Network Working Group   Pradosh Mohapatra
Internet Draft   Cisco Systems, Inc.
Expiration Date: February 2008

                                                      Eric Rosen
                                               Cisco Systems, Inc.

                                                     August 2007

       BGP Encapsulation SAFI and BGP Tunnel Encapsulation Attribute

                    draft-ietf-idr-encaps-safi-00.txt

Status of this Memo

Abstract

   In certain situations, transporting a packet from one BGP speaker to
   another, the BGP next hop, requires that the packet be encapsulated
   by the first BGP speaker and decapsulated by the second. To support
   these situations, there needs to be some agreement between the two
   BGP speakers with regard to the "encapsulation information", i.e.,
   the format of the encapsulation header as well as the contents of
   various fields of the header.

   The encapsulation information need not be signaled for all

encapsulation types. In the cases where the signaling is required
(such as L2TPv3, GRE with key), This draft specifies a method by
which BGP speakers can signal encapsulation information to each
other. The signaling is done by sending BGP updates using the
"Encapsulation SAFI" and IPv4 or IPv6 AFI. In the cases where no
encapsulation information needs to be signaled (such as GRE without
key), this draft specifies a BGP extended community that can be
attached to UPDATE messages that carry payload prefixes to indicate
the encapsulation protocol type to be used.

Table of Contents

1. Specification of requirements

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT",
"SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this
document are to be interpreted as described in [RFC2119].

2. Introduction

   Consider the case of a router R1 forwarding an IP packet P. Let D be
   P's IP destination address. R1 must look up D in its forwarding
   table. Suppose that the "best match" route for D is route Q, where Q
   is a BGP-distributed route whose "BGP next hop" is router R2. And
   suppose further that the routers along the path from R1 to R2 have
   entries for R2 in their forwarding tables, but do NOT have entries
   for D in their forwarding tables. For example, the path from R1 to R2
   may be part of a "BGP-free core", where there are no BGP-distributed
   routes at all in the core. Or, as in [Softwires-Mesh-Frame-work], D
   may be an IPv4 address while the intermediate routers along the path
   from R1 to R2 may support only IPv6.

   In cases such as this, in order for R1 to properly forward packet P,
   it must encapsulate P, and send P "through a tunnel" to R2. For
   example, R1 may encapsulate P using GRE, L2TPv3, IP-in-IP, etc.,
   where the destination IP address of the encapsulation header is the
   address of R2.

   In order for R1 to encapsulate P for transport to R2, R1 must know
   what encapsulation protocol to use for transporting what sorts of
   packets to R2. R1 must also know how to fill in the various fields of
   the encapsulation header. With certain encapsulation types, this
   knowledge may be acquired by default or through manual configuration.
   Other encapsulation protocols have fields such as session id, key, or
   cookie which must be filled in. It would not be desirable to require
   every BGP speaker to be manually configured with the encapsulation
   information for every one of its BGP next hops.

   In this draft, we specify a way in which BGP itself can be used by a
   given BGP speaker to tell other BGP speakers, "if you need to
   encapsulate packets to be sent to me, here's the information you need
   to properly form the encapsulation header". A BGP speaker signals
   this information to other BGP speakers by using a distinguished SAFI
   value, the Encapsulation SAFI. The encapsulation SAFI can be used
   with the AFI for IPv4 or with the AFI for IPv6. The IPv4 AFI is used
   when the encapsulated packets are to be sent using IPv4; the IPv6 AFI
   is used when the encapsulated packets are to be sent using IPv6.

   In a given BGP update, the NLRI of the encapsulation SAFI consists of
   the IP address (in the family specified by the AFI) of the originator
   of that update. The encapsulation information is specified in one or
   more BGP "tunnel encapsulation attributes" (specified herein). These
   attributes specify the encapsulation protocols that may be used, as
   well as specifying whatever additional information (if any) is needed
   in order to properly use those protocols. Other attributes, e.g.,
   communities or extended communities, may also be included.

Since the encapsulation information is coded as a set of attributes,
one could ask whether a new SAFI is really required. After all, a
BGP speaker could simply attach the tunnel encapsulation attributes
to each prefix (like Q in our example) that it advertises. But with
that technique, any change in the encapsulation information would
cause a very large number of updates. Unless one really wants to
specify different encapsulation information for each prefix, it is
much better to have a   mechanism in which a change in the
encapsulation information causes a BGP speaker to advertise only a
single update. Conversely, when prefixes get modified, the tunnel
encapsulation information need not be exchanged.

In this specification, a single SAFI is used to carry information for
all encapsulation protocols. One could have taken an alternative
approach of defining a new SAFI for each encapsulation protocol.
However, with the specified approach, encapsulation information can
pass transparently and automatically through intermediate BGP
speakers (e.g., route reflectors) that do not necessarily understand
the encapsulation information. This works because the encapsulation
attribute is defined as an optional transitive attribute. New
encapsulations can thus be added without the need to reconfigure any
intermediate BGP system. If adding a new encapsulation required using
a new SAFI, the information for that encapsulation would not pass
through intermediate BGP systems unless those systems were
reconfigured to support the new SAFI.

For encapsulation protocols where no encapsulation information needs
to be signaled (such as GRE without key), the egress router MAY still
want to specify the protocol to use for transporting packets from the
ingress router. This draft specifies a new BGP extended community
that can be attached to UPDATE messages that carry payload prefixes
for this purpose.

3. Encapsulation NLRI Format

The NLRI, defined below, is carried in BGP UPDATE messages [RFC4271]
using BGP multiprotocol extensions [RFC4760] with an AFI of 1 or 2
(IPv4 or IPv6) [IANA-AF] and a SAFI value to be assigned by IANA
(called as Encapsulation SAFI).

The NLRI is encoded in a format as defined in section 5 of [RFC4760]
(a 2-tuple of the form <length, value>). The value field is
structured as follows:

```
               +------------------------------------------------+
               |          Endpoint address (Variable)           |
               +------------------------------------------------+
```

   - Endpoint Address: This field identifies the BGP speaker
     originating the update. It is typically one of the interface
     addresses configured at the router. The length of the endpoint
     address is dependent on the AFI being advertised. If the AFI is
     set to IPv4 (1), the the endpoint address is a 4-octet IPv4
     address whereas if the AFI is set to IPv6 (2), the endpoint
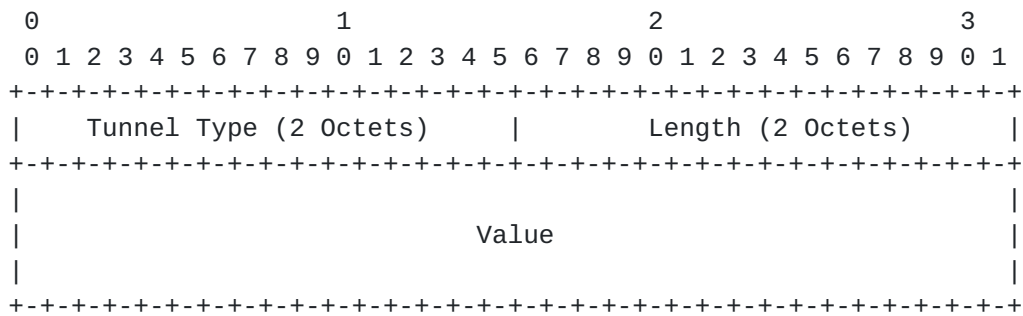   address is a 16-octet IPv6 address.

   An update message that carries the MP_REACH_NLRI or MP_UNREACH_NLRI
   with Encapsulation SAFI MUST also carry the BGP mandatory attributes:
   ORIGIN, AS_PATH, and LOCAL_PREF (for IBGP neighbors) as defined in
   [RFC4271]. In addition, such an update message can also contain any
   of the BGP optional attributes, like Community or Extended Community
   attribute to influence an action on the receiving speaker.

   When a BGP speaker advertises the Encapsulation NLRI via BGP, it uses
   its own address as the BGP nexthop in the MP_REACH_NLRI or
   MP_UNREACH_NLRI attribute. The nexthop address is set based on the
   AFI in the attribute. For example, if the AFI is set to IPv4 (1),
   the nexthop is encoded as a 4-byte IPv4 address. If the AFI is set to
   IPv6 (2), the nexthop is encoded as a 16-byte IPv6 address of the
   router. On the receiving router, the BGP nexthop of such an update
   message is validated by performing a recursive route lookup operation
   in the routing table.

   Bestpath selection of Encapsulation NLRIs is governed by the decision
   process outlined in section 9.1 of [RFC4271]. The encapsulation data
   carried through other attributes in the message are to be used by the
   receiving router only if the NLRI has a bestpath.

4. Tunnel Encapsulation Attribute

   Tunnel Encapsulation attribute is an optional transitive attribute
   that is composed of a set of TLVs. The type code of the attribute is
   to be assigned by IANA. Each TLV contains information corresponding
   to a particular tunnel technology. The TLV is structured 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
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |     Tunnel Type (2 Octets)     |       Length (2 Octets)      |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                                                               |
    |                             Value                             |
    |                                                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
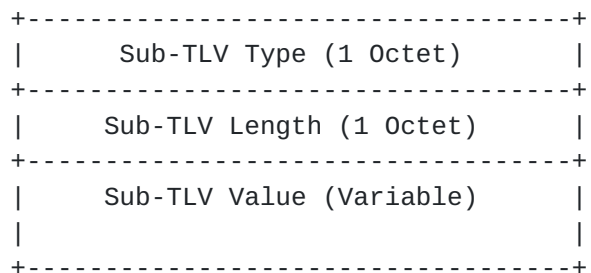
Tunnel Type (2 octets): It identifies the type of the tunneling
technology being signaled. This document defines the following types:

   - L2TPv3: Tunnel Type = 1

   - GRE: Tunnel Type = 2

Unknown types are to be ignored and skipped upon receipt.

Length (2 octets): the total number of octets of the Value field.

Value (variable): The value is comprised of multiple sub-TLV's. Each
sub-TLV consists of three fields: a one-octet type, one-octet length,
and zero or more octets of value. The sub-TLV is structured as
follows:

```
                +----------------------------------+
                |        Sub-TLV Type (1 Octet)    |
                +----------------------------------+
                |      Sub-TLV Length (1 Octet)    |
                +----------------------------------+
                |      Sub-TLV Value (Variable)    |
                |                                  |
                +----------------------------------+
```

Sub-TLV Type (1 octet): Each sub-TLV type defines a certain property
about the tunnel TLV that contains this sub-TLV. The following are
the types defined in this document:

   - Encapsulation: sub-TLV type = 1

   - Protocol type: sub-TLV type = 2

When the TLV is being processed by a BGP speaker that will be
performing encapsulation, any unknown sub-TLVs MUST be ignored and
skipped. However if the TLV is understood, the entire TLV MUST NOT be
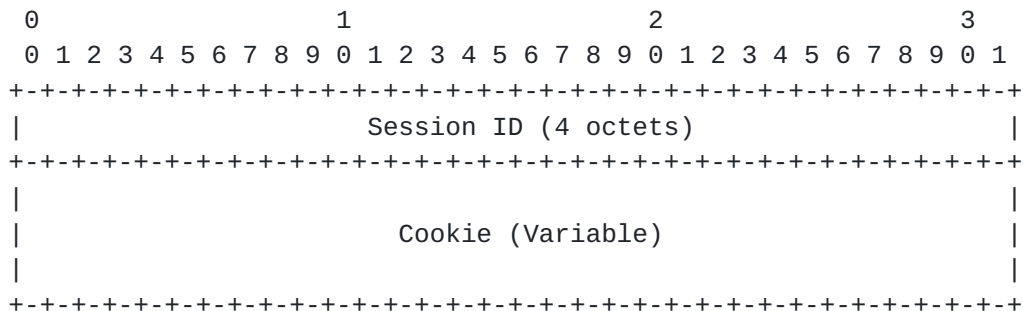ignored just because it contains an unknown sub-TLV.

   Sub-TLV Length (1 octet): the total number of octets of the sub-TLV
   value field.

   Sub-TLV Value (variable): Encodings of the value field depend on the
   sub-TLV type as enumerated above. The following sub-sections define
   the encoding in detail.

4.1. Encapsulation sub-TLV

   The syntax and semantics of the encapsulation sub-TLV is determined
   by the tunnel type of the TLV that contains this sub-TLV.

   When the tunnel type of the TLV is L2TPv3, the following is the
   structure of the value field of the encapsulation sub-TLV:


```
     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
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                    Session ID (4 octets)                     |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                                                              |
    |                     Cookie (Variable)                        |
    |                                                              |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```


   * Session ID: a 4-octet value locally assigned by the advertising
     router that serves as a lookup key in the incoming packet's
     context.

   * Cookie: an optional, variable length (encoded in octets - 0 to 8
     octets) value used by L2TPv3 to check the association of a
     received data message with the session identified by the Session
     ID. The Cookie value is tightly coupled with the Session ID.

     The length of the cookie is not encoded explicitly, but can be
     calculated as: (sub-TLV length - 4)

   When the tunnel type of the TLV is GRE, the following is the
   structure of the value field of the encapsulation sub-TLV:

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       GRE Key (4 octets)                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

   * GRE Key: A 4 Octet field that is generated by the advertising
     router. The actual method by which the key is obtained is beyond
     the scope of the document. The key is inserted into the GRE
     encapsulation header of the payload packets sent by ingress
     routers to the advertising router. It is intended to be used for
     identifying extra context information about the received payload.

     Note that the key is optional. Unless a key value is being
     advertised, the GRE encapsulation sub-TLV MUST NOT be present.

4.2. Protocol Type sub-TLV

   The protocol type sub-TLV MAY be encoded to indicate the type of the
   payload packets that will be encapsulated with the tunnel parameters
   being signaled in the TLV. The value field of the sub-TLV contains a
   2-octet protocol type that is one of the types defined in [IANA-AF]
   as ETHER TYPEs.

   For example, if we want to use three L2TPv3 sessions, one carrying
   IPv4 packets, one carrying IPv6 packets, and one carrying MPLS
   packets, the egress router will include three TLVs of L2TPv3
   encapsulation type, each specifying a different session id and a
   different payload type. The protocol type sub-TLV for these will be
   IPv4 (protocol type = 0x0800), IPv6 (protocol type = 0x86dd), and
   MPLS (protocol type = 0x8847) respectively. This informs the ingress
   routers of the appropriate encapsulation information to use with each
   of the given protocol types. Insertion of the specified session id at
   the ingress routers allows the egress to process the incoming packets
   correctly, according to their protocol type.

   Note that the protocol type sub-TLV is optional, e.g. if the
   tunneling technology is GRE, this sub-TLV is not required.

[4.3](4.3). **Tunnel Type Selection**

   A BGP speaker may include multiple tunnel TLVs in the tunnel
   attribute. The receiving speaker MAY have local policies defined to
   choose different tunnel types for different sets/types of payload
   prefixes received from the same BGP speaker. For instance, if a BGP
   speaker includes both L2TPv3 and GRE tunnel types in the tunnel
   attribute and it also advertises IPv4 and IPv6 prefixes, the ingress
   router may have local policy defined to choose L2TPv3 for IPv4
   prefixes (provided the protocol type received in the tunnel attribute
   matches) and GRE for IPv6 prefixes.

   Additionally, the Encapsulation SAFI UPDATE message can contain a
   community or extended-community as a way to color the corresponding
   tunnel TLV(s). The same community or extended community can then be
   attached to the UPDATE messages that contain payload prefixes. This
   way, the BGP speaker can express the fact that it expects the packets
   corresponding to these payload prefixes to be received with a
   particular tunnel encapsulation header.

   In a multi-vendor deployment that has routers supporting different
   tunneling technologies, attaching community and/or extended-community
   to the Encapsulation SAFI UPDATE message can serve as a
   classification mechanism (for example, set A of routers for GRE and
   set B of routers for L2TPv3). The ingress router can then choose the
   encapsulation data appropriately while sending packets to an egress
   router.

   These communities/extended communities, if used, will be user defined
   and configured locally on the routers.

[4.4](4.4). **BGP Encapsulation Extended Community**

   We define a BGP opaque extended community that can be attached to BGP
   UPDATE messages to indicate the encapsulation protocol to be used for
   sending packets from an ingress router to an egress router.
   Considering our example from the "Introduction" section, R2 MAY
   include this extended community specifying a particular tunnel type
   to be used in the UPDATE message that carries route Q to R1. This is
   useful if there are no explicit encapsulation information to be
   signaled using the encap SAFI for a tunneling protocol (such as GRE
   without key).

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       0x03      |       TBD      |            Reserved         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|             Reserved            |          Tunnel Type        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The value of the high-order octet of the extended type field is 0x03,
which indicates it's transitive. The value of the low-order octet of
the extended type field is TBD.

The last two octets of the value field encode a tunnel type as
defined in this document.

5. **Capability advertisement**

A BGP speaker that wishes to exchange tunnel endpoint information
must use the Multiprotocol Extensions Capability Code as defined in
[RFC4760], to advertise the corresponding (AFI, SAFI) pair.

6. **Security Considerations**

If a third party is able to modify any of the information that is
used to form encapsulation headers, or to choose a tunnel type, or to
choose a particular tunnel for a particular payload type, user data
packets may end up getting misrouted, misdelivered, and/or dropped.

7. **IANA Considerations**

This document defines a new NLRI format, called Encapsulation NLRI,
to be carried in BGP using multiprotocol extensions. It is to be
assigned its own SAFI.

This document defines a new BGP optional transitive attribute type,
called Tunnel attribute and a new opaque extended community sub-type.
These values are to be assigned by IANA.

This document introduces Tunnel TLVs and sub-TLVs. The type space for
both of these should be set up by IANA as a registry of 2-octet
tunnel types and 1-octet sub-TLV types. These should be assigned on a
first-come- first-serve basis.

8. Acknowledgements

This specification builds on prior work by Gargi Nalawade, Ruchi
Kapoor, Dan Tappan, David Ward, Scott Wainner, Simon Barber, and
Chris Metz. The current authors wish to thank all these authors for
their contribution.

The authors would like to thank John Scudder, Robert Raszuk, Keyur
Patel, Chris Metz, and Yakov Rekhter for their valuable comments and
suggestions.

9. Normative References

[RFC4271] Rekhter, Y., Li T., and Hares S.(editors), "A Border
Gateway Protocol 4 (BGP-4)," RFC 4271, January 2006.

[RFC4760] Bates et al, "Multiprotocol Extensions for BGP-4," RFC
4760, January 2007.

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

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

10. Informative References

[SOFTWIRE] Dawkins S. (editor), "Softwire Problem Statement," draft-
ietf-softwire-problem-statement-02.txt, May 2006.

[Softwires-Mesh-Frame-work] Wu, J. et al, "Softwire Mesh Framework,"
draft-ietf-softwire-mesh-framework-01.txt, June 2007.

11. Authors' Addresses

    Pradosh Mohapatra
    Cisco Systems, Inc.
    170 Tasman Drive
    San Jose, CA, 95134
    Email: pmohapat@cisco.com

Eric Rosen
Cisco Systems, Inc.
1414 Massachusetts Avenue
Boxborough, MA, 01719
E-mail: erosen@cisco.com

12. **Full Copyright Statement**

13. **Intellectual Property**

ipr@ietf.org.