveal the location
   of an MRT router's peers [GEOMRT].

7. References

7.1. Normative References

    [IANA-AF]   IANA, "Address Family Numbers",
                       <http://www.iana.org/numbers.html>.

    [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. and R. 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.

    [RFC5226]   Narten, T. and H. Alvestrand, "Guidelines for
                       Writing an IANA Considerations Section in RFCs",
                       BCP 26, RFC 5226, May 2008.

    [RFC5340]   Coltun, R., Ferguson, D., Moy, J., and A. Lindem,
                       "OSPF for IPv6", RFC 5340, July 2008.

7.2. Informative References

    [GEOMRT]   Manderson, T., "Multi-Threaded Routing Toolkit
                       (MRT) Border Gateway Protocol (BGP) Routing
                       Information Export Format with Geo-Location
                       Extensions", RFC 6397, October 2011.

   [MRT_PROG_GUIDE] Labovitz, C., "MRT Programmer's Guide",
                     November 1999, <http://www.merit.edu/
                     networkresearch/mrtprogrammer.pdf>.

   [POSIX]   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.

   [RFC2080]   Malkin, G. and R. Minnear, "RIPng for IPv6",
                     RFC 2080, January 1997.

   [RFC2453]   Malkin, G., "RIP Version 2", STD 56, RFC 2453,
                     November 1998.

   [RFC4272]   Murphy, S., "BGP Security Vulnerabilities
                     Analysis", RFC 4272, January 2006.

Appendix A. MRT Encoding Examples

   This appendix, which is not normative, contains MRT encoding
   examples.

   The following example shows the encoding for an MRT record type of
   BGP4MP and subtype BGP4MP_MESSAGE_AS4. The Peer AS and Local AS
   numbers are encoded in 4-byte 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
```

                 Figure 16: MRT BGP4MP_MESSAGE_AS4 Example

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
```

                    Figure 17: BGP Message Contents

The following example displays the encoding for an 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                          |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

                 Figure 18: MRT PEER_INDEX_TABLE Example

   The following example displays the encoding for an 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
```

               Figure 19: MRT RIB_IPV6_UNICAST Example

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
```

Figure 20: BGP Path Attribute Contents

Appendix B. Deprecated MRT Types

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

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
```

B.1.1. NULL Type

   The NULL Type message causes no operation.

B.1.2. START Type

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

B.1.3. DIE Type

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

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.

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 that duplicate this functionality.

B.2. Other Deprecated MRT Types

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

B.2.1. BGP Type

   The BGP Type indicates that 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
```

B.2.1.1. BGP_NULL Subtype

   The BGP_NULL Subtype is a reserved Subtype.

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

                    Figure 21: BGP_UPDATE Subtype

   The BGP UPDATE Contents include the entire BGP UPDATE message, which
   follows the BGP Message Header. The BGP Message Header itself is not
   included. The Peer AS Number and IP Address fields contain the AS
   number and IP address of the remote system that is 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 that is
   archiving the messages.

B.2.1.3. BGP_PREF_UPDATE Subtype

   The BGP_PREF_UPDATE Subtype is not defined.

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           |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
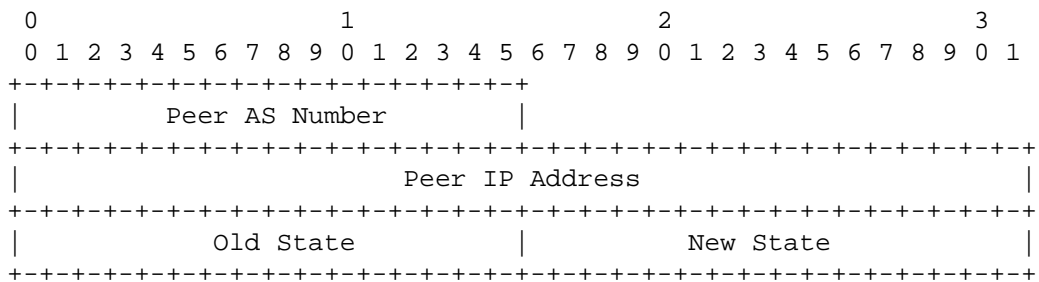
                  Figure 22: BGP_STATE_CHANGE Subtype

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 Number was to
   correspond to the View Number 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 Number           |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |             File Name... (variable)
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

                      Figure 23: BGP_SYNC Subtype

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

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.

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.

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.
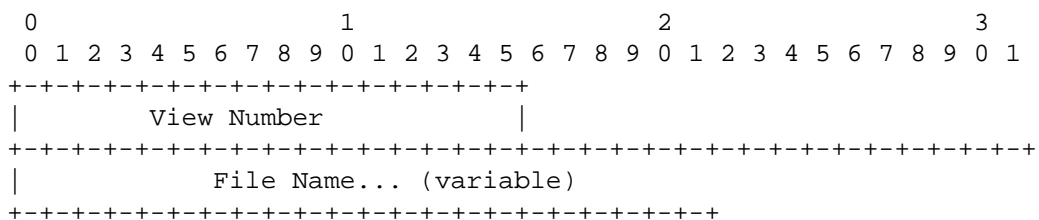
B.2.2. RIP Type

   The RIP Type is used to export RIP 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)
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

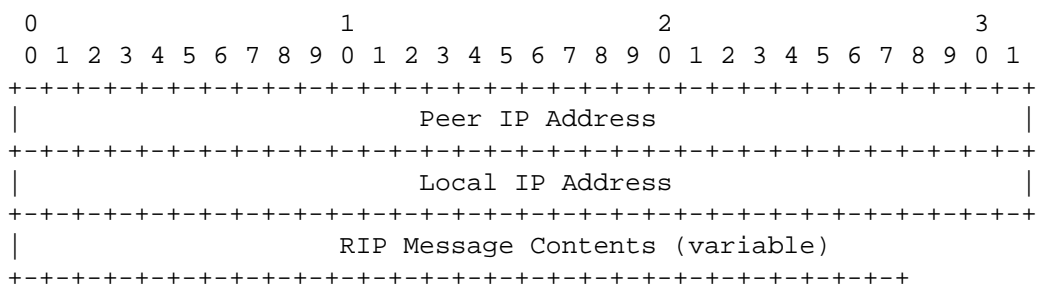   Figure 24: RIP Type

B.2.3. IDRP Type

   The IDRP Type was intended to be used to export Inter-Domain Routing
   Protocol (IDRP) 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.

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)            |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
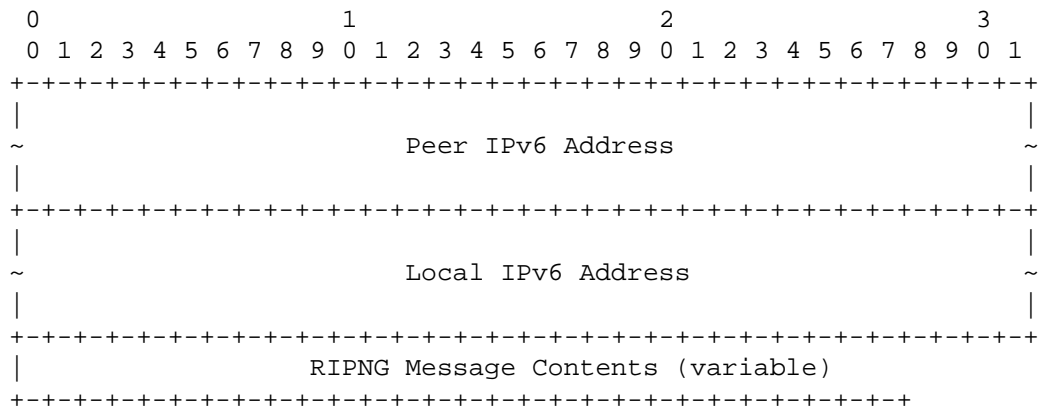
                       Figure 25: RIPNG Type

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 that became RFC 4760, "Multiprotocol
   Extensions to BGP-4". The BGP4PLUS_01 Type was specified to
   correspond to the -01 revision of that 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 were superseded by the
   BGP4MP Type.

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

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
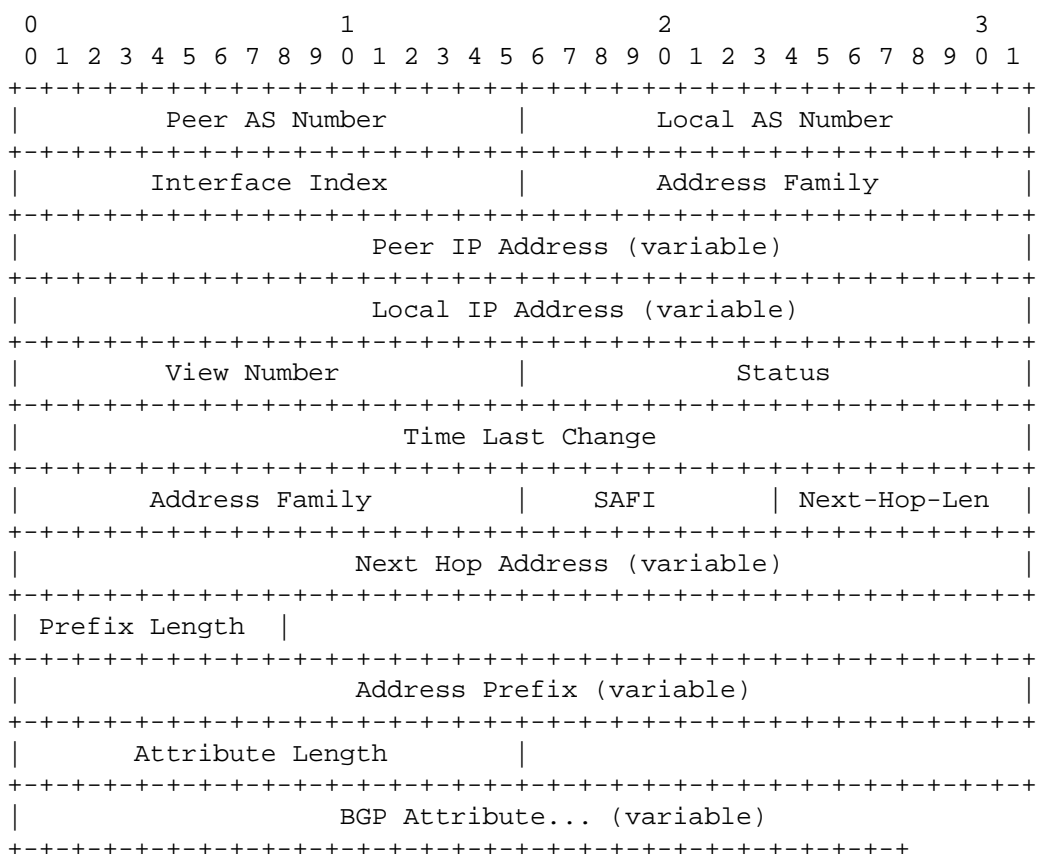   and has not been widely implemented.

```
        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 Number          |              Status           |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                       Time Last Change                        |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |         Address Family        |      SAFI     | Next-Hop-Len  |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                    Next Hop Address (variable)                |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       | Prefix Length  |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                    Address Prefix (variable)                  |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |       Attribute Length        |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                     BGP Attribute... (variable)               |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

                   Figure 26: BGP4MP_ENTRY Subtype

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
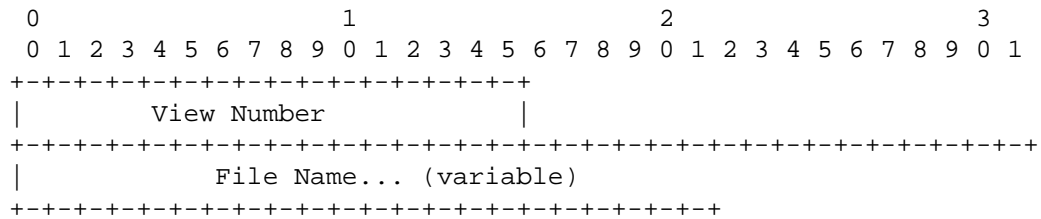   Subtype and is deprecated.

```
     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 Number           |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |             File Name... (variable)
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

                    Figure 27: BGP4MP_SNAPSHOT Subtype

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 Routing Toolkit (PyRT) developed by Richard Mortier while at
   Sprint Advanced Technology Labs.

Authors' Addresses

   Larry Blunk
   Merit Network

   EMail: ljb@merit.edu

   Manish Karir
   Merit Network

   EMail: mkarir@merit.edu

   Craig Labovitz
   Deepfield Networks

   EMail: labovit@deepfield.net