IETF
Internet-Draft
Intended status: Informational
Expires: August 19, 2009

P. Cheng, Ed.
UCLA IRL
H. Yan, Ed.
K. Burnett, Ed.
D. Massey, Ed.
CSU-Netsec Group
L. Zhang, Ed.
UCLA IRL

Feb 15, 2009

BGP routing information in XML format
draft-cheng-grow-bgp-xml-00

Status of this Memo

Copyright Notice

Abstract

This document describes the XML format for BGP routing information (XFB). It can be used to describe both BGP messages and BGP control information. Compared with MRT, XFB is more extensible, human and machine-readable and can serve as a common interface for a variety of tools.

Requirements Language

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 RFC 2119 [RFC2119].

Table of Contents

1. Introduction

BGP routing information is an essential resource for both researchers and operation communities in Internet routing. In order to facilitate the collection of BGP data from multiple sources and the usage of the collected data by multiple parties, it is important to define a standard format to encapsulate, export, and archive it. A well designed format should have the following nice properties

- Human and machine-readable

- Easily accessible

- Suitable for further processing by existing tools

- Easy to add user annotations

- Easy to reconstuct raw BGP messages / ability to replay into router

- Record full control information

- Support BGP extensions

In this document, we describe XFB, a XML-based format for BGP routing information, which is designed to meet these requirements. XML (eXensible Markup Language) is a general-purpose markup language; its primary purpose is to facilitate the message exchange across different information systems, particularly via the Internet. Using XML as the base for our XFB markup provides the following advantages:

- XFB is human and machine-readable. By using CSS or XSL, XFB can be easily displayed on websites. Because XFB is based on XML which is a common interface to many applications, XFB can be processed by a variety of existing tools.

- XFB can easily be extended with additional information based on the raw BGP routing information. The BGP data is simply annotated with additional attributes and/or elements; programs which are not looking for this new information will simply ignore it. This allows us to easily modify XFB in general (or particular BGPmons) to allow for newly required information. We include guidelines for adding new standard elements in each section.

- XFB messages can be used to reconstruct the raw BGP messages, if needed.

Though XFB pays a storage cost since a compact binary message is unpacked into ASCII text together with additional tags, our experiments shows that by using the default compression parameters for bzip2, we can still store XFB data efficiently. For details, please refer to the section of storage size comparison (Appendix A).

In addition to XFB, we will briefly describe BGPmon (BGP Monitoring Service), an implementation of a service to collect BGP data and make it available in XFB format. Currently there are two types of BGP routing information encoded in XFB: BGP messages which come "over the wire" and may or may not be tagged with additional "helper" information, and BGP control/status information that originates from the BGPmon itself.

2. Terminology

2.1. XML Notations

The syntax of XFB, being an extension of XML, is very simple. There are only three terms throughout the XFB file:

o An "element" refers to a start tag, an end tag, and all the characters in between, e.g., "<example>text and/or nested elements</example>".

o An "empty element" combines the start tag and the end tag, e.g., "<empty/>".

o An "attribute" is part of an element. If present, they occur in the start tag, e.g., "<example name='value'>". Of course, they can also appear in empty elements, e.g., "<empty name='value'/>".

3. BGP Monitoring Service

BGP Monitoring System (BGPMon) is designed to monitor realtime BGP updates and routing tables from peering BGP routers. It supports distributed delployment to concurrently monitor many BGP routers. Most importantly, BGPmon can produce XFB stream for realtime processing and storage. The XFB format can accurately encode BGP data without losing any information and can be easiliy extended to represent new BGP features.

More specifically, BGPMon generates two types of messages: the BGP messages come "over the wire" and the status messages which periodically report the operational status of BGPMon itself. The status messages carry useful information such as the peering relationship between BGPMon and operational routers. Currently status messages convey three kind of information:

BGPMON
BGPMon information, currently only includes server up and down events.

SESSION
Peering information, including operation status such as uptime, the number of resets, the number of messages received, the number of announce/withdrawl received, etc.

CHAIN
Chaining information. Chaining is a unique feature by which BGPMon servers can be linked together to achieve high scalability and performance. Chaining information includes the operation status such as uptime, the number of messages received, etc.

QUEUE
Queue information, indicating the BGPMon internal queue usage for administrative purpose.

Please refer to BGPMon [1] for detail information

Moreover, please note that XFB is independent from BGPMon. Any other application is free to read and produce, or even extend XFB messages. We simpliy includes BGPMon here to demonstrate the ease of usage and flexibility of the XFB format.

4. Data Types

The purpose of XFB is to represent information relevent to BGP, therefore it only supports limited number of data types. XFB uses standard datatypes defined by W3C XML schema. Please refer to XML Schema [2] for detail datatype definition.

INTEGER
A integer is represented by the INTEGER data type. Integer data MUST be encoded in Base 10.

STRING
A single character is represented by the CHARACTER data type. A character string is represented by the STRING data type. Special characters must be encoded using entity references.

HEXBIN
A binary octet string is represented by the HEXBIN data type. Specifically, each octet is encoded by two hexadecimal digits.

ENUM
A enumerated type is represented by the ENUM data type, and consist of an ordered list of acceptable values. Each value has a representative keyword

DATETIME
A date-time string is represented by the DATETIME data type. Each date-time string identifies a particular instant in time; ranges are not supported

5. The XFB Data Model

In the following sections, the XFB format will be discussed in detail.

5.1. BGP_MESSAGE

BGP_MESSAGE is the top level message class. Every routing information is encoded in exactly one BGP_MESSAGE xml element.

```
+--------------------+
| BGP_MESSAGE        |
+--------------------+
| STRING xmlns       |
| STRING version     |<>----------[ TIME           ]
| INTEGER length     |<>--{0..1}--[ PEERING        ]
|                    |<>--{0..1}--[ ASCII_MSG      ]
|                    |<>--{0..1}--[ OCTET_MSG      ]
|                    |<>--{0..1}--[ STATUS_MSG     ]
+--------------------+
```

Figure 1: The root BGP_MESSAGE element

The BGP_MESSAGE class contains the following element classes:

TIME
One. The time associated with the BGP message

PEERING
Zero or one. The connection information of from which peer the message was received over.

ASCII_MSG
Zero or one. BGP message encoded in ascii xml format.

OCTET_MSG
Zero or one. BGP message encoded in hexadecimal format. A BGP_MESSAGE message SHOULD contain at least one of asciimsg and octests element, except the status messages.

STATUS_MSG
Zero or one. Periodic status message in ascii xml format.Applications which only care about BGP messages can safely ignore this element

The BGPData class has two attributes:

xmlns
Required. STRING. The namespace of XFB specification. Required for validation.

version
Required. STRING. The XFB specification version number.

length
Required. INTEGER. The total length of the message in characters, including the BGP_MESSAGE tag itself.

5.2. TIME

Every BGP_MESSAGE MUST contain a child time element, indicating the time when the BGP message has been received/generated. The time can be represented in either or both of timestamp and human readable formats; additional formats MAY be included as well.

```
+--------------------+
| TIME               |
+--------------------+
|                    |<>----------[ TIMESTAMP       ]
|                    |<>--{0..1}--[ DATETIME        ]
|                    |<>--{0..1}--[ PRECISION_TIME  ]
+--------------------+
```

Figure 2: TIME Class

The time class contains the following element classes:

TIMESTAMP

One. Unix timestamp (number of seconds since midnight UTC, January 1, 1970). The UNIX timestamp is recommended when it is expected that the data will be processed entirely by programs without human intervention.

DATETIME

Zero or one. Human readable time in DATETIME format (ex: 2008-12-30T01:26:42Z)

PRECISION_TIME

Zero or one. If it is desired for the time to be given more accurately, an additional precisiontime element MAY be used; if given, this element SHOULD contain the number of microseconds past the second that the message arrived.

5.3. PEERING

PEERING element uniquely identify the connection over which the data arrived.

```
+--------------------+
| PEERING            |
+--------------------+
|                    |<>----------[ SRC_ADDR        ]
|                    |<>----------[ SRC_PORT        ]
|                    |<>--{0..1}--[ SRC_AS          ]
|                    |<>----------[ DST_ADDR        ]
|                    |<>----------[ DST_PORT        ]
|                    |<>--{0..1}--[ DST_AS          ]
+--------------------+
```

The peering_session element contains the following four subelements.

SRC_ADDR

One. The source(local) address.

DST_ADDR

One. The destination(remote) address.

SRC_PORT

One. The source(local) port.

DST_PORT

One. The destination(remote) port

Other information MAY be inferred from the above elements; however, If it is desired to explicitly include the source and destination AS, the following elements MAY be used.

SRC_AS
   Zero or one. The source(local) AS number.

DST_AS
   Zero or one. The destination(local) AS number.

The address elements (SRC_ADDR/DST_ADDR) has a AFI attribute to indicate the corresponding address family. Currently, the expected values for the AFI attribute are "IPv4" and "IPv6", while additional families are allowed.

5.4. Representing the message

BGP routing information MAY be sent in either binary or ascii format. Binary messages contain BGP information exactly as it comes over the wire. They take up less space than ASCII messages, can be easily converted to play directly into routers. ASCII messages are human-readable and can be played into scripts. Messages MUST be sent in at least one of these two formats.

If possible, implmentations SHOULD prefer to send BGP messages in both formats; if only one is used, they SHOULD prefer binary

5.5. OCTET_MESSAGE (BGP Message in Octet Format)

The OCTET_MSG element simply embed a hexadecimal string.

```
+--------------------+
| OCTET_MSG          |
+--------------------+
|                    |<>----------[ MARKER        ]
|                    |<>----------[ LENGTH        ]
|                    |<>----------[ TYPE          ]
|                    |<>----------[ OCTETS        ]
+--------------------+
```

Figure 3: Octets Message Class

The OCTET_MSG element contains the following subelements.

MARKER
   Required. HEXBIN. The Marker (Mask) field in octets.

LENGTH

Required. INTEGER. The length of the message in octets.

TYPE

Required. ENUM. BGP message type. Message types 1-5 are defined in RFC 4271 [RFC4271]and RFC 2918 [RFC2918]; additional types MAY be handled as well. In the event of an unrecognized type, the type element MUST contain the value "UNKNOWN"..

1 - OPEN

2 - UPDATE

3 - NOTIFICATION

4 - KEEPALIVE

5 - ROUTE-REFRESH

6+ - UNKNOWN

OCTETS

Required. HEXBIN. BGP message in octets

5.6. ASCII_MSG (BGP Message in ASCII Format)

The ASCII_MSG represents the BGP information in a hierarchical tree structure.

```
+--------------------+
| ASCII_MSG          |
+--------------------+
|                    |<>--------[ MARKER              ]
|                    |<>--------[ LENGTH              ]
|                    |<>--------[ TYPE                ]
|                    |<>---+----[ OPEN                ]
|                    |     |----[ UPDATE              ]
|                    |     |----[ NOTIFICATION        ]
|                    |     |----[ KEEPALIVE           ]
|                    |     |----[ ROUTE-REFRESH       ]
|                    |     |----[ CISCO-ROUTE-REFRESH ]
|                    |     |----[ UNKNOWN             ]
+--------------------+
```

Figure 4: ASCII_MSG Class

The ASCII_MSG element contains the following child elements

MARKER
Required. HEXBIN. The Marker (Mask) field in octets.

LENGTH
Required. INTEGER. The length of the message in octets.

TYPE
Required. ENUM. BGP message type. Message types 1-5 are defined in RFC 4271 [RFC4271]and RFC 2918 [RFC2918]; additional types MAY be handled as well. In the event of an unrecognized type, the type element MUST contain the value "UNKNOWN"..

The ASCII_MSG element contains one of the following subelements, determined by the value of the type attribute.

OPEN
Zero or one. BGP open message.

UPDATE
Zero or one. BGP update message.

NOTIFICATION
Zero or one. BGP notification message.

KEEPALIVE
Zero or one. BGP keepalive message.

ROUTE_REFRESH
Zero or one. BGP route-refresh message.

CISCO_ROUTE_REFRESH
Zero or one. BGP cisco-route-refresh message.

UNKNOWN
Zero or one. Unrecognized BGP message. Binary data would be preserved in this element.

In the following sections, we would describe each message type individually.

5.6.1. OPEN

The OPEN element represent the BGP open messages.

```
      +--------------------+
      | OPEN               |
      +--------------------+
      |                    |<>----------[ VERSION         ]
      |                    |<>----------[ SRC_AS          ]
      |                    |<>----------[ HOLD_TIME       ]
      |                    |<>----------[ SRC_BGP         ]
      |                    |<>----------[ OPT_PAR_LEN     ]
      |                    |<>----------[ OPT_PAR         ]
      +--------------------+
```

Figure 5: OPEN class

The open element contains the following subelements.

VERSION
One. The protocol version number, in decimal.

SRC_AS
One. The Autonomous System number of the sender.

HOLD_TIME
One. the sender-proposed hold time, in second.

SRC_BGP
One. the BGP identifier of the sender.

OPT_PAR_LEN
One. The length of the optional parameters field in octets

OPT_PAR
One. The container of optional parameters.

5.6.1.1. Defining and Processing the Parameter Value field

OPT_PAR is a container class for zero or more parameter subelements.

```
      +--------------------+
      | OPT_PAR            |
      +--------------------+
      | INTEGER count      |<>--{0..*}--[ PARAMETER       ]
      +--------------------+
```

Figure 6: OPT_PAR Class

The opt_par element has the following attributes.

count
Required. INTEGER. The number of parameters.

The opt_par element contains the following subelements.

PARAMETER
Zero or more. The optional parameters.

For each optional parameter, the OPT_PAR element contains a PARAMETER subelement with attribute "code" containing the integer code for the parameter type. Each PARAMETER contains three subelements: a LENGTH tag giving the length of the data in octets, a TYPE tag giving the name of the parameter, and a tag whose label is the name of the parameter and which contains parameter-defined elements. If the code is not recognized, TYPE is set to OTHER and the OTHER element holds unprocessed hexadecimal data.

The last element s processed differently depending on the type of parameter. This section describes the most common parameter types. Additional parameter types MAY be defined, and MUST confirm to the following format. Every parameter has attribute "code" giving the integer value of its type, element LENGTH giving the length in octets of its data, and element TYPE giving its name. It then contains an element whose label is the name of the parameter. This element may have attributes and subelements defined as desired; however, attributes SHOULD describe the data, and the actual data SHOULD be in the form of subelements. Parameters do not need to contain information; self-closing elements are permitted. Thus, every parameter will be in the following format:

```
+--------------------+
| PARAMETER          |
+--------------------+
| INTEGER code       |<>--------[ LENGTH         ]
|                    |<>--------[ TYPE           ]
|                    |<>---+----[ AUTHENTICATION ]
|                    |     |----[ CAPABILITIES   ]
|                    |     |----[ OTHER          ]
+--------------------+
```

The parameter element has the following attributes.

code
Required. INTEGER. The integer code for the parameter type

The parameter element contains the following subelements.

LENGTH
One. The length of the data in octets

TYPE
One. The name of the parameter

AUTHENTICATION/CAPABILITES/OTHER
One. Two most common paraameter type: authentication or capabilities. Additional parameter types can be defined by the OTHER element.

5.6.1.1.1. AUTHENTICATION

An authentication parameter is stored with type "AUTHENTICATION" and additional attribute "code". This element contains hexadecimal authentication data.

```
+--------------------+
| AUTHENTICATION     |
+--------------------+
| INTEGER code       |
|                    |
+--------------------+
```

Figure 7: AUTHENTICATION class

For example, this parameter is expressed as

```
<PARAMETER code="1">
   <LENGTH>int</LENGTH>
   <TYPE>AUTHENTICATION</TYPE>
   <AUTHENTICATION code="int">hexadecimal data</AUTHENTICATION>
</PARAMETER>
```

5.6.1.1.2. CAPABILITIES

The capabilities parameter is encoded in CAPABILITIES element and one or more child CAP elements, as well as an attribute "count" giving the number of such elements. Each CAP element contains LENGTH, TYPE, and subelements decided by the type value.

```
    +---------------+
    | CAPABILITIES  |
    +---------------+           +---------+
    | INTEGER count |<>-{0..*}- |  CAP    |
    |               |           +---------+
    +---------------+           |         |<>----[ LENGTH    ]
                                |         |<>----[ TYPE      ]
                                |         |<>-+--[ MULTIPROTOCOL       ]
                                +---------+   |--[ ROUTE_REFRESH       ]
                                              |--[ CISCO_ROUTE_REFRESH ]
                                              |--[ OTHER ]
```

Figure 8: CAPABILITIES Class

For example, this parameter is expressed as

```
      <PARAMETER code="2">
         <LENGTH>int</LENGTH>
         <TYPE>CAPABILITIES</TYPE>
         <CAPABILITIES count="1">
            <CAP>
               <LENGTH>int</length>
               <TYPE>ROUTE_REFRESH</TYPE>
               <ROUTE_REFRESH/>
            </CAP>
         </CAPABILITIES>
      </PARAMETER>
```

5.6.2. The UPDATE element

The following subelements are included in the UPDATE message: <withdrawn_routes_len>, <withdrawn_routes>, <path_len>, <path_attrib>, and <NLRI>.

```
      +--------------------+
      | UPDATE             |
      +--------------------+
      |                    |<>----------[ WITHDRAWN_LEN         ]
      |                    |<>----------[ WITHDRAWN             ]
      |                    |<>----------[ PATH_ATTRIBUTES_LEN   ]
      |                    |<>----------[ PATH_ATTRIBUTES       ]
      |                    |<>----------[ NLRI                  ]
      +--------------------+
```

Figure 9

The UPDATE element contains the following subelements:

WITHDRAWN_LEN
One. the total length of the withdrawn routes field in octets

WITHDRAWN
One. contains zero or more PREFIX subelements, each of which contains an address prefix. This element has attribute "count" giving the total number of PREFIX subelements. PREFIX represents an address prefix. Each PREFIX also has an attirbute "label", which carries additional information associated with this prefix, such as new announcement, duplicated announcement, etc. These labels are uniquely produced by BGPMon, wich add meaningful information to prefixes.

PATH_ATTRIBUTES_LEN
One. the total length of the path attributes field in octets

PATH_ATTRIBUTES
One. contains zero or more ATTRIBUTE elements and a "count" attribute giving the number of such elements.

NLRI
One. contains zero or more PREFIX subelements, each of which contains an address prefix. This element has attribute "count" giving the total number of PREFIX subelements

5.6.3. PATH_ATTRIBUTES Class

PATH_ATTRIBUTES contains zero or more ATTRIBUTE.

```
    +-----------------+
    | PATH_ATTRIBUTES |
    +-----------------+            +---------------+
    | INTEGER counT   |<>--{0..*}--| ATTRIBUTE     |
    |                 |            +---------------+
    +-----------------+            | INTEGER code  |<>----[ FLAGS     ]
                                   |               |<>----[ LENGTH    ]
                                   |               |<>----[ TYPE      ]
                                   |               |<>-+--[ ORIGIN    ]
                                   +---------------+   |--[ AS_PATH   ]
                                                       |...
```

Moreover, the ATTRIBUTE element contains the following attributes:

count
Required. INTEGER. The number of attributes.

The ATTRIBUTE element contains the following subelements:

FLAGS

One. This element contains an attribute "code" giving the flag byte as an octet. Additionally, it contains subelements for all defined flags. The subelements for the current flags, as defined in RFC 1771, are OPTIONAL, TRANSITIVE, PARTIAL, and EXTENDED.

LENGTH

One. The attribute length

TYPE

One. The name of the attribute type

Each type correponds to a specific element. For brevitiy, the attribute types are summarized in the following table. Please refer to the schema and example sections for details.

Code	Type	Description
1	ORIGIN	This is a well-known mandatory attribute with value IGP, EGP, or INCOMPLETE.
2	AS_PATH	This is a well-known mandatory attribute with "type" attribute set to either as_set or as_sequence. It contains one or more AS subelements, each of which holds an Autonomous System number.
3	NEXT_HOP	This is a well-known mandatory attribute that holds the next hop address as its value.
4	MULTI_EXIT_DISC	This is an optional non-transitive attribute with integer value.
5	LOCAL_PREF	This is a well-known discretionary attribute with integer value.
6	ATOMIC_AGGREGATE	This is a well-known discretionary attribute with no value; the ATOMIC_AGGREGATE element is self-closing.
7	AGGREGATOR	This is an optional transitive attribute containing two subelements: AS, containing an AS number, and ADDR, containing an address.
8	COMMUNITITIES	This is an optional transitive attribute containing a 32-bit integer. [RFC1997]

9	ORIGINATOR_ID	This is an optional non-transitive attribute containing a 32-bit integer.
10	CLUSTER_LIST	This is an optional non-transitive attribute containing zero or more ID subelements and a "count" attribute giving the number of such subelements.
12	ADVERTISER	This is an optional non-transitive attribute containing an address.
13	RCID_PATH	This is an optional non-transitive attribute containing zero or more ID subelements and a "count" attribute giving the number of such subelements.
15	MP_REACH_NLRI	This is an optional non-transitive attribute that contains the following subelements: AFI contains the name of the Address Family Identifier being used. SAFI contains the Subsequent Address Family Identifier. NEXT_HOP contains the network address of the next hop. SNPA_LIST_LEN contains the length of the SNPA_LIST in octets. SNPA_LIST contains zero or more SNPA elements and a "count" attribute giving the total number of SNPA elements it contains. Furthermore, each SNPA represents a address. NLRI contains one or more PREFIX elements and a "count" attribute giving the number of such elements.[RFC4760]
15	MP_UNREACH_NLRI	This is an optional non-transitive attribute containing the following subelements: AFI contains the name of the Address Family Identifier being used. SAFI contains the Subsequent Address Family Identifier. WITHDRAWN contains one or more PREFIX elements and a "count" attribute giving the number of such elements.
16	EXTENDED_COMMUNITIES	This is an optional non-transitive attribute containing the octets value.[Rosen]

.7	AS4_PATH	This is a well-known mandatory attribute with "type" attribute set to either as_set or as_sequence. It contains one or more AS subelements, each of which holds an Autonomous System number. [RFC4893]
18	AS4_AGGREGATOR	This is an optional transitive attribute containing two subelements: AS, containing an AS number, and ADDR, containing an AFI address.

Additional attributes MAY be defined; they MUST follow the structure described above. Suppose that data is stored in the <NAME> element or in subelements.In the event that the element contains one subelement that may appear a number of times, the main <NAME> tag should include a "count" attribute. The <NAME> element may have any desired attributes and elements, but attributes SHOULD describe the data while elements SHOULD contain the data. If the code value is not defined, the <type> subelement has the value "UNKNOWN".

5.6.4. The KEEPALIVE

The KEEPALIVE element is self-closing and contains no attributes or subelements.

```
+--------------------+
| keepalive          |
+--------------------+
|                    |
+--------------------+
```

Figure 10: Keepalive Class

5.6.5. The ROUTE_REFRESH

The ROUTE_REFRESH element is self-closing and contains no attributes or subelements.

```
+--------------------+
| ROUTE_REFRESH      |
+--------------------+
|                    |
+--------------------+
```

Figure 11: ROUTE_REFRESH Class

5.6.6. The CISCO_ROUTE_REFRESH element

The CISCO_ROUTE_REFRESH is self-closing and contains no attributes or subelements.

```
+---------------------+
| CISCO_ROUTE_REFRESH |
+---------------------+
|                     |
+---------------------+
```

Figure 12: CISCO_ROUTE_REFRESH

5.6.7. The NOTIFICATION

The notification element contains the information for bgp notification.

```
+------------------+
| NOTIFICATION     |
+------------------+
|                  |<>----------[ CODE      ]
|                  |<>----------[ SUBCODE   ]
|                  |<>----------[ DATA      ]
+------------------+
```

Figure 13: NOTIFICATION Class

The notification element contains the following subelements.

CODE
One. Human readable error string cooresponding to a numberic 'value' attribute.

SUBCODE
One. Human readable sub error string cooresponding to a numberic 'value' attribute

DATA
One. The remainder of the message, whose content depends on the error code and subcode.

The <error_code> element is set according to the value of the "code" attribute:

code	error_code
1	Message Header Errori
2	OPEN Message Error
3	UPDATE Message Error
4	Hold Timer Expired
5	Finite State Machine Error
6	Cease
7-255	Undefined code

The <error_subcode> element is set according to the value of the "subcode" attribute:

code	subcode	error_subcode
1	1	Connection Not Synchronized
1	2	Bad Message Length
1	3	Bad Message Type
2	1	Unsupported Version Number
2	2	Bad Peer AS
2	3	Bad BGP Identifier
2	4	Unsupported Optional Parameter
2	5	Authentication Failure
2	6	Unacceptable Hold Time
3	1	Malformed Attribute List
3	2	Unrecognized Well-known Attribute
3	3	Missing Well-known Attribute
3	4	Attribute Flags Error
3	5	Attribute Length Error
3	6	Invalid ORIGIN Attribute
3	7	AS Routing Loop
3	8	Invalid NEXT_HOP Attribute
3	9	Optional Attribute Error
3	10	Invalid Network Field
3	11	Malformed AS_PATH
6	1	Maximum Number of Prefixes Reached"
6	2	Administrative Shutdown"
6	3	Peer De-configured"
6	4	Administrative Reset"
6	5	Connection Rejected"
6	6	Other Configuration Change"
6	7	Connection Collision Resolution"
6	8	Out of Resources"

For all other codes, the <error_subcode> subelement has the value "Undefined error subcode". The BGPmon MAY handle additional error subcodes.

5.6.7.1. Example

The NOTIFICATION message might look like this:

```
<NOTIFICATION code="3" subcode="7">
   <error_code>Bad Message Type</error_code>
   <error_subcode>AS Routing Loop</error_code>
   <data>hexadecimal</data>
</NOTIFICATION>
```

5.7. STATUS_MSG

The STATUS_MSG contains one of four kinds of messages: QUEUE_STATUS, SESSION_STATUS, and CHAIN_STATUS, BGPMON_STATUS, representing the operation status of the BGPMon internal quque, BGP peering routers, peering BGPMon server, and BGPMon server itself respectively.

```
   +---------------+
   | STATUS_MSG    |
   +---------------+
   |               |<>-----[ BGPMON         ]
   |               |<>--+--[ SESSION_STATUS ]
   |               |    |--[ CHAIN_STATUS   ]
   |               |    |--[ QUEUE_STATUS   ]
   |               |    |--[ BGPMON_STATUS  ]
   +---------------+
```

Figure 14: STATUS_MSG class

5.7.1. BGPMON

Because of the BGPMon chaining feature, it is possilbe that a client could receive status messages originated from multiple BGPMon servers. This BGPMON element is used by clients to identify the origin of the status message.

```
 +---------------+
 | BGPMON        |
 +---------------+
 |               |<>---------[ ADDR       ]
 |               |<>---------[ PORT       ]
 |               |<>-{0..1}--[ AS         ]
 +---------------+
```

Figure 15: BGPMON

The BGPMON element contains the following elements

ADDR
   One. The BGPMon network address, usually identical to the address which accepts user connections.

PORT
   One. The BGPMon port

AS
   Zero or one. Optional AS number

5.7.2. QUEUE_STATUS / QUEUE

QUEUE_STATUS includes zero or multiple child QUEUE elements, which in turn carry the information about BGPMon internal message queues.

```
 +---------------+
 | QUEUE_STATUS  |
 +---------------+           +---------+
 | INTEGER count |<>-{0..*}- |  QUEUE  |
 |               |           +---------+
 +---------------+           |         |<>----[ NAME   ]
                             |         |<>----[ ITEM   ]
                             |         |<>----[ READER ]
                             |         |<>----[ WRITER ]
                             |         |<>----[ PACING ]
                             +---------+
```

Figure 16: QUEUE_STATUS

The QUEUE_STATUS contains the following attributes:

count
   Required. INTEGER. The number of QUEUE elements.

The QUEUE_STATUS contains the following elements

QUEUE
   Zero or more. The QEUEU element which is described as the following.

The QUEUE element contains the following elements

NAME
   One. Human readalbe name

ITEM
   One. The statistic information about the queue usage.

READER

One. Ths statistic information about the queue reader.

WRITER

One. The statistic information about the queue writer.

PACING

One. Configuration and statistic information about the queue pacing machanism. Pacing is used to maintain stable queu usage. Please refer BGPMon specification for detail information

```
    +-----------+
    | PACING    |
    +-----------+
    |           |<>----[ FLAG        ]
    |           |<>----[ COUNT       ]
    |           |<>----[ WRITE_LIMIT ]
    +-----------+
```

Figure 17: QUEUE_STATUS

5.7.3. CHAIN_STATUS / CHAIN

CHAIN_STATUS includes zero or multiple child CHAIN elements.

```
    +---------------+
    | CHAIN_STATUS  |
    +---------------+           +-------+
    | INTEGER count |<>-{0..*}- | CHAIN |
    |               |           +-------+
    +---------------+           |       |<>--------[ ADDR         ]
                                |       |<>--------[ PORT         ]
                                |       |<>-{0..1}-[ AS           ]
                                |       |<>-{0..1}-[ STATE        ]
                                |       |<>-{0..1}-[ STATE_CHANGE ]
                                |       |<>-{0..1}-[ OPTIME       ]
                                |       |<>-{0..1}-[ RECV_MESSAGE ]
                                |       |<>-{0..1}-[ RESET        ]
                                +-------+
```

Figure 18: CHAIN_STATUS

The CHAIN_STATUS contains the following attributes:

count

Required. INTEGER. The number of CHAIN elements.

The CHAIN_STATUS contains the following elements

CHAIN
Zero or more. The CHAIN element which is described as the following.

The QUEUE element contains the following elements

ADDR
One. The address of the peering BGPMon server

ITEM
One. The port of the peering BGPMon server

AS
Zero or one. The AS number of the peering BGPMon server

STATE
Zero or one. The current state

STATE_CHANGE
Zero or one. The state change

OPTIME
Zero or one. The operation time, such as uptime, last down, etc.

RECV_MESSAGE
Zero or one. The statstic information about the number of messages received

RESET
One. The statistic information about the number of chain resets.

5.7.4. SESSION_STATUS / SESSION

SESSION_STATUS includes zero or multiple child SESSION elements.

```
    +----------------+
    | SESSION_STATUS |
    +----------------+          +-------+
    | INTEGER count  |<>-{0..*}-|SESSION|
    |                |          +-------+
    +----------------+          |       |<>--------[ ADDR         ]
                                |       |<>--------[ PORT         ]
                                |       |<>-{0..1}-[ AS           ]
                                |       |<>-{0..1}-[ STATE        ]
                                |       |<>-{0..1}-[ STATE_CHANGE ]
                                |       |<>-{0..1}-[ OPTIME       ]
                                |       |<>-{0..1}-[ RECV_MESSAGE ]
                                |       |<>-{0..1}-[ RESET            ]
                                |       |<>-{0..1}-[ PREFIX           ]
                                |       |<>-{0..1}-[ ATTRIBUTE        ]
                                |       |<>-{0..1}-[ MEMORY_USAGE     ]
                                |       |<>-{0..1}-[ ANNOUNCEMENT     ]
                                |       |<>-{0..1}-[ DUP_ANNOUNCEMENT ]
                                |       |<>-{0..1}-[ SAME_PATH        ]
                                |       |<>-{0..1}-[ DIFF_PATH        ]
                                |       |<>-{0..1}-[ WITHDRWAL        ]
                                |       |<>-{0..1}-[ DUP_WITHDRWAL    ]
                                +-------+
```

Figure 19: SESSION_STATUS

The CHAIN_STATUS contains the following attributes:

count
   Required. INTEGER. The number of CHAIN elements.

The CHAIN_STATUS contains the following elements

SESSION
   Zero or more. The SESSION element which is described as the following.

The QUEUE element contains the following elements. Please refer to CHAIN for identical elements.

PREFIX
   Zero or one. The statstic information about the number of prefixes received

ATTRIBUTE
   Zero or one. The statstic information about the number of path attributes received

MEMORY_USAGE
One. The statistic information about the memory usage

ANNOUNCEMENT
Zero or one. The statstic information about the number of announcements received

DUP_ANNOUNCEMENT
Zero or one. The statstic information about the number of duplicate announcements received

SAME_PATH
Zero or one. The statstic information about the number of same as path received

DIFF_PATH
Zero or one. The statstic information about the number of different as path received

WITHDRWAL
Zero or one. The statstic information about the number of withdrawl received

DUP_WITHDRWAL
Zero or one. The statstic information about the number of duplicate withdrawal received

6. Examples

6.1. BGP Open Message

```
<?xml version="1.0" encoding="utf-8"?>
<BGP_MESSAGE version="0.1" length="9999">
  <TIME>
    <TIMESTAMP>1228973922</TIMESTAMP>
    <DATETIME>2008-12-09T00:00:00Z</DATETIME>
    <PRECISION_TIME>328</PRECISION_TIME>
  </TIME>
  <PEERING_SESSION>
    <SRC_ADDR afi="IPv4">89.149.178.10</SRC_ADDR>
    <DST_ADDR afi="IPv4">129.82.138.6</DST_ADDR>
    <SRC_PORT>139</SRC_PORT>
    <DST_PORT>139</DST_PORT>
    <SRC_AS>3257</SRC_AS>
    <DST_AS>6447</DST_AS>
  </PEERING_SESSION>
  <ASCII_MSG length="87" type="OPEN">
```

```
       <MARKER>1122220</MARKER>
       <OPEN>
         <VERSION>4.0</VERSION>
         <SRC_AS>3593</SRC_AS>
         <HOLD_TIME>2943</HOLD_TIME>
         <SRC_BGP>89.149.22.11</SRC_BGP>
         <OPT_PAR_LEN>646</OPT_PAR_LEN>
         <OPT_PAR count="3">
           <PARAMETER code="1">
             <TYPE>AUTHENTICATION</TYPE>
             <LENGTH>234</LENGTH>
             <AUTHENTICATION code="8">FFDDEE00E039D2</AUTHENTICATION>
           </PARAMETER>
           <PARAMETER code="2">
             <TYPE>CAPABILITIES</TYPE>
             <LENGTH>2700</LENGTH>
             <CAPABILITIES count="2">
               <CAP>
                 <CODE>11</CODE>
                 <LENGTH>26</LENGTH>
                 <VALUE>CAP_STRING</VALUE>
               </CAP>
               <CAP>
                 <CODE>13</CODE>
                 <LENGTH>6</LENGTH>
                 <VALUE>CAP_STRING</VALUE>
               </CAP>
             </CAPABILITIES>
           </PARAMETER>
           <PARAMETER code="99">
             <TYPE>Unknown</TYPE>
             <LENGTH>1087</LENGTH>
             <UNKNOWN>123423A3D3BBEE</UNKNOWN>
           </PARAMETER>
         </OPT_PAR>
       </OPEN>
     </ASCII_MSG>
     <OCTETS length="3811" type="OPEN">FFFFFFFFFFFF...</OCTETS>
   </BGP_MESSAGE>
```

6.2. BGP Update Message

```
   <?xml version="1.0" encoding="utf-8"?>
   <BGP_MESSAGE version="0.1" length="9999">
     <TIME>
       <TIMESTAMP>1228973922</TIMESTAMP>
       <DATETIME>2008-12-09T00:00:00Z</DATETIME>
       <PRECISION_TIME>328</PRECISION_TIME>
     </TIME>
     <PEERING_SESSION>
       <SRC_ADDR afi="IPv4">89.149.178.10</SRC_ADDR>
       <DST_ADDR afi="IPv4">129.82.138.6</DST_ADDR>
       <SRC_PORT>139</SRC_PORT>
       <DST_PORT>139</DST_PORT>
       <SRC_AS>3257</SRC_AS>
       <DST_AS>6447</DST_AS>
     </PEERING_SESSION>
     <ASCII_MSG length="87" type="UPDATE">
       <MARKER>1110011122</MARKER>
       <UPDATE>
         <WITHDRAWN_ROUTES_LEN>5781</WITHDRAWN_ROUTES_LEN>
         <WITHDRAWN_ROUTES count="2">
           <PREFIX label="WITH">66.231.212</PREFIX>
           <PREFIX label="DUPW">66.186.174</PREFIX>
         </WITHDRAWN_ROUTES>
         <PATH_ATTRIBUTES_LEN>814</PATH_ATTRIBUTES_LEN>
         <PATH_ATTRIBUTE count="0">
           <ATTRIBUTE code="1">
             <Flags code="1111">
               <Optional/>
               <Transitive/>
               <Partial/>
               <Extended/>
             </Flags>
             <Length>343</Length>
             <TYPE>ORIGIN</TYPE>
             <ORIGIN>IGP</ORIGIN>
           </ATTRIBUTE>
           <ATTRIBUTE code="1">
             <Flags code="1111">
               <Optional/>
               <Transitive/>
               <Partial/>
               <Extended/>
             </Flags>
             <Length>343</Length>
             <TYPE>AS4_PATH</TYPE>
             <AS_PATH type="AS_SEQUENCE">
               <AS>1524</AS>
               <AS>5936</AS>
               <AS>178</AS>
             </AS_PATH>
           </ATTRIBUTE>
           <ATTRIBUTE code="1">
             <Flags code="1111">
               <Optional/>
```

```
              <Transitive/>
              <Partial/>
              <Extended/>
            </Flags>
            <Length>343</Length>
            <TYPE>NEXT_HOP</TYPE>
            <NEXT_HOP afi="IPv4">1.1.1.1</NEXT_HOP>
          </ATTRIBUTE>
          <ATTRIBUTE code="1">
            <Flags code="0010"><Partial/></Flags>
            <Length>343</Length>
            <TYPE>MULTI_EXIT_DISC</TYPE>
            <MULTI_EXIT_DISC>1905</MULTI_EXIT_DISC>
          </ATTRIBUTE>
          <ATTRIBUTE code="1">
            <Flags code="0010"><Partial/></Flags>
            <Length>343</Length>
            <TYPE>LOCAL_PREFERENCE</TYPE>
            <LOCAL_PREFERENCE>3678</LOCAL_PREFERENCE>
          </ATTRIBUTE>
          <ATTRIBUTE code="1">
            <Flags code="0010"><Partial/></Flags>
            <Length>343</Length>
            <TYPE>ATOMIC_AGGREGATE</TYPE>
            <ATOMIC_AGGREGATE/>
          </ATTRIBUTE>
          <ATTRIBUTE code="1">
            <Flags code="0010"><Partial/></Flags>
            <Length>343</Length>
            <TYPE>AGGREGATOR</TYPE>
            <AGGREGATOR>
              <AS>3547</AS>
              <ADDR afi="IPv4">1.2.2.2</ADDR>
            </AGGREGATOR>
          </ATTRIBUTE>
          <ATTRIBUTE code="1">
            <Flags code="0010"><Partial/></Flags>
            <Length>343</Length>
            <TYPE>COMMUNITY</TYPE>
            <COMMUNITY>6891</COMMUNITY>
          </ATTRIBUTE>
          <ATTRIBUTE code="1">
            <Flags code="0010"><Partial/></Flags>
            <Length>343</Length>
            <TYPE>ORIGINATOR_ID</TYPE>
            <ORIGINATOR_ID>977</ORIGINATOR_ID>
          </ATTRIBUTE>
          <ATTRIBUTE code="1">
```

```
             <Flags code="0010"><Partial/></Flags>
             <Length>343</Length>
             <TYPE>CLUSTER_LIST</TYPE>
             <CLUSTER_LIST count="3">
               <ID>CLUSTER_ID1</ID>
               <ID>CLUSTER_ID2</ID>
               <ID>CLUSTER_ID3</ID>
             </CLUSTER_LIST>
           </ATTRIBUTE>
           <ATTRIBUTE code="1">
             <Flags code="0010"><Partial/></Flags>
             <Length>343</Length>
             <TYPE>ADVERTISER</TYPE>
             <ADVERTISER afi="IPv4">10.1.1.1</ADVERTISER>
           </ATTRIBUTE>
           <ATTRIBUTE code="1">
             <Flags code="0010"><Partial/></Flags>
             <Length>343</Length>
             <TYPE>RCID_PATH</TYPE>
             <RCID_PATH count="2">
               <RCID>RCID1</RCID>
               <RCID>RCID2</RCID>
             </RCID_PATH>
           </ATTRIBUTE>
           <ATTRIBUTE code="1">
             <Flags code="0010"><Partial/></Flags>
             <Length>343</Length>
             <TYPE>MP_REACH_NLRI</TYPE>
             <MP_REACH_NLRI>
               <AFI>IPv6</AFI>
               <SUB_AFI>sub_afi</SUBS_AFI>
               <NEXT_HOP afi="IPv4">10.2.2.2</NEXT_HOP>
               <LENGTH>8890</LENGTH>
               <SNPA_LIST count="3">
                 <SNPA>1.1.1.1</SNPA>
                 <SNPA>1.1.2.1</SNPA>
                 <SNPA>1.1.3.1</SNPA>
               </SNPA_LIST>
               <NLRI count="3">
                 <PREFIX label="NANN">2.2.2.1</PREFIX>
                 <PREFIX label="NANN">2.2.3.1</PREFIX>
                 <PREFIX>2.2.4.1</PREFIX>
               </NLRI>
             </MP_REACH_NLRI>
           </ATTRIBUTE>
           <ATTRIBUTE code="1">
             <Flags code="0010"><Partial/></Flags>
             <Length>343</Length>
             <TYPE>MP_UNREACH_NLRI</TYPE>
             <MP_UNREACH_NLRI>
               <AFI>IPv6</AFI>
               <SNPA_LIST count="3">
                 <SNPA>1.1.1.1</SNPA>
                 <SNPA>1.1.2.1</SNPA>
                 <SNPA>1.1.3.1</SNPA>
               </SNPA_LIST>
             </MP_UNREACH_NLRI>
           </ATTRIBUTE>
           <ATTRIBUTE code="1">
             <Flags code="0010"><Partial/></Flags>
             <Length>343</Length>
             <TYPE>EXTENDED_COMMUNITIES</TYPE>
             <EXTENDED_COMMUNITIES>
               <FLAGS code="11">
                 <IANA/>
                 <TRANSITIVE/>
               </FLAGS>
               <DATA>ext_community_string</VALUE>
             </EXTENDED_COMMUNITIES>
           </ATTRIBUTE>
           <ATTRIBUTE code="1">
             <Flags code="0010"><Partial/></Flags>
             <Length>343</Length>
             <TYPE>AS4_PATH</TYPE>
             <AS4_PATH type="AS_SEQUENCE">
               <AS>5717</AS>
               <AS>3336</AS>
               <AS>2132</AS>
             </AS4_PATH>
           </ATTRIBUTE>
           <ATTRIBUTE code="1">
             <Flags code="0010"><Partial/></Flags>
             <Length>343</Length>
             <TYPE>AS4_AGGREGATOR</TYPE>
             <AS4_AGGREGATOR>
               <AS>5289</AS>
               <ADDR afi="IPv4">10.1.1.2</ADDR>
             </AS4_AGGREGATOR>
           </ATTRIBUTE>
         </PATH_ATTRIBUTE>
         <NLRI count="3">
           <PREFIX label="NANN">66.231.212.1</PREFIX>
           <PREFIX label="NANN">66.186.174.2</PREFIX>
           <PREFIX label="DANN">66.186.177.3</PREFIX>
         </NLRI>
       </UPDATE>
```

```
     </ASCII_MSG>
     <OCTETS length="3811" type="UPDATE">FFFFFFA..</OCTETS>
   </BGP_MESSAGE>
```

6.3. BGP Keepalive Message

```
 <BGP_MESSAGE xmlns="urn:ietf:params:xml:ns:xfb-0.1"
             length="000723" version="0.1">
   <TIME>
     <TIMESTAMP>1230600402</TIMESTAMP>
     <DATETIME>2008-12-30T01:26:42Z</DATETIME>
     <PRECISION_TIME>805</PRECISION_TIME>
   </TIME>
   <PEERING>
     <SRC_ADDR afi="IPv4">89.149.178.10</SRC_ADDR>
     <DST_ADDR afi="IPv4">129.82.138.6</DST_ADDR>
     <SRC_PORT>139</SRC_PORT>
     <DST_PORT>139</DST_PORT>
     <SRC_AS>3257</SRC_AS>
     <DST_AS>6447</DST_AS>
   </PEERING>
   <ASCII_MSG>
     <MARKER length="16">FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF</MARKER>
     <LENGTH>19</LENGTH>
     <TYPE value="4">KEEPALIVE</TYPE>
     <KEEPALIVE/>
   </ASCII_MSG>
   <OCTET_MSG>
     <MARKER length="16">FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF</MARKER>
     <LENGTH>19</LENGTH>
     <TYPE value="4">KEEPALIVE</TYPE>
     <OCTETS length="19">FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF001304</OCTETS>
   </OCTET_MSG>
 </BGP_MESSAGE>
```

6.4. BGP Notification Message

```
 <?xml version="1.0" encoding="utf-8"?>
   <BGP_MESSAGE version="0.1" length="9999">
     <TIME>
     <TIMESTAMP>1228973922</TIMESTAMP>
     <DATETIME>2008-12-09T00:00:00Z</DATETIME>
     <PRECISION_TIME>328</PRECISION_TIME>
   </TIME>
   <PEERING>
     <SRC_ADDR afi="IPv4">89.149.178.10</SRC_ADDR>
     <DST_ADDR afi="IPv4">129.82.138.6</DST_ADDR>
     <SRC_PORT>139</SRC_PORT>
     <DST_PORT>139</DST_PORT>
     <SRC_AS>3257</SRC_AS>
     <DST_AS>6447</DST_AS>
   </PEERING>
   <ASCII_MSG>
     <MARKER length="16">FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF</MARKER>
     <LENGTH>19</LENGTH>
     <TYPE value="5">NOTIFICATION</TYPE>
     <NOTIFICATION>
       <CODE value="3">Bad Message Type</ERROR_CODE>
       <SUBCODE value="7">AS Routing Loop</ERROR_SUBCODE>
       <DATA>hexadecimal</DATA>
     </NOTIFICATION>
   </ASCII_MSG>
   <OCTET_MSG>
     <MARKER length="16">FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF</MARKER>
     <LENGTH>19</LENGTH>
     <TYPE value="5">NOTIFICATION</TYPE>
     <OCTETS length="19">FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF001304</OCTETS>
   </OCTET_MSG>
 </BGP_MESSAGE>
```

6.5. Status Message

6.5.1. Queue Status Message

```
   <?xml version="1.0" encoding="utf-8"?>
   <BGP_MESSAGE xmlns="urn:ietf:params:xml:ns:xfb-0.1"
                length="001207" version="0.1">
     <TIME>
       <TIMESTAMP>1233804618</TIMESTAMP>
       <DATETIME>2009-02-05T03:30:18Z</DATETIME>
       <PRECISION_TIME>420</PRECISION_TIME>
     </TIME>
     <PEERING>
       <SRC_ADDR>ipv4any</SRC_ADDR>
       <SRC_PORT>4321</SRC_PORT>
       <SRC_AS>6447</SRC_AS>
       <DST_ADDR>65.49.129.101</DST_ADDR>
       <DST_PORT>179</DST_PORT>
       <DST_AS>3043</DST_AS>
     </PEERING>
     <STATUS_MSG>
       <QUEUE_STATUS count="3">
         <QUEUE>
           <NAME>PeerQueue</NAME>
           <ITEM max="19588" limit="30000">0</ITEM>
           <WRITER max="5" limit="0">5</WRITER>
           <READER max="1" limit="0">1</READER>
           <PACING>
             <COUNT max="0" limit="0">0</COUNT>
             <WRITE_LIMIT max="0" limit="0">192</WRITE_LIMIT>
           </PACING>
         </QUEUE>
         <QUEUE>
           <NAME>LabelQueue</NAME>
           <ITEM max="15637" limit="30000">11856</ITEM>
           <WRITER max="3" limit="0">3</WRITER>
           <READER max="1" limit="0">1</READER>
           <PACING>
             <COUNT max="0" limit="0">0</COUNT>
             <WRITE_LIMIT max="0" limit="0">197</WRITE_LIMIT>
           </PACING>
         </QUEUE>
         <QUEUE>
           <NAME>XMLQueue</NAME>
           <ITEM max="15476" limit="30000">13216</ITEM>
           <WRITER max="1" limit="0">1</WRITER>
           <READER max="1" limit="0">1</READER>
           <PACING>
             <COUNT max="0" limit="0">0</COUNT>
             <WRITE_LIMIT max="0" limit="0">678</WRITE_LIMIT>
           </PACING>
         </QUEUE>
       </QUEUE_STATUS>
     </STATUS_MSG>
   </BGP_MESSAGE>',
```

6.5.2. Session Status Message

```
 <BGP_MESSAGE xmlns="urn:ietf:params:xml:ns:xfb-0.1"
              length="003894" version="0.1">
   <TIME>
     <TIMESTAMP>1233804919</TIMESTAMP>
     <DATETIME>2009-02-05T03:35:19Z</DATETIME>
     <PRECISION_TIME>286</PRECISION_TIME>
   </TIME>
   <PEERING>
     <SRC_ADDR>ipv4any</SRC_ADDR>
     <SRC_PORT>4321</SRC_PORT>
     <SRC_AS>6447</SRC_AS>
     <DST_ADDR>65.49.129.101</DST_ADDR>
     <DST_PORT>179</DST_PORT>
     <DST_AS>3043</DST_AS>
   </PEERING>
   <STATUS_MSG>
     <SESSION_STATUS count="3">
       <SESSION>
         <ADDR>65.49.129.101</ADDR>
         <PORT>179</PORT>
         <AS>3043</AS>
         <STATE>6</STATE>
         <OPTIME last_down="0" last_action="1233804913">64516</OPTIME>
         <RECV_MESSAGE max="0" limit="0">10477308</RECV_MESSAGE>
         <RESET max="0" limit="0">0</RESET>
         <PREFIX max="0" limit="0">275527</PREFIX>
         <ATTRIBUTE max="0" limit="0">47661</ATTRIBUTE>
         <MEMORY_USAGE max="0" limit="0">13562636</MEMORY_USAGE>
         <ANNOUNCEMENT max="0" limit="0">287747</ANNOUNCEMENT>
         <DUP_ANNOUNCEMENT max="0" limit="0">29513778</DUP_ANNOUNCEMENT>
         <SAME_PATH max="0" limit="0">4820</SAME_PATH>
         <DIFF_PATH max="0" limit="0">92837</DIFF_PATH>
         <WITHDRAWAL max="0" limit="0">12220</WITHDRAWAL>
         <DUP_WITHDRAWAL max="0" limit="0">0</DUP_WITHDRAWAL>
       </SESSION>
       <SESSION>
         <ADDR>205.167.76.241</ADDR>
         <PORT>179</PORT>
         <AS>10876</AS>
         <STATE>6</STATE>
         <OPTIME last_down="0" last_action="1233804918">64515</OPTIME>
         <RECV_MESSAGE max="0" limit="0">10383216</RECV_MESSAGE>
         <RESET max="0" limit="0">0</RESET>
         <PREFIX max="0" limit="0">273496</PREFIX>
         <ATTRIBUTE max="0" limit="0">47395</ATTRIBUTE>
         <MEMORY_USAGE max="0" limit="0">12829036</MEMORY_USAGE>
         <ANNOUNCEMENT max="0" limit="0">287055</ANNOUNCEMENT>
         <DUP_ANNOUNCEMENT max="0" limit="0">29225124</DUP_ANNOUNCEMENT>
         <SAME_PATH max="0" limit="0">2239</SAME_PATH>
         <DIFF_PATH max="0" limit="0">59761</DIFF_PATH>
         <WITHDRAWAL max="0" limit="0">13559</WITHDRAWAL>
         <DUP_WITHDRAWAL max="0" limit="0">0</DUP_WITHDRAWAL>
       </SESSION>
       <SESSION>
         <ADDR>89.149.178.10</ADDR>
         <PORT>179</PORT>
         <AS>3257</AS>
         <STATE>6</STATE>
         <OPTIME last_down="0" last_action="1233804918">64515</OPTIME>
         <RECV_MESSAGE max="0" limit="0">245796</RECV_MESSAGE>
         <RESET max="0" limit="0">0</RESET>
         <PREFIX max="0" limit="0">271825</PREFIX>
         <ATTRIBUTE max="0" limit="0">47476</ATTRIBUTE>
         <MEMORY_USAGE max="0" limit="0">14745401</MEMORY_USAGE>
         <ANNOUNCEMENT max="0" limit="0">290449</ANNOUNCEMENT>
         <DUP_ANNOUNCEMENT max="0" limit="0">0</DUP_ANNOUNCEMENT>
         <SAME_PATH max="0" limit="0">175480</SAME_PATH>
         <DIFF_PATH max="0" limit="0">77044</DIFF_PATH>
         <WITHDRAWAL max="0" limit="0">18624</WITHDRAWAL>
         <DUP_WITHDRAWAL max="0" limit="0">0</DUP_WITHDRAWAL>
       </SESSION>
     </SESSION_STATUS>
   </STATUS_MSG>
 </BGP_MESSAGE>
```

7. The XFB Schema

```
 <?xml version="1.0" encoding="utf-8" ?>
 <xs:schema targetNamespace="urn:ietf:params:xml:ns:xfb-0.1"
            xmlns="urn:ietf:params:xml:ns:xfb-0.1"
            xmlns:xfb="urn:ietf:params:xml:ns:xfb-0.1"
            xmlns:xs="http://www.w3.org/2001/XMLSchema"
            elementFormDefault="qualified"
            attributeFormDefault="unqualified">
   <xs:annotation>
     <xs:documentation>
       XML Format for BGP Information v0.1, see RFC XXX
       </xs:documentation>
   </xs:annotation>
   <xs:element name="BGP_MESSAGE">
     <xs:complexType>
```

```
       <xs:sequence>
         <xs:element ref="xfb:TIME"/>
         <xs:element ref="xfb:PEERING" minOccurs="0" maxOccurs="1"/>
         <xs:element ref="xfb:ASCII_MSG" minOccurs="0" maxOccurs="1"/>
         <xs:element ref="xfb:OCTET_MSG" minOccurs="0" maxOccurs="1"/>
         <xs:element minOccurs="0" maxOccurs="1" ref="xfb:STATUS_MSG"/>
       </xs:sequence>
       <xs:attribute type="xs:string" name="version"
                     fixed="0.1" use="required"/>
       <xs:attribute use="required" name="length"
                     type="xs:nonNegativeInteger"/>
     </xs:complexType>
   </xs:element>
   <xs:element name="TIME">
     <xs:complexType>
       <xs:sequence>
         <xs:element type="xs:long" name="TIMESTAMP"/>
         <xs:element type="xs:dateTime" name="DATETIME"
                     minOccurs="0" maxOccurs="1"/>
         <xs:element name="PRECISION_TIME" type="xs:long"
                     minOccurs="0" maxOccurs="1"/>
       </xs:sequence>
     </xs:complexType>
   </xs:element>
   <xs:element name="PEERING">
     <xs:complexType>
       <xs:sequence>
         <xs:element name="SRC_ADDR" type="xfb:address"/>
         <xs:element name="SRC_PORT" type="xfb:port"/>
         <xs:element type="xfb:asn" name="SRC_AS"
                     minOccurs="0" maxOccurs="1"/>
         <xs:element type="xfb:address" name="DST_ADDR"/>
         <xs:element name="DST_PORT" type="xfb:port"/>
         <xs:element name="DST_AS" type="xfb:asn"
                     minOccurs="0" maxOccurs="1"/>
       </xs:sequence>
     </xs:complexType>
   </xs:element>
   <xs:element name="OCTET_MSG">
     <xs:complexType>
       <xs:sequence>
         <xs:element name="MARKER" type="xfb:octets"/>
         <xs:element name="LENGTH" type="xs:nonNegativeInteger"/>
         <xs:element name="TYPE" type="xfb:bgp_type"/>
         <xs:element name="OCTETS" type="xfb:octets"/>
       </xs:sequence>
     </xs:complexType>
   </xs:element>
   <xs:element name="ASCII_MSG">
     <xs:complexType>
       <xs:sequence>
         <xs:element name="MARKER" type="xfb:octets"/>
         <xs:element name="LENGTH" type="xs:nonNegativeInteger"/>
         <xs:element name="TYPE" type="xfb:bgp_type"/>
         <xs:choice>
           <xs:element ref="xfb:OPEN"/>
           <xs:element ref="xfb:UPDATE"/>
           <xs:element ref="xfb:NOTIFICATION"/>
           <xs:element ref="xfb:KEEPALIVE"/>
           <xs:element ref="xfb:ROUTE_REFRESH"/>
           <xs:element ref="xfb:CISCO_ROUTE_REFRESH"/>
           <xs:element ref="xfb:UNKNOWN"/>
         </xs:choice>
       </xs:sequence>
     </xs:complexType>
   </xs:element>
   <xs:element name="STATUS_MSG">
     <xs:complexType>
       <xs:sequence>
         <xs:element ref="xfb:BGPMON"/>
         <xs:choice>
           <xs:element ref="xfb:SESSION_STATUS"
                       minOccurs="0" maxOccurs="1"/>
           <xs:element ref="xfb:CHAIN_STATUS"
                       minOccurs="0" maxOccurs="1"/>
           <xs:element minOccurs="0" maxOccurs="1"
                       ref="xfb:QUEUE_STATUS"/>
           <xs:element ref="xfb:BGPMON_STATUS"
                       minOccurs="0" maxOccurs="1"/>
         </xs:choice>
       </xs:sequence>
     </xs:complexType>
   </xs:element>
   <xs:element name="OPEN">
     <xs:complexType>
       <xs:sequence>
         <xs:element name="VERSION" type="xs:string"/>
         <xs:element name="SRC_AS" type="xfb:asn"/>
         <xs:element name="HOLD_TIME" type="xs:nonNegativeInteger"/>
         <xs:element name="SRC_BGP" type="xfb:address"/>
         <xs:element name="OPT_PAR_LEN" type="xs:nonNegativeInteger"/>
         <xs:element ref="xfb:OPT_PAR"/>
       </xs:sequence>
     </xs:complexType>
   </xs:element>
   <xs:element name="OPT_PAR">
```

```
     <xs:complexType>
       <xs:sequence>
         <xs:element maxOccurs="unbounded"
                     minOccurs="0" ref="xfb:PARAMETER"/>
       </xs:sequence>
       <xs:attribute name="count" type="xs:nonNegativeInteger"
                     use="required"/>
     </xs:complexType>
   </xs:element>
   <xs:element name="PARAMETER">
     <xs:complexType>
       <xs:sequence>
         <xs:element name="LENGTH" type="xs:nonNegativeInteger"/>
         <xs:element name="TYPE" type="xfb:par_type"/>
         <xs:choice>
           <xs:element ref="xfb:AUTHENTICATION"/>
           <xs:element ref="xfb:CAPABILITIES"/>
           <xs:element ref="xfb:OTHER"/>
         </xs:choice>
       </xs:sequence>
       <xs:attribute name="code" type="xs:string"/>
     </xs:complexType>
   </xs:element>
   <xs:element name="AUTHENTICATION">
     <xs:complexType>
       <xs:simpleContent>
         <xs:extension base="xs:hexBinary">
           <xs:attribute name="code" type="xs:string"/>
         </xs:extension>
       </xs:simpleContent>
     </xs:complexType>
   </xs:element>
   <xs:element name="CAPABILITIES">
     <xs:complexType>
       <xs:sequence>
         <xs:element minOccurs="0" maxOccurs="unbounded" name="CAP">
           <xs:complexType>
             <xs:sequence>
               <xs:element name="CODE" type="xs:string"/>
               <xs:element name="LENGTH" type="xs:long"/>
               <xs:element name="DATA" type="xs:string"/>
             </xs:sequence>
           </xs:complexType>
         </xs:element>
       </xs:sequence>
       <xs:attribute name="count" type="xs:integer" use="required"/>
     </xs:complexType>
   </xs:element>
   <xs:element name="UNKNOWN" type="xs:hexBinary"/>
   <xs:element name="UPDATE">
     <xs:complexType>
       <xs:sequence>
         <xs:element name="WITHDRAWN_LEN" type="xs:nonNegativeInteger"/>
         <xs:element ref="xfb:WITHDRAWN"/>
         <xs:element name="PATH_ATTRIBUTES_LEN"
                     type="xs:nonNegativeInteger"/>
         <xs:element ref="xfb:PATH_ATTRIBUTES"/>
         <xs:element ref="xfb:NLRI"/>
       </xs:sequence>
     </xs:complexType>
   </xs:element>
   <xs:element name="PATH_ATTRIBUTES">
     <xs:complexType>
       <xs:sequence>
         <xs:element minOccurs="0" maxOccurs="unbounded"
                     ref="xfb:ATTRIBUTE"/>
       </xs:sequence>
       <xs:attribute name="count" type="xs:nonNegativeInteger"/>
     </xs:complexType>
   </xs:element>
   <xs:element name="ATTRIBUTE">
     <xs:complexType>
       <xs:sequence>
         <xs:element name="FLAGS">
           <xs:complexType>
             <xs:sequence>
               <xs:element name="OPTIONAL"
                           minOccurs="0" maxOccurs="1"/>
               <xs:element name="TRANSITIVE"
                           minOccurs="0" maxOccurs="1"/>
               <xs:element name="PARTIAL" minOccurs="0" maxOccurs="1"/>
               <xs:element name="EXTENDED" minOccurs="0" maxOccurs="1"/>
             </xs:sequence>
             <xs:attribute name="code" type="xs:string"/>
           </xs:complexType>
         </xs:element>
         <xs:element name="LENGTH" type="xs:long"/>
         <xs:element name="TYPE" type="xfb:attr_type"/>
         <xs:choice>
           <xs:element ref="xfb:ORIGIN"/>
           <xs:element ref="xfb:AS_PATH"/>
           <xs:element ref="xfb:NEXT_HOP"/>
           <xs:element ref="xfb:MULTI_EXIT_DISC"/>
           <xs:element ref="xfb:LOCAL_PREF"/>
           <xs:element ref="xfb:ATOMIC_AGGREGATE"/>
           <xs:element ref="xfb:AGGREGATOR"/>
           <xs:element ref="xfb:COMMUNITIES"/>
           <xs:element ref="xfb:ORIGINATOR_ID"/>
           <xs:element ref="xfb:CLUSTER_LIST"/>
           <xs:element ref="xfb:ADVERTISER"/>
           <xs:element ref="xfb:RCID_PATH"/>
           <xs:element ref="xfb:MP_REACH_NLRI"/>
           <xs:element ref="xfb:MP_UNREACH_NLRI"/>
           <xs:element ref="xfb:EXTENDED_COMMUNITIES"/>
           <xs:element ref="xfb:AS4_PATH"/>
           <xs:element ref="xfb:AS4_AGGREGATOR"/>
           <xs:element ref="xfb:OTHER"/>
         </xs:choice>
       </xs:sequence>
     </xs:complexType>
   </xs:element>
   <xs:element name="ORIGIN" type="xfb:origin_type"/>
   <xs:element name="AS_PATH" type="xfb:as_path"/>
   <xs:element name="NEXT_HOP" type="xfb:address"/>
   <xs:element name="MULTI_EXIT_DISC" type="xs:integer"/>
   <xs:element name="LOCAL_PREF" type="xs:integer"/>
   <xs:element name="ATOMIC_AGGREGATE"/>
   <xs:element name="AGGREGATOR" type="xfb:aggregator"/>
   <xs:element name="COMMUNITIES">
     <xs:complexType>
       <xs:choice minOccurs="0" maxOccurs="unbounded">
         <xs:element name="NO_EXPORT"/>
         <xs:element name="NO_ADVERTISE"/>
         <xs:element name="NO_EXPORT_SUBCONFED"/>
         <xs:element ref="xfb:COMMUNITY"/>
         <xs:element ref="xfb:RESERVED_COMMUNITY"/>
       </xs:choice>
     </xs:complexType>
   </xs:element>
   <xs:element name="COMMUNITY" type="xfb:community_type"/>
   <xs:element name="RESERVED_COMMUNITY" type="xfb:community_type"/>
   <xs:element name="ORIGINATOR_ID" type="xs:integer"/>
   <xs:element name="CLUSTER_LIST">
     <xs:complexType>
       <xs:sequence>
         <xs:element name="ID" type="xs:string"
                     minOccurs="0" maxOccurs="unbounded"/>
       </xs:sequence>
       <xs:attribute name="count" type="xs:integer"/>
     </xs:complexType>
   </xs:element>
   <xs:element name="ADVERTISER" type="xfb:address"/>
   <xs:element name="RCID_PATH">
     <xs:complexType>
```

```
       <xs:sequence>
         <xs:element name="ID" type="xs:string"
                     minOccurs="0" maxOccurs="unbounded"/>
       </xs:sequence>
       <xs:attribute name="count" type="xs:integer"/>
     </xs:complexType>
   </xs:element>
   <xs:element name="MP_REACH_NLRI">
     <xs:complexType>
       <xs:sequence>
         <xs:element name="AFI" type="xfb:afi_type"/>
         <xs:element name="SAFI" type="xfb:safi_type"/>
         <xs:element ref="xfb:NEXT_HOP"/>
         <xs:element name="SNPA_LIST_LEN" type="xs:nonNegativeInteger"/>
         <xs:element ref="xfb:SNPA_LIST"/>
         <xs:element ref="xfb:NLRI"/>
       </xs:sequence>
     </xs:complexType>
   </xs:element>
   <xs:element name="SNPA_LIST">
     <xs:complexType>
       <xs:sequence>
         <xs:element name="SNPA" type="xfb:prefix"
                     minOccurs="0" maxOccurs="unbounded"/>
       </xs:sequence>
       <xs:attribute name="count" type="xs:nonNegativeInteger"/>
     </xs:complexType>
   </xs:element>
   <xs:element name="MP_UNREACH_NLRI">
     <xs:complexType>
       <xs:sequence>
         <xs:element name="AFI" type="xfb:afi_type"/>
         <xs:element name="SAFI" type="xfb:safi_type"/>
         <xs:element ref="xfb:WITHDRAWN"/>
       </xs:sequence>
     </xs:complexType>
   </xs:element>
   <xs:element name="EXTENDED_COMMUNITIES" type="xfb:octets_attribute"/>
   <xs:element name="AS4_PATH" type="xfb:as_path"/>
   <xs:element name="AS4_AGGREGATOR" type="xfb:aggregator"/>
   <xs:element name="OTHER">
     <xs:complexType>
       <xs:sequence>
         <xs:element name="OCTETS" type="xfb:octets"/>
       </xs:sequence>
     </xs:complexType>
   </xs:element>
   <xs:element name="WITHDRAWN" type="xfb:prefixes"/>
   <xs:element name="NLRI" type="xfb:prefixes"/>
   <xs:element name="NOTIFICATION">
     <xs:complexType>
       <xs:sequence>
         <xs:element name="CODE" type="xfb:error_type"/>
         <xs:element name="SUBCODE" type="xfb:error_type"/>
         <xs:element name="DATA" type="xs:string"/>
       </xs:sequence>
     </xs:complexType>
   </xs:element>
   <xs:element name="KEEPALIVE"/>
   <xs:element name="ROUTE_REFRESH">
     <xs:complexType>
       <xs:attributeGroup ref="xfb:attr_group_afi"/>
     </xs:complexType>
   </xs:element>
   <xs:element name="CISCO_ROUTE_REFRESH"/>
   <xs:element name="BGPMON_STATUS">
     <xs:complexType>
       <xs:choice>
         <xs:element name="START"/>
         <xs:element name="STOP"/>
       </xs:choice>
     </xs:complexType>
   </xs:element>
   <xs:element name="QUEUE_STATUS">
     <xs:complexType>
       <xs:sequence>
         <xs:element minOccurs="0" maxOccurs="unbounded"
                     ref="xfb:QUEUE"/>
       </xs:sequence>
       <xs:attribute name="count" type="xs:nonNegativeInteger"/>
     </xs:complexType>
   </xs:element>
   <xs:element name="QUEUE">
     <xs:complexType>
       <xs:sequence>
         <xs:element name="NAME" type="xs:string"/>
         <xs:element name="ITEM" type="xfb:stat_type"/>
         <xs:element name="WRITER" type="xfb:stat_type"/>
         <xs:element name="READER" type="xfb:stat_type"/>
         <xs:element name="PACING">
           <xs:complexType>
             <xs:sequence>
               <xs:element name="FLAG" type="xs:string"/>
               <xs:element name="COUNT" type="xfb:stat_type"/>
               <xs:element type="xfb:stat_type" name="WRITE_LIMIT"/>
             </xs:sequence>
           </xs:complexType>
         </xs:element>
       </xs:sequence>
     </xs:complexType>
   </xs:element>
   <xs:element name="CHAIN_STATUS">
     <xs:complexType>
       <xs:sequence>
         <xs:element ref="xfb:CHAIN"
                     minOccurs="0" maxOccurs="unbounded"/>
       </xs:sequence>
       <xs:attribute name="count" type="xs:nonNegativeInteger"/>
     </xs:complexType>
   </xs:element>
   <xs:element name="CHAIN">
     <xs:complexType>
       <xs:sequence>
         <xs:group ref="PEERING_STAT"/>
       </xs:sequence>
     </xs:complexType>
   </xs:element>
   <xs:element name="SESSION_STATUS">
     <xs:complexType>
       <xs:sequence>
         <xs:element ref="xfb:SESSION"
                     minOccurs="0" maxOccurs="unbounded"/>
       </xs:sequence>
       <xs:attribute name="count" type="xs:nonNegativeInteger"/>
     </xs:complexType>
   </xs:element>
   <xs:element name="SESSION">
     <xs:complexType>
       <xs:sequence>
         <xs:group ref="xfb:PEERING_STAT"/>
         <xs:group ref="xfb:SESSION_STAT"
                   minOccurs="0" maxOccurs="1"/>
       </xs:sequence>
     </xs:complexType>
   </xs:element>
   <xs:element name="BGPMON">
     <xs:complexType>
       <xs:sequence>
         <xs:element name="ADDR" type="xfb:address"/>
         <xs:element name="PORT" type="xfb:port"/>
         <xs:element type="xfb:asn" name="AS"
                     minOccurs="0" maxOccurs="1"/>
       </xs:sequence>
     </xs:complexType>
   </xs:element>
   <xs:group name="PEERING_STAT">
     <xs:sequence>
       <xs:element type="xfb:address" name="ADDR"/>
       <xs:element name="PORT" type="xfb:port"
                   minOccurs="0" maxOccurs="1"/>
       <xs:element name="AS" type="xfb:asn"
                   minOccurs="0" maxOccurs="1"/>
       <xs:element name="STATE" type="xs:integer"
                   minOccurs="0" maxOccurs="1"/>
       <xs:element name="STATE_CHANGE" minOccurs="0" maxOccurs="1">
         <xs:complexType>
           <xs:sequence>
             <xs:element name="OLD_STATE" type="xs:integer"/>
             <xs:element name="NEW_STATE" type="xs:integer"/>
             <xs:element name="REASON" type="xs:integer"/>
           </xs:sequence>
         </xs:complexType>
       </xs:element>
       <xs:element name="OPTIME" type="xfb:time_type"
                   minOccurs="0" maxOccurs="1"/>
       <xs:element name="RECV_MESSAGE" type="xfb:stat_type"
                   minOccurs="0" maxOccurs="1"/>
       <xs:element name="RESET" type="xfb:stat_type"
                   minOccurs="0" maxOccurs="1"/>
     </xs:sequence>
   </xs:group>
   <xs:group name="SESSION_STAT">
     <xs:sequence>
       <xs:element name="PREFIX"/>
       <xs:element name="ATTRIBUTE"/>
       <xs:element name="MEMORY_USAGE"/>
       <xs:element name="ANNOUNCEMENT" type="xfb:stat_type"/>
       <xs:element name="DUP_ANNOUNCEMENT" type="xfb:stat_type"/>
       <xs:element name="SAME_PATH" type="xfb:stat_type"/>
       <xs:element name="DIFF_PATH" type="xfb:stat_type"/>
       <xs:element name="WITHDRAWAL" type="xfb:stat_type"/>
       <xs:element name="DUP_WITHDRAWAL" type="xfb:stat_type"/>
     </xs:sequence>
   </xs:group>
   <xs:complexType name="address">
     <xs:simpleContent>
       <xs:extension base="xs:string">
         <xs:attributeGroup ref="attr_group_afi"/>
       </xs:extension>
     </xs:simpleContent>
   </xs:complexType>
   <xs:complexType name="prefix">
```

```
     <xs:simpleContent>
       <xs:extension base="xs:string">
         <xs:attribute name="label" type="xs:string"/>
         <xs:attributeGroup ref="attr_group_afi"/>
       </xs:extension>
     </xs:simpleContent>
   </xs:complexType>
   <xs:complexType name="prefixes">
     <xs:sequence>
       <xs:element minOccurs="0" maxOccurs="unbounded"
                   type="xfb:prefix" name="PREFIX"/>
     </xs:sequence>
     <xs:attribute name="count" type="xs:nonNegativeInteger"/>
   </xs:complexType>
   <xs:simpleType name="port">
     <xs:restriction base="xs:integer">
       <xs:maxInclusive value="65535"/>
       <xs:minInclusive value="0"/>
     </xs:restriction>
   </xs:simpleType>
   <xs:simpleType name="asn">
     <xs:restriction base="xs:nonNegativeInteger"/>
   </xs:simpleType>
   <xs:complexType name="as_path">
     <xs:sequence>
       <xs:element name="AS" type="xfb:asn"
                   minOccurs="1" maxOccurs="unbounded"/>
     </xs:sequence>
     <xs:attribute name="type" type="xfb:enum_as_path_type"/>
   </xs:complexType>
   <xs:complexType name="aggregator">
     <xs:sequence>
       <xs:element name="AS" type="xfb:asn"/>
       <xs:element name="ADDR" type="xfb:address"/>
     </xs:sequence>
   </xs:complexType>
   <xs:complexType name="octets">
     <xs:simpleContent xml:base="xs:hexBinary">
       <xs:extension base="xs:string">
         <xs:attribute name="length" type="xs:nonNegativeInteger"
                       use="optional"/>
       </xs:extension>
     </xs:simpleContent>
   </xs:complexType>
   <xs:complexType name="bgp_type">
     <xs:simpleContent>
       <xs:extension base="enum_bgp_type">
         <xs:attribute type="xs:nonNegativeInteger" name="value"/>
       </xs:extension>
     </xs:simpleContent>
   </xs:complexType>
   <xs:complexType name="origin_type">
     <xs:simpleContent>
       <xs:extension base="enum_origin_type">
         <xs:attribute type="xs:nonNegativeInteger" name="value"/>
       </xs:extension>
     </xs:simpleContent>
   </xs:complexType>
   <xs:complexType name="par_type">
     <xs:simpleContent>
       <xs:extension base="enum_par_type">
         <xs:attribute type="xs:nonNegativeInteger" name="value"/>
       </xs:extension>
     </xs:simpleContent>
   </xs:complexType>
   <xs:complexType name="afi_type">
     <xs:simpleContent>
       <xs:extension base="enum_afi_type">
         <xs:attribute type="xs:nonNegativeInteger" name="value"/>
       </xs:extension>
     </xs:simpleContent>
   </xs:complexType>
   <xs:complexType name="safi_type">
     <xs:simpleContent>
       <xs:extension base="enum_safi_type">
         <xs:attribute type="xs:nonNegativeInteger" name="value"/>
       </xs:extension>
     </xs:simpleContent>
   </xs:complexType>
   <xs:complexType name="attr_type">
     <xs:simpleContent>
       <xs:extension base="xs:string">
         <xs:attribute type="xs:nonNegativeInteger" name="value"/>
       </xs:extension>
     </xs:simpleContent>
   </xs:complexType>
   <xs:complexType name="error_type">
     <xs:simpleContent>
       <xs:extension base="xs:string">
         <xs:attribute type="xs:nonNegativeInteger" name="value"/>
       </xs:extension>
     </xs:simpleContent>
   </xs:complexType>
   <xs:complexType name="community_type">
     <xs:sequence>
       <xs:element name="AS" type="xfb:asn"/>
       <xs:element name="VALUE" type="xs:nonNegativeInteger"/>
     </xs:sequence>
   </xs:complexType>
   <xs:complexType name="octets_attribute">
     <xs:sequence>
       <xs:element type="xfb:octets" name="OCTETS"/>
     </xs:sequence>
   </xs:complexType>
   <xs:simpleType name="enum_yes_no_type">
     <xs:restriction base="xs:string">
       <xs:enumeration value="YES"/>
       <xs:enumeration value="NO"/>
     </xs:restriction>
   </xs:simpleType>
   <xs:simpleType name="enum_origin_type">
     <xs:restriction base="xs:NMTOKEN">
       <xs:enumeration value="IGP"/>
       <xs:enumeration value="BGP"/>
       <xs:enumeration value="INCOMPLETE"/>
       <xs:enumeration value="OTHER"/>
     </xs:restriction>
   </xs:simpleType>
   <xs:simpleType name="enum_par_type">
     <xs:restriction base="xs:NMTOKEN">
       <xs:enumeration value="AUTHENTICATION"/>
       <xs:enumeration value="CAPABILITIES"/>
       <xs:enumeration value="OTHER"/>
     </xs:restriction>
   </xs:simpleType>
   <xs:simpleType name="enum_bgp_type">
     <xs:restriction base="xs:NMTOKEN">
       <xs:enumeration value="OPEN"/>
       <xs:enumeration value="UPDATE"/>
       <xs:enumeration value="NOTIFICATION"/>
       <xs:enumeration value="KEEPALIVE"/>
       <xs:enumeration value="ROUTE_REFRESH"/>
       <xs:enumeration value="CISCO_ROUTE_REFRESH"/>
       <xs:enumeration value="UNKNOWN"/>
     </xs:restriction>
   </xs:simpleType>
   <xs:simpleType name="enum_afi_type">
     <xs:restriction base="xs:NMTOKEN">
       <xs:enumeration value="IPV4"/>
       <xs:enumeration value="IPV6"/>
       <xs:enumeration value="OTHER"/>
     </xs:restriction>
   </xs:simpleType>
   <xs:simpleType name="enum_as_path_type">
```

```
     <xs:restriction base="xs:NMTOKEN">
       <xs:enumeration value="as_sequence"/>
       <xs:enumeration value="as_set"/>
     </xs:restriction>
   </xs:simpleType>
   <xs:simpleType name="enum_safi_type">
     <xs:restriction base="xs:NMTOKEN">
       <xs:enumeration value="NLRI_UNICAST"/>
       <xs:enumeration value="NLRI_MULTICAST"/>
       <xs:enumeration value="NLRI_MPLS"/>
       <xs:enumeration value="OTHER"/>
     </xs:restriction>
   </xs:simpleType>
   <xs:complexType name="stat_type">
     <xs:simpleContent>
       <xs:extension base="xs:float">
         <xs:attributeGroup ref="xfb:attr_group_stat"/>
       </xs:extension>
     </xs:simpleContent>
   </xs:complexType>
   <xs:complexType name="time_type">
     <xs:simpleContent>
       <xs:extension base="xs:float">
         <xs:attributeGroup ref="xfb:attr_group_time"/>
       </xs:extension>
     </xs:simpleContent>
   </xs:complexType>
   <xs:attributeGroup name="attr_group_stat">
     <xs:attribute name="avg" use="optional" type="xs:float"/>
     <xs:attribute name="sdv" use="optional" type="xs:float"/>
     <xs:attribute name="min" type="xs:float" use="optional"/>
     <xs:attribute name="max" type="xs:float" use="optional"/>
     <xs:attribute name="accu" type="xs:float"/>
     <xs:attribute name="limit" type="xs:float" use="optional"/>
   </xs:attributeGroup>
   <xs:attributeGroup name="attr_group_time">
     <xs:attribute name="last_startup" use="optional" type="xs:long"/>
     <xs:attribute name="last_down" use="optional" type="xs:long"/>
     <xs:attribute name="last_action" use="optional" type="xs:long"/>
   </xs:attributeGroup>
   <xs:attributeGroup name="attr_group_afi">
     <xs:attribute name="afi" type="xfb:enum_afi_type" use="optional"/>
     <xs:attribute name="afi_value" type="xs:nonNegativeInteger"
                   use="optional"/>
     <xs:attribute name="safi" type="xfb:enum_safi_type"
                   use="optional"/>
     <xs:attribute name="safi_value" type="xs:nonNegativeInteger"
                   use="optional"/>
   </xs:attributeGroup>
 </xs:schema>
```

8. Acknowledgements

9. IANA Considerations

This document uses URNs to describe an XML namespace and schema. Two registrations are needed: (1) registration for the XFB namespace: urn:ietf:params:xml:ns:xfb-0.1 and (2) registration for the XFB XML schema: urn:ietf:params:xml:schema:xfb-0.1

10. Security Considerations

The XFB format untilizes XML to flexibly represent BGP information. The XFB document structure and fields are only descriptive and do not create additional security risks.

11. References

11.1. Normative References

[I-D.ietf-grow-mrt]
Blunk, L., Karir, M., and C. Labovitz, "MRT routing information export format", draft-ietf-grow-mrt-08 (work in progress), July 2008.

[RFC1997] Chandrasekeran, R., Traina, P., and T. Li, "BGP Communities Attribute", RFC 1997, August 1996.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2918] Chen, E., "Route Refresh Capability for BGP-4", RFC 2918, September 2000.

[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.

[RFC4893] Vohra, Q. and E. Chen, "BGP Support for Four-octet AS Number Space", RFC 4893, May 2007.

11.2. Informative References

[Rosen] Rosen, Eric., "draft-ramachandra-bgp-ext-communities-00.txt", March 1999.

URIs

[1] <http://bgpmon.netsec.colostate.edu/>

[2] <http://www.w3.org/TR/xmlschema-2/>

Appendix A. Storage Size Comparison

Experiment results are promising using the default compression parameters for bzip2. As shown in the table below, the uncompressed XFB require more space, but the compressed XFB require less space. Compared with bgpdump format, less storage will be used in compressed XFB format. Even compared with MRT [I-D.ietf-grow-mrt] binary format, XFB almost consumes the same storage. This table shows the size of data used in the above comparison, using MRT as the baseline.

format	umcompressed	ratio	compressed	ratio
XFB	74389091	7.22	2200472	1.03
bgpdump	54466310	5.29	2418845	1.13
MRT	10298545	1.0	2142657	1.00

Storage Size Comparison

Authors' Addresses

Peichun Cheng (editor)
UCLA IRL
Los Angeles, CA
US

Phone:
Fax:
Email: pccheng@cs.ucla.edu
URI:

He Yan (editor)
CSU-Netsec Group
Fort Collins, CO
US

Phone:
Fax:
Email: yanhe@cs.colostate.edu
URI:

Kevin Burnett (editor)
CSU-Netsec Group
Fort Collins, CO
US

Phone:
Fax:
Email: burnet@cs.colostate.edu
URI:

Daniel F. Massey (editor)
CSU-Netsec Group
Fort Collins, CO
US

Phone:
Email: massey@cs.colostate.edu

Lixia Zhang (editor)
UCLA IRL
Los Angeles, CA
US

Phone:
Fax:
Email: lixia@cs.ucla.edu
URI: