Network Working Group	L. Blunk
Internet-Draft	M. Karir
Intended status: Standards Track	Merit Network
Expires: May 19, 2008	C. Labovitz
	Arbor Networks
	November 16, 2007

MRT routing information export format
draft-ietf-grow-mrt-05.txt

Status of this Memo

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.

Table of Contents

1. Requirements notation

TOC

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

---

2. Introduction

TOC

Researchers and engineers often wish to analyze network behavior by studying routing protocol transactions and routing information base snapshots. To this end, the MRT format was developed to encapsulate, export, and archive this information in a standardized data representation. The BGP routing protocol, in particular, has been the subject of extensive study and analysis which has been significantly aided by the availability of the MRT format. The MRT format was initially defined in the MRT Programmer's Guide (Labovitz, C., “MRT Programmer's Guide,” November 1999.) [MRT PROG GUIDE].
This memo serves to document the MRT format as currently implemented in publicly available software. The format has been extended since it's original introduction in the MRT toolset and these extensions are also included in this memo. Further extensions may be introduced at a later date through additional definitions of the MRT Type field and Subtype fields.

---

3. Basic MRT Format

TOC

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

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

Header Field Descriptions:

Timestamp:

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

Type:

A 2-octet field that indicates the Type of information contained in the message field. Types 1 through 5 are used for MRT control information while Types 6 and higher are used for routing information.

Subtype:

A 2-octet message Subtype field

Length:

A 4-octet message length field. The length does not include the header.

Message:

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

TOC

4. MRT Control Types

The MRT format defines five Control Type messages. These messages are OPTIONAL and MAY be used to communicate the state of the MRT message source and it's peering sessions. The message field MAY contain an OPTIONAL message string for diagnostic purposes. The message string encoding MUST follow the UTF-8 transformation format. The Subtype field is unused for these Types and SHOULD be set to 0.
The MRT Control Types are defined below:

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

4.1. NULL Type

TOC

The NULL Type message causes no operation, A sender may wish to send these for synchronization or keep-alive purposes.

4.2. START Type

TOC

The START Type indicates a sender is about to begin sending MRT messages

---

4.3. DIE Type

TOC

A DIE Type signals that the receiver should shut down.

4.4. I_AM_DEAD Type

TOC

A I_AM_DEAD indicates that the sender is shutting down.

TOC

4.5. PEER_DOWN Type

A PEER_DOWN indicates when one of the sender's peers is down. In practice, a sender will likely have multiple peers. The sender SHOULD use the Message field to convey the IP address of the peer represented in UTF-8.

---

5. MRT Routing Information Types

TOC

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

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

5.1. BGP Type

TOC

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

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

5.1.1. BGP_NULL Subtype

TOC

The BGP_NULL Subtype is a reserved Subtype.

5.1.2. BGP_UPDATE Subtype

TOC

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

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        Source AS number       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Source IP address                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Destination AS number      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     Destination 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.

5.1.3. BGP_PREF_UPDATE Subtype

TOC

The BGP_PREF_UPDATE Subtype is not defined.

5.1.4. BGP_STATE_CHANGE Subtype

TOC

The BGP_STATE_CHANGE Subtype is used to record changes in the BGP finite state machine. These FSM states and their numeric encodings are defined in RFC 4271 (Rekhter, Y., Li, T., and S. Hares, “A Border Gateway Protocol 4 (BGP-4),” January 2006.) [RFC4271], Appendix 1. Both the old state value and the new state value are encoded as 2-octet numbers. The format of the MRT Message field is as follows:

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

5.1.5. BGP_SYNC Subtype

TOC

The BGP_SYNC Subtype is used to indicate a File Name where BGP Table Dump messages should be recorded. The View # corresponds to the View # provided in the TABLE_DUMP Type messages. The following format applies to this Subtype:

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

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

5.1.6. BGP_OPEN Subtype

TOC

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

5.1.7. BGP_NOTIFY Subtype

TOC

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

5.1.8. BGP_KEEPALIVE Subtype

TOC

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

5.2. RIP Type

TOC

The RIP Type is used to export RIP protocol packets as defined in RFC 1058 (Hedrick, C., "Routing Information Protocol," June 1988.) [RFC1058]. The Subtype field is currently reserved for this Type and SHOULD be set to 0.
The format of the MRT Message field for the RIP Type is as follows:

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

5.3. IDRP Type

TOC

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

5.4. RIPNG Type

TOC

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

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Source IP address                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Destination IP address                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                RIPNG Message Contents (variable)
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

5.5. BGP4PLUS and BGP4PLUS_01 Types

TOC

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

5.6. OSPF Type

TOC

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

    0   OSPF_STATE_CHANGE
    1   OSPF_LSA_UPDATE

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

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

5.7. TABLE_DUMP Type

TOC

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

    1   AFI_IPv4
    2   AFI_IPv6

The format of the TABLE_DUMP Type is illustrated below.

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

The View field is normally 0 and is intended for cases where an implementation may have multiple RIB views (such as a route server). The Sequence field is a simple incremental counter for each RIB entry. A typical RIB dump will exceed the 16-bit bounds of this counter and implementation should simply wrap back to zero and continue incrementing the counter in such cases.
The Prefix field contains the IP address of a particular routing RIB entry. The size of this field is dependent on the value of the Subtype for this message. For AFI_IPv4, this field is 4 octets, for AFI_IPv6, it is 16 octets in length. The Prefix Length field indicates the length in bits of the prefix mask for the preceding Prefix field.
The Status octet is not used in the TABLE_DUMP Type and SHOULD be set to 1.
The Originated Time contains the 4-octet time at which this prefix was heard. The value represents the time in seconds since 1 January 1970 00:00:00 UTC.
The Peer IP field is the IP address of the peer which provided the update for this RIB entry. As with the Prefix field, the size of this field is dependent on the Subtype. AFI_IPv4 indicates a 4 octet field and an IPv4 address, while a Subtype of AFI_IPv6 requires a 16 octet field and an IPv6 address. The Peer AS field contains the AS number of the peer.
Attribute length is the length of Attribute field and is 2-octets. The Attribute field contains the attribute information for the RIB entry.

TOC

5.8. TABLE_DUMP_V2 Type

The TABLE_DUMP_V2 type updates the TABLE_DUMP type to include 32-bit ASN support and full support for BGP Multiprotocol extensions. It also improves upon the space efficiency of the TABLE_DUMP type by employing an index table for peers and permitting a single MRT record per NLRI entry. The following subtypes are used with the TABLE_DUMP_V2 type.

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

An initial PEER_INDEX_TABLE MRT record provides the BGP ID of the collector, an optional view name, and a list of indexed peers. Following the PEER_INDEX_TABLE MRT record, a series of MRT records are used to encode RIB table entries. The header of the PEER_INDEX_TABLE Subtype is shown below. The View Name is optional and, if not present, the View Name Length MUST be set to 0. The View Name encoding MUST follow the UTF-8 transformation format.

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

The format of the peer entries is shown below. The PEER_INDEX_TABLE record contains Peer Count peer entries.

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

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

The Peer Type field is a bit field which encodes the type of the AS and IP address as follows:

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

The records which follow the PEER_INDEX_TABLE record constitute the RIB entries and include a header which specifies a sequence number, NLRI, and a count of the number of RIB entries which follow.

The format for the RIB_IPV4_UNICAST, RIB_IPV4_MULTICAST, RIB_IPV6_UNICAST, and RIB_IPV6_MULTICAST headers are shown below. The Prefix Length and Prefix fields are encoded in the same manner as the BGP NLRI encoding for IPV4 and IPV6 prefixes. Namely, the Prefix field contains address prefixes followed by enough trailing bits to make the end of the field fall on an octet boundary. Note that the value of trailing bits is irrelevant.

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

The RIB_GENERIC header is shown below. It includes Address Family Identifier (AFI), Subsequent AFI and a single NLRI entry. The NLRI information is specific to the AFI and SAFI values. An implementation which does not recognize particular AFI and SAFI values SHOULD discard the remainder of the MRT record.

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

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

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

There is one exception to the encoding of BGP attributes for the BGP MP_REACH_NLRI attribute (BGP Type Code 14) [RFC 4760]. Since the AFI, SAFI, and NLRI information is already encoded in the MULTIPROTOCOL header, only the Next Hop Address Length and Next Hop Address fields are included. The Reserved field is omitted. The attribute length is also adjusted to reflect only the length of the Next Hop Address Length and Next Hop Address fields.

5.9. BGP4MP Type

TOC

This Type was initially defined in the Zebra software package for the BGP protocol with multiprotocol extension support as defined by RFC 4760 (Bates, T., Chandra, R., Katz, D., and Y. Rekhter, “Multiprotocol Extensions for BGP-4,” January 2007.) [RFC4760]. It supersedes the BGP,

BGP4PLUS, BGP4PLUS_01 Types. The BGP4MP Type has four Subtypes which are defined as follows:

```
0    BGP4MP_STATE_CHANGE
1    BGP4MP_MESSAGE
2    BGP4MP_ENTRY                  *DEPRECATED*
3    BGP4MP_SNAPSHOT
4    BGP4MP_STATE_CHANGE_AS4
5    BGP4MP_MESSAGE_AS4
```

5.9.1. BGP4MP_STATE_CHANGE Subtype

TOC

This record is used to encode state changes in the BGP finite state machine. As with the BGP_STATE_CHANGE Subtype, the BGP FSM states are encoded in the Old State and New State fields to indicate the previous and current state. The format is illustrated below:

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

While BGP4MP_STATE_CHANGE message is similar to the BGP_STATE_CHANGE message, it also includes interface index and Address Family fields. As with the BGP_STATE_CHANGE message, the FSM states and their numeric encodings are defined in RFC 4271 (Rekhter, Y., Li, T., and S. Hares, “A Border Gateway Protocol 4 (BGP-4),” January 2006.) [RFC4271], Appendix 1. The interface index provides the interface number of the peering session. The index value is OPTIONAL and MAY be zero if unknown or unsupported. The Address Family indicates what types of addresses are in the the address fields. At present, the following AFI Types are supported:

```
1    AFI_IPv4
2    AFI_IPv6
```

5.9.2. BGP4MP_MESSAGE Subtype

TOC

This Subtype is used to encode BGP Messages. It is similar to the BGP_UPDATE Subtype, except that is can be used to encode any Type of message (not just BGP UPDATES). In order to determine the BGP message Type, the entire BGP message, including the BGP header, is included in the BGP Message field. 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Source AS number      |     Destination AS number     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Interface Index       |        Address Family         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   Source IP address (variable)                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                 Destination 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
```

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

5.9.3. BGP4MP_ENTRY Subtype

TOC

This Subtype is similar to the TABLE_DUMP Type and is used to record RIB table entries. It extends the TABLE_DUMP Type to include true multiprotocol support. However, this type does not support 32-bit AS numbers and has not been widely implemented. This type is deprecated in favor of the TABLE_DUMP_V2 which includes 32-bit AS number support and a more compact format.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        Source AS number       |     Destination AS number     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        Interface Index        |        Address Family         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   Source IP address (variable)                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                Destination 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)
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

5.9.4. BGP4MP_SNAPSHOT Subtype

TOC

This Subtype is used to indicate a filename containing BGP4MP_ENTRY records. It is similar to the BGP_SYNC message Subtype and shares the same fields.

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

5.9.5. BGP4MP_STATE_CHANGE_AS4 Subtype

TOC

This Subtype updates the BGP4MP_STATE_CHANGE Subtype to support 32BIT Autonomous System numbers. As with the BGP4MP_STATE_CHANGE Subtype, the BGP FSM states are encoded in the Old State and New State fields to indicate the previous and current state. Aside from the extension of the source and destination AS fields to 32 bits, 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Source AS number                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Destination AS number                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        Interface Index        |        Address Family         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Source IP address (variable)             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Destination IP address (variable)          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            Old State          |          New State            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

5.9.6. BGP4MP_MESSAGE_AS4 Subtype

TOC

This Subtype updates the BGP4MP_MESSAGE Subtype to support 32BIT Autonomous System 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Source AS number                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     Destination AS number                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        Interface Index        |        Address Family         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   Source IP address (variable)                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                 Destination IP address (variable)             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    BGP Message... (variable)
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

5.10. BGP4MP_ET

TOC

This Type was initially defined in the Sprint Labs Python Routing Toolkit (PyRT). It extends the MRT common header field to include a 32-bit microsecond timestamp field. The type and subtype field definitions remain as defined for the BGP4MP Type. The 32-bit microsecond timestamp immediately follows the length field in the MRT common header and precedes all other fields in the message. The 32-bit microsecond field is included in the computation of the length field value. The MRT common header modification is illustrated below.

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

TOC

5.11. ISIS Type

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

5.12. ISIS_ET Type

TOC

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

5.13. OSPFv3 Type

TOC

The OSPFv3 Type extends the original OSPF Type to support IPv6 addresses for the OSPFv3 protocol as defined in RFC 2740 (Coltun, R., Ferguson, D., and J. Moy, "OSPF for IPv6," December 1999.) [RFC2740]. The format of the MRT Message field for the OSPFv3 Type is as follows:

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

5.14. OSPFv3_ET Type

TOC

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

---

6. IANA Considerations

TOC

This section provides guidance to the Internet Assigned Numbers Authority (IANA) regarding registration of values related to the MRT specification, in accordance with BCP 26, RFC 2434 (Narten, T. and H. Alvestrand, “Guidelines for Writing an IANA Considerations Section in RFCs,” October 1998.) [RFC2434].

There are two name spaces in MRT that require registration: Type Codes and Subtype Codes.

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

The following policies are used here with the meanings defined in BCP 26: "Specification Required", "IETF Consensus".

6.1. Type Codes

TOC

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

6.2. Subtype Codes

TOC

Subtype Codes have a range from 0 to 65535. Subtype definitions are specific to a particular Type Code definition. New Subtype Code definition must reference an existing Type Code to which the Subtype belongs. As Subtype Codes are specific to Type Codes, new numbers must be unique for the particular Type Code to which the Subtype applies. Subtype Codes specific to the Type Codes 0 - 32767 are assigned by IETF Consensus. Subtype Codes specific to Type Codes 32768 - 65535 are assigned based on Specification Required.

TOC

7. Security Considerations

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

8. References

TOC

8.1. Normative References

TOC

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

8.2. Informative References

TOC

[MRT PROG GUIDE]	Labovitz, C., "MRT Programmer's Guide," November 1999 (HTML).

Authors' Addresses

TOC

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

Full Copyright Statement

TOC

Intellectual Property

that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.