Network Working Group
Internet-Draft
Expires: December 28, 2006

L. Blunk
M. Karir
Merit Network
C. Labovitz
Arbor Networks
June 26, 2006

MRT routing information export format
draft-ietf-grow-mrt-03.txt

Status of this Memo

Copyright Notice

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 MRT format was initially defined in the MRT Programmer's Guide [9]. The format can be used to export routing protocol messages, state changes, and routing information base contents.

Table of Contents

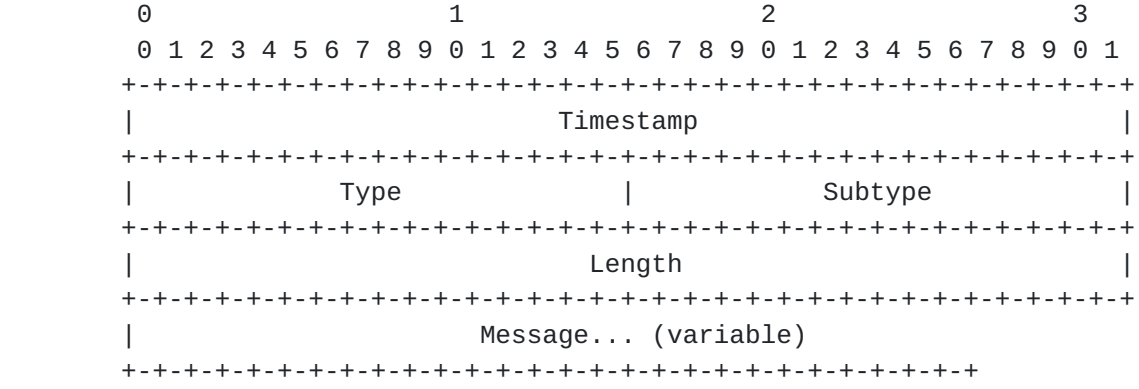
1. Introduction

Researchers and engineers often wish to analyze network behavior by studying routing protocol transactions and routing information base snapshots. To this end, the MRT 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.

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.

2. Basic MRT Format

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 basic MRT format is illustrated below.



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.

3. MRT Control Types

The MRT format defines five Control Type messages. These messages are using to relay the current state of MRT message source. The message field may contain an optional ASCII text string for diagnostic purposes. These control messages are unidirectional in nature and there is no form of an acknowledgment or response from the receiver to the sender. 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

3.1. NULL Type

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

3.2. START Type

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

3.3. DIE Type

A DIE Type signals that the receiver should shut down.

3.4. I_AM_DEAD Type

A I_AM_DEAD indicates that the sender is shutting down.

3.5. PEER_DOWN Type

A PEER_DOWN is sent when the sender's peer is down. In practice, a sender will likely have multiple peers. It is recommended that the sender use the Message field to convey the IP address of the peer represented in US-ASCII.

4. MRT Routing Information Types

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.

5	BGP
6	RIP
7	IDRP
8	RIPNG
9	BGP4PLUS
10	BGP4PLUS_01
11	OSPF
12	TABLE_DUMP
16	BGP4MP
17	BGP4MP_ET
32	ISIS
33	ISIS_ET
64	OSPF_ET

4.1. BGP Type

The BGP Type indicates the Message field contains BGP routing information. The BGP routing protocol is defined in RFC 1771 [1]. The information in the message is dependent on the Subtype value. The BGP Type is considered to be deprecated by the BGP4MP Type.

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

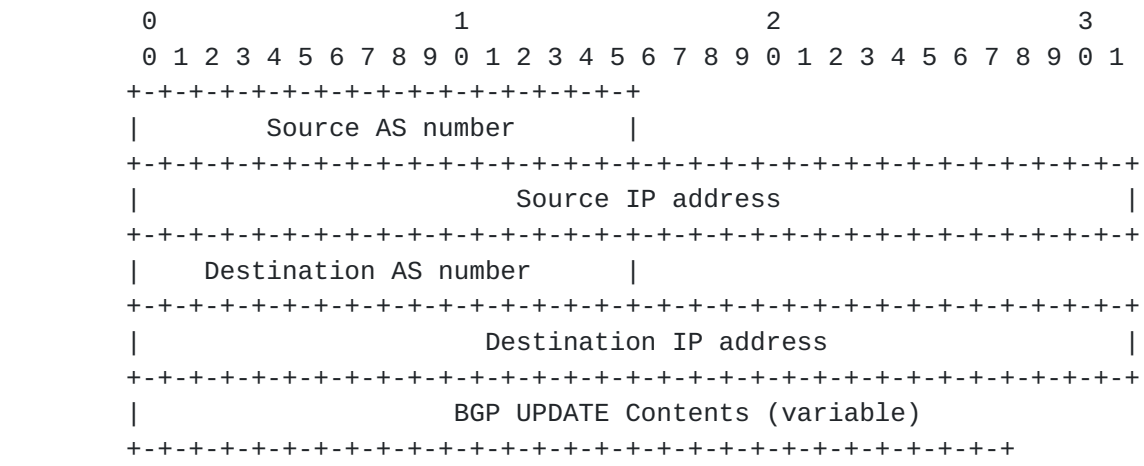
0	BGP_NULL
1	BGP_UPDATE
2	BGP_PREF_UPDATE
3	BGP_STATE_CHANGE
4	BGP_SYNC
5	BGP_OPEN
6	BGP_NOTIFY
7	BGP_KEEPALIVE

4.1.1. BGP_NULL Subtype

The BGP_NULL Subtype is a reserved Subtype.

4.1.2. 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:



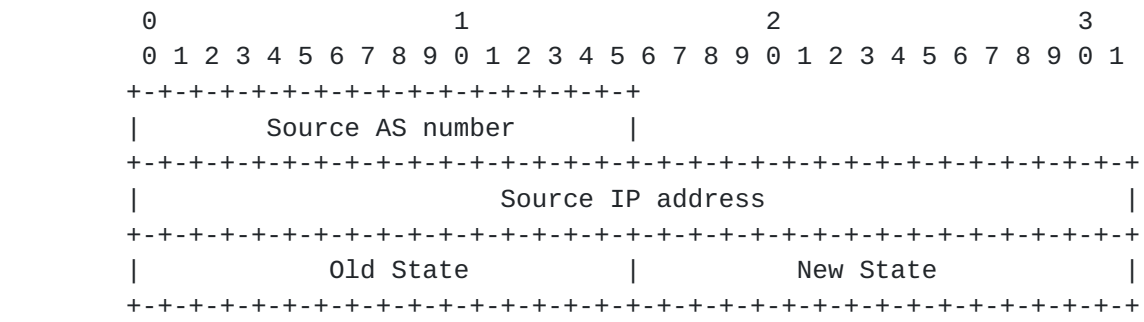
The BGP UPDATE contents include the entire BGP UPDATE message which follows the BGP Message Header. The BGP Message Header itself is not included.

4.1.3. BGP_PREF_UPDATE Subtype

The BGP_PREF_UPDATE Subtype is not defined.

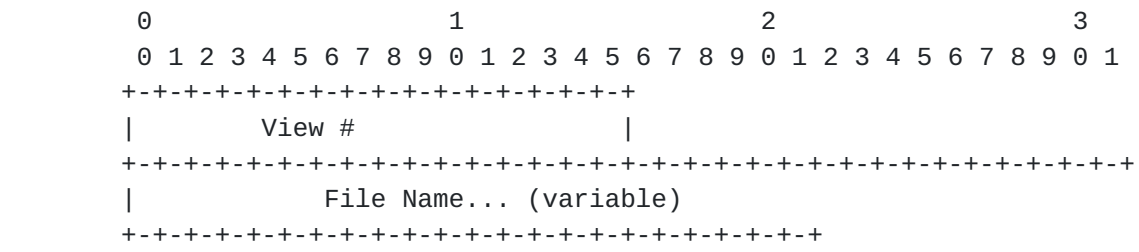
4.1.4. BGP_STATE_CHANGE Subtype

The BGP_STATE_CHANGE Subtype is used to record changes in the BGP finite state machine. These FSM states and their numeric encodings are defined in RFC 1771 [1], 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:



4.1.5. BGP_SYNC Subtype

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:



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

4.1.6. BGP_OPEN

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.

4.1.7. BGP_NOTIFY

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.

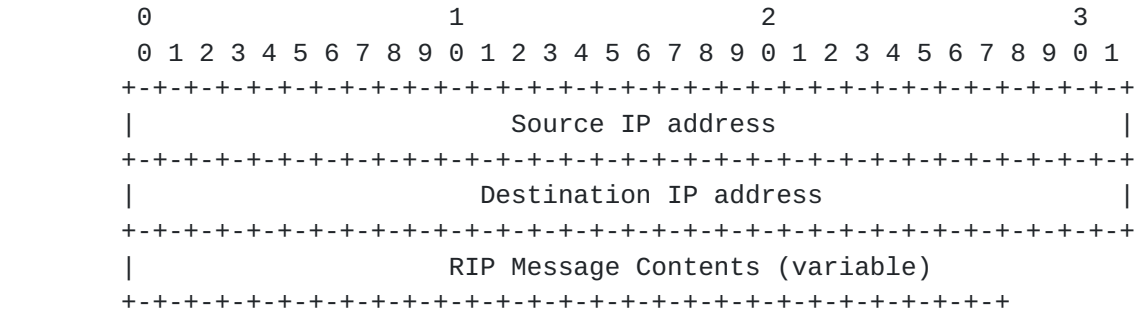
4.1.8. BGP_KEEPALIVE

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.

4.2. RIP Type

The RIP Type is used to export RIP protocol packets as defined in RFC 1058 [2]. 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:



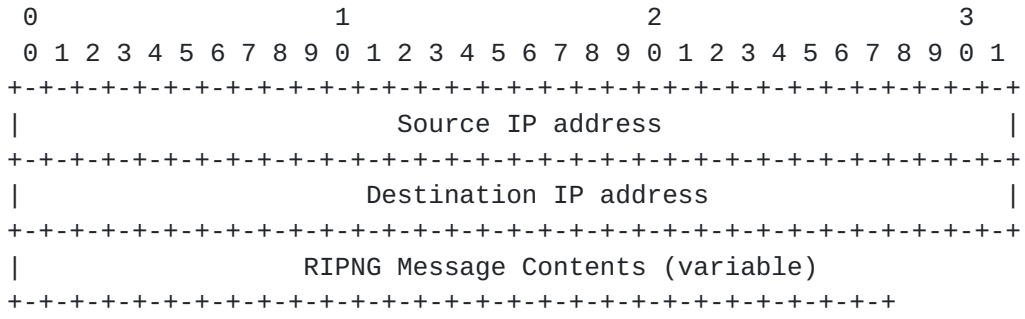
4.3. IDRP Type

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.

4.4. RIPNG Type

The RIPNG Type is used to export RIPNG protocol packets as defined in RFC 2080 [3]. 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:



4.5. BGP4PLUS and BGP4PLUS_01 Types

The BGP4PLUS and BGP4PLUS_01 Types were defined to support IPv6 BGP routing information. The BGP4PLUS Type was specified based on the initial Internet Draft for Multiprotocol Extensions to BGP-4. The BGP4PLUS_01 Type was specified to correspond to the -01 revision of this Internet Draft. The two Types share the same definitions in terms of their MRT format specifications.

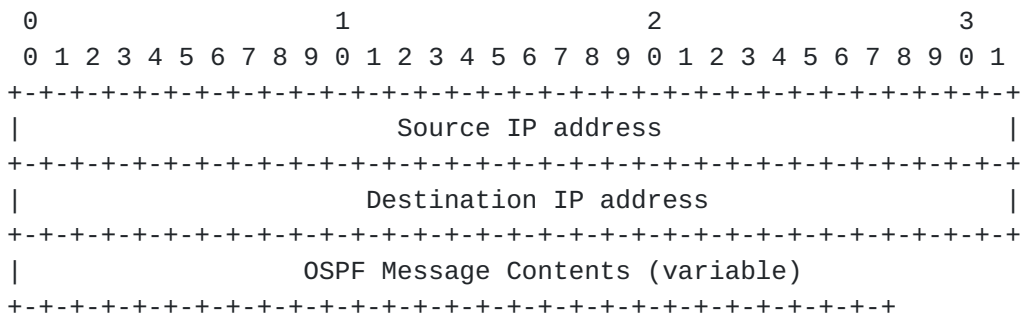
The Subtype field definitions are shared with the BGP Type, however, the address fields in the BGP_UPDATE, BGP_OPEN, BGP_NOTIFY, BGP_KEEPALIVE, and BGP_STATE_CHANGE Subtype messages are extended to 16 octets for IPv6 addresses. As with the BGP Type, the BGP4PLUS and BGP4PLUS_01 Types are deprecated as they superseded by the BGP4MP Type.

4.6. OSPF Type

This Type supports the OSPF Protocol as defined in RFC 2328 [4]. 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:

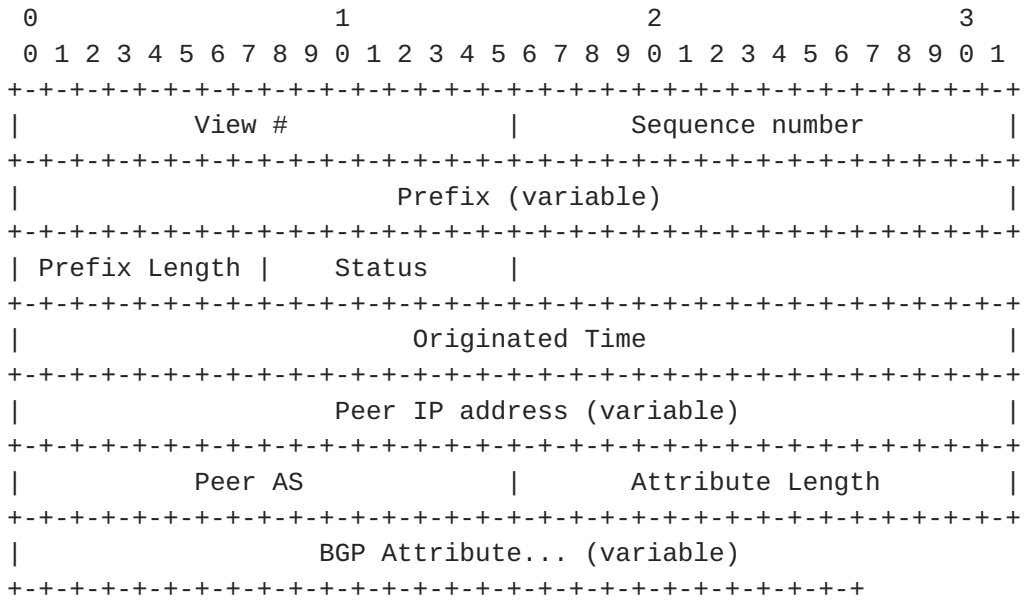


4.7. TABLE_DUMP Type

The TABLE_DUMP Type is used to encode routing table dumps. The Subtype is used to encode whether the table entry contains IPv4 or IPv6 addresses. There are currently 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.



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 a concatenated series of TABLE_DUMP Type messages.

The Prefix field contains the IP address of a particular routing

table dump 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 ID field is the IP address of the peer which provided the update for this routing table 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 route table entry.

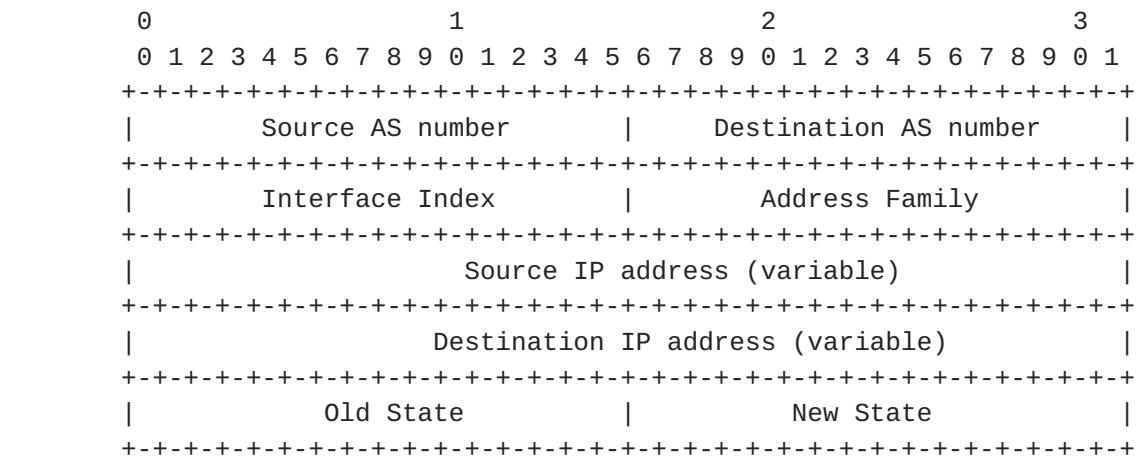
4.8. BGP4MP Type

This Type was initially defined in the Zebra software package for the BGP protocol with multiprotocol extension support as defined by RFC 2858 [5]. 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
3	BGP4MP_SNAPSHOT
4	BGP4MP_MESSAGE_32BIT_AS

4.8.1. BGP4MP_STATE_CHANGE Subtype

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:

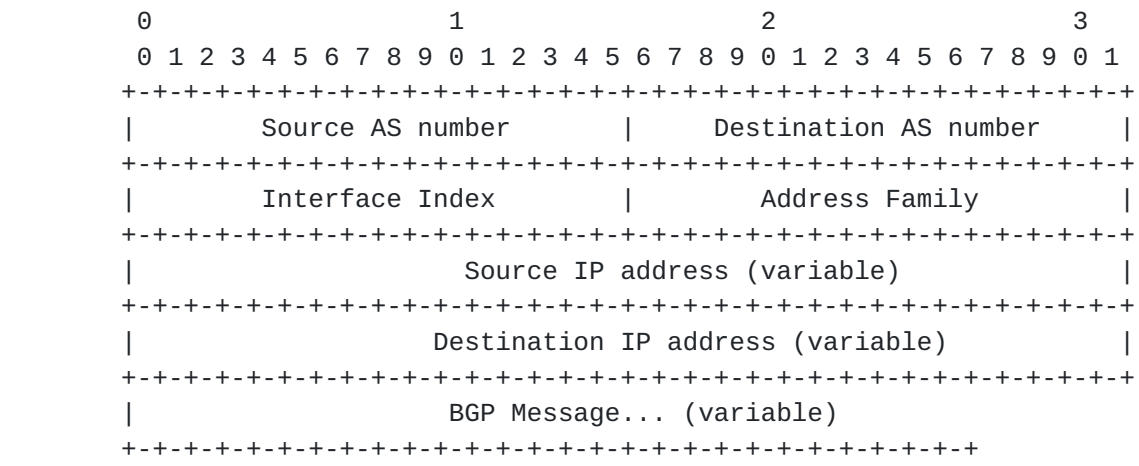


While BGP4MP_STATE_CHANGE message is similar to the BGP_STATE_CHANGE message, however, it also includes interface index and Address Family fields. As with the BGP_STATE_CHANGE message, the FSM states and their numeric encodings are defined in RFC 1771 [1], Appendix 1. Future updates to the BGP protocol specification will introduce a new state machine and thus render this message Type obsolete. The interface index provides the interface number of the peering session and the Address Family indicates what Types of addresses are in the the address fields. At present, only the following AFI Types are supported:

1 AFI_IPv4
2 AFI_IPv6

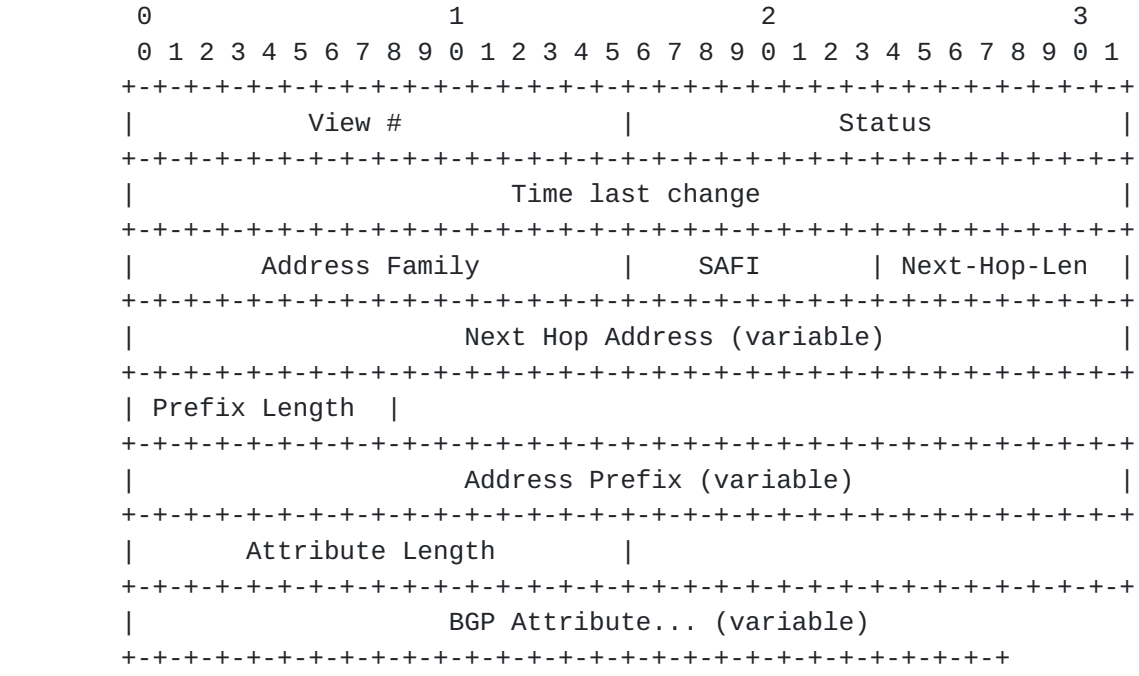
4.8.2. BGP4MP_MESSAGE Subtype

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:



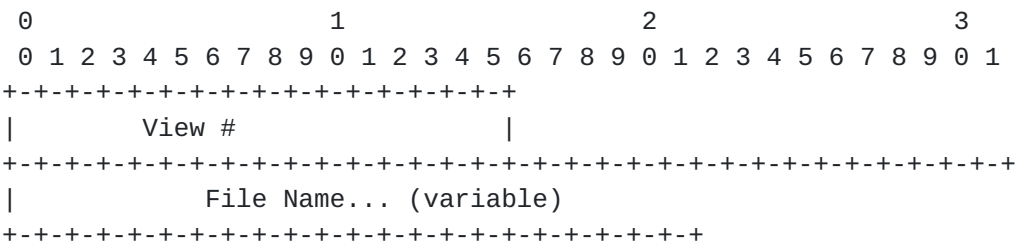
4.8.3. BGP4MP_ENTRY Subtype

This Subtype is used to record routing table entries. It is similar to the TABLE_DUMP Type. The primary difference being that the Address Family is encoded in the Message itself. Further, a Subsequence Address Family field (SAFI) is included as well.



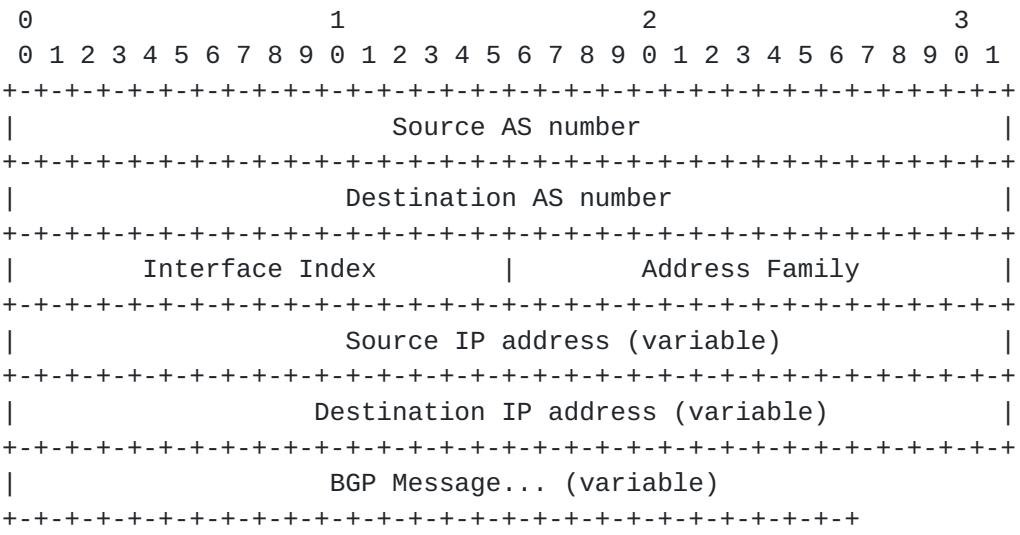
4.8.4. BGP4MP_SNAPSHOT Subtype

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.



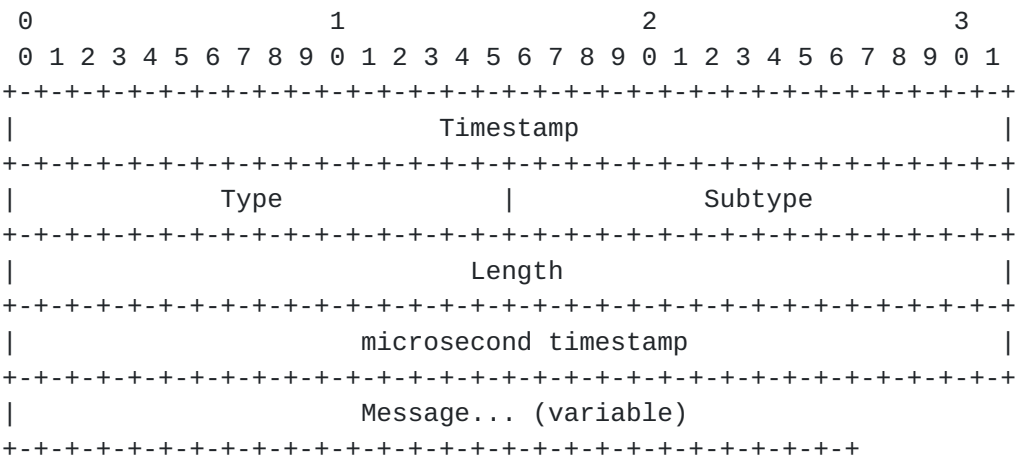
4.8.5. BGP4MP_MESSAGE_32BIT_AS Subtype

This Subtype updates the BGP4MP_MESSAGE Subtype to support 32BIT Autonomous System numbers. As the current 16 bit Autonomous System number space nears exhaustion, the introduction of 32 bit numbers will be required to support future Autonomous System number allocations. The BGP4MP_MESSAGE_32BIT_AS fields are shown below:



4.9. BGP4MP_ET

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



4.10. ISIS Type

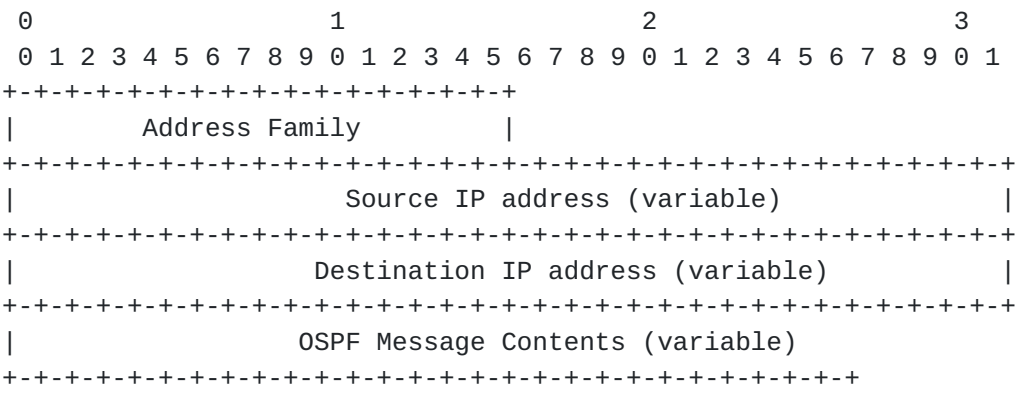
This Type was initially defined in the Sprint Labs Python Routing and supports the IS-IS routing protocol as defined in RFC 1195 [6]. There is no Type specific header for the ISIS Type. The Subtype code for this Type is undefined. The ISIS PDU directly follows the MRT common header fields.

4.11. ISIS_ET Type

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.

4.12. OSPF_ET Type

The OSPF_ET Type extends the the OSPF Type to support microsecond timestamps. As with the BGP4MP_ET and ISIS_ET Types, a 32-bit microsecond timestamp field is appended to the MRT common header after the length field. The OSPF_ET Type also extends the OSPF Type to support IPv6 addresses for the OSPFv3 protocol as defined in RFC 2740 [7]. The format of the MRT Message field for the OSPF_ET 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. 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 2434 [8].

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".

5.1. Type Codes

Type Codes have a range from 0 to 65535, of which 0-64 have been allocated. New Type Codes should 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.

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 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. Suptype Codes specific to Type Codes 32768 - 65535 are assigned based on Specification Required.

6. Security Considerations

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

7. References

7.1. Normative References

[1] Rekhter, Y. and T. Li, "A Border Gateway Protocol 4 (BGP-4)", RFC 1771, March 1995.

[2] Hedrick, C., "Routing Information Protocol", RFC 1058, June 1988.

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

[4] Moy, J., "OSPF Version 2", STD 54, RFC 2328, April 1998.

[5] Bates, T., Rekhter, Y., Chandra, R., and D. Katz, "Multiprotocol Extensions for BGP-4", RFC 2858, June 2000.

[6] Callon, R., "Use of OSI IS-IS for routing in TCP/IP and dual environments", RFC 1195, December 1990.

[7] Coltun, R., Ferguson, D., and J. Moy, "OSPF for IPv6", RFC 2740, December 1999.

[8] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 2434, October 1998.

7.2. Informative References

[9] "MRT Programmer's Guide", November 1999, <http://www.merit.edu/nrd/resources/mrtprogrammer.pdf>.

Authors' Addresses

Larry Blunk
Merit Network

Email: ljb@merit.edu

Manish Karir
Merit Network

Email: mkarir@merit.edu

Craig Labovitz
Arbor Networks

Email: labovit@arbor.net

Intellectual Property Statement

Disclaimer of Validity

Copyright Statement

Acknowledgment

Funding for the RFC Editor function is currently provided by the Internet Society.