      Encapsulating MPLS in IP or Generic Routing Encapsulation (GRE)

Status of This Memo

   This document specifies an Internet standards track protocol for the
   Internet community, and requests discussion and suggestions for
   improvements. Please refer to the current edition of the "Internet
   Official Protocol Standards" (STD 1) for the standardization state
   and status of this protocol. Distribution of this memo is unlimited.

Copyright Notice

Abstract

   Various applications of MPLS make use of label stacks with multiple
   entries. In some cases, it is possible to replace the top label of
   the stack with an IP-based encapsulation, thereby enabling the
   application to run over networks that do not have MPLS enabled in
   their core routers. This document specifies two IP-based
   encapsulations: MPLS-in-IP and MPLS-in-GRE (Generic Routing
   Encapsulation). Each of these is applicable in some circumstances.

Table of Contents

1. Motivation

    In many applications of MPLS, packets traversing an MPLS backbone
    carry label stacks with more than one label. As described in section
    3.15 of [RFC3031], each label represents a Label Switched Path (LSP).
    For each LSP, there is a Label Switching Router (LSR) that is the
    "LSP Ingress", and an LSR that is the "LSP Egress". If LSRs A and B
    are the Ingress and Egress, respectively, of the LSP corresponding to
    a packet's top label, then A and B are adjacent LSRs on the LSP
    corresponding to the packet's second label (i.e., the label
    immediately beneath the top label).

    The purpose (or one of the purposes) of the top label is to get the
    packet delivered from A to B, so that B can further process the
    packet based on the second label. In this sense, the top label
    serves as an encapsulation header for the rest of the packet. In
    some cases, other sorts of encapsulation headers can replace the top
    label without loss of functionality. For example, an IP header or a
    Generic Routing Encapsulation (GRE) header could replace the top
    label. As the encapsulated packet would still be an MPLS packet, the
    result is an MPLS-in-IP or MPLS-in-GRE encapsulation.

    With these encapsulations, it is possible for two LSRs that are
    adjacent on an LSP to be separated by an IP network, even if that IP
    network does not provide MPLS.

To use either of these encapsulations, the encapsulating LSR must
know

-   the IP address of the decapsulating LSR, and

-   that the decapsulating LSR actually supports the particular
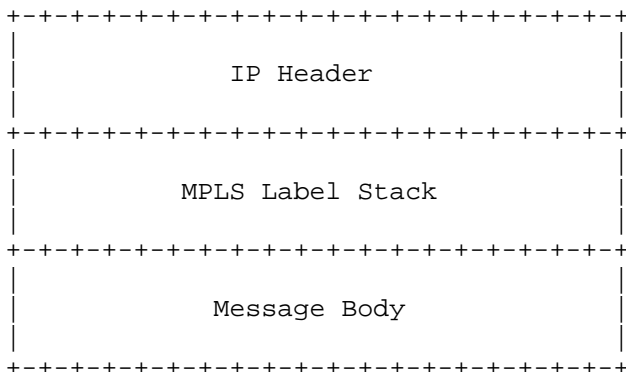   encapsulation.

This knowledge may be conveyed to the encapsulating LSR by manual
configuration, or by means of some discovery protocol. In
particular, if the tunnel is being used to support a particular
application and that application has a setup or discovery protocol,
then the application's protocol may convey this knowledge. The means
of conveying this knowledge is outside the scope of the this
document.

2. Specification of Requirements

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

3. Encapsulation in IP

MPLS-in-IP messages have the following format:

```
        +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
        |                                 |
        |           IP Header             |
        |                                 |
        +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
        |                                 |
        |        MPLS Label Stack         |
        |                                 |
        +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
        |                                 |
        |          Message Body           |
        |                                 |
        +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

        IP Header
            This field contains an IPv4 or an IPv6 datagram header
            as defined in [RFC791] or [RFC2460], respectively. The
            source and destination addresses are set to addresses
            of the encapsulating and decapsulating LSRs, respectively.

      MPLS Label Stack
           This field contains an MPLS Label Stack as defined in
           [RFC3032].

      Message Body
           This field contains one MPLS message body.

   The IPv4 Protocol Number field or the IPv6 Next Header field is set
   to 137, indicating an MPLS unicast packet. (The use of the MPLS-in-
   IP encapsulation for MPLS multicast packets is not supported by this
   specification.)

   Following the IP header is an MPLS packet, as specified in [RFC3032].
   This encapsulation causes MPLS packets to be sent through "IP
   tunnels". When the tunnel's receive endpoint receives a packet, it
   decapsulates the MPLS packet by removing the IP header. The packet
   is then processed as a received MPLS packet whose "incoming label"
   [RFC3031] is the topmost label of the decapsulated packet.

4. Encapsulation in GRE

   The MPLS-in-GRE encapsulation encapsulates an MPLS packet in GRE
   [RFC2784]. The packet then consists of an IP header (either IPv4 or
   IPv6), followed by a GRE header, followed by an MPLS label stack as
   specified in [RFC3032]. The protocol type field in the GRE header
   MUST be set to the Ethertype value for MPLS Unicast (0x8847) or
   Multicast (0x8848).

   This encapsulation causes MPLS packets to be sent through "GRE
   tunnels". When the tunnel's receive endpoint receives a packet, it
   decapsulates the MPLS packet by removing the IP and the GRE headers.
   The packet is then processed as a received MPLS packet whose
   "incoming label" [RFC3031] is the topmost label of the decapsulated
   packet.

   [RFC2784] specifies an optional GRE checksum, and [RFC2890] specifies
   optional GRE key and sequence number fields. These optional fields
   are not very useful for the MPLS-in-GRE encapsulation. The sequence
   number and checksum fields are not needed, as there are no
   corresponding fields in the native MPLS packets being tunneled. The
   GRE key field is not needed for demultiplexing, as the top MPLS label
   of the encapsulated packet is used for that purpose. The GRE key
   field is sometimes considered a security feature, functioning as a
   32-bit cleartext password, but this is an extremely weak form of
   security. In order (a) to facilitate high-speed implementations of
   the encapsulation/decapsulation procedures and (b) to ensure
   interoperability, we require that all implementations be able to
   operate correctly without these optional fields.

More precisely, an implementation of an MPLS-in-GRE decapsulator MUST
be able to process packets correctly without these optional fields.
It MAY be able to process packets correctly with these optional
fields.

An implementation of an MPLS-in-GRE encapsulator MUST be able to
generate packets without these optional fields. It MAY have the
capability to generate packets with these fields, but the default
state MUST be that packets are generated without these fields. The
encapsulator MUST NOT include any of these optional fields unless it
is known that the decapsulator can process them correctly. Methods
for conveying this knowledge are outside the scope of this
specification.

5. Common Procedures

Certain procedures are common to both the MPLS-in-IP and the MPLS-
in-GRE encapsulations. In the following, the encapsulator, whose
address appears in the IP source address field of the encapsulating
IP header, is known as the "tunnel head". The decapsulator, whose
address appears in the IP destination address field of the
decapsulating IP header, is known as the "tunnel tail".

If IPv6 is being used (for either MPLS-in-IPv6 or MPLS-in-GRE-in-
IPv6), the procedures of [RFC2473] are generally applicable.

5.1. Preventing Fragmentation and Reassembly

If an MPLS-in-IP or MPLS-in-GRE packet were fragmented (due to
"ordinary" IP fragmentation), the tunnel tail would have to
reassemble it before the contained MPLS packet could be decapsulated.
When the tunnel tail is a router, this is likely to be undesirable;
the tunnel tail may not have the ability or the resources to perform
reassembly at the necessary level of performance.

Whether fragmentation of the tunneled packets is allowed MUST be
configurable at the tunnel head. The default value MUST be that
packets are not fragmented. The default value would only be changed
if it were known that the tunnel tail could perform the reassembly
function adequately.

THE PROCEDURES SPECIFIED IN THE REMAINDER OF THIS SECTION ONLY APPLY
IF PACKETS ARE NOT TO BE FRAGMENTED.

Obviously, if packets are not to be fragmented, the tunnel head MUST
NOT fragment a packet before encapsulating it.

If IPv4 is used, then the tunnel MUST set the DF bit. This prevents
intermediate nodes in the tunnel from performing fragmentation. (If
IPv6 is used, intermediate nodes do not perform fragmentation in any
event.)

The tunnel head SHOULD perform Path MTU Discovery ([RFC1191] for
IPv4, or [RFC1981] for IPv6).

The tunnel head MUST maintain a "Tunnel MTU" for each tunnel; this is
the minimum of (a) an administratively configured value, and, if
known, (b) the discovered Path MTU value minus the encapsulation
overhead.

If the tunnel head receives, for encapsulation, an MPLS packet whose
size exceeds the Tunnel MTU, that packet MUST be discarded. However,
silently dropping such packets may cause significant operational
problems; the originator of the packets will notice that his data is
not getting through, but he may not realize that large packets are
causing packet loss. He may therefore continue sending packets that
are discarded. Path MTU discovery can help (if the tunnel head sends
back ICMP errors), but frequently there is insufficient information
available at the tunnel head to identify the originating sender
properly. To minimize problems, it is advised that MTUs be
engineered to be large enough in practice to avoid fragmentation.

In some cases, the tunnel head receives, for encapsulation, an IP
packet, which it first encapsulates in MPLS and then encapsulates in
MPLS-in-IP or MPLS-in-GRE. If the source of the IP packet is
reachable from the tunnel head, and if the result of encapsulating
the packet in MPLS would be a packet whose size exceeds the Tunnel
MTU, then the value that the tunnel head SHOULD use for fragmentation
and PMTU discovery outside the tunnel is the Tunnel MTU value minus
the size of the MPLS encapsulation. (That is, the Tunnel MTU value
minus the size of the MPLS encapsulation is the MTU that is to be
reported in ICMP messages.) The packet will have to be discarded,
but the tunnel head should send the IP source of the discarded packet
the proper ICMP error message as specified in [RFC1191] or [RFC1981].

5.2. TTL or Hop Limit

The tunnel head MAY place the TTL from the MPLS label stack into the
TTL field of the encapsulating IPv4 header or the Hop Limit field of
the encapsulating IPv6 header. The tunnel tail MAY place the TTL
from the encapsulating IPv4 header or the Hop Limit from the
encapsulating IPv6 header into the TTL field of the MPLS header, but
only if this does not increase the TTL value in the MPLS header.

   Whether such modifications are made, and the details of how they are
   made, will depend on the configuration of the tunnel tail and the
   tunnel head.

5.3. Differentiated Services

   The procedures specified in this document enable an LSP to be sent
   through an IP or GRE tunnel. [RFC2983] details a number of
   considerations and procedures that have to be applied to support the
   Differentiated Services Architecture properly in the presence of IP-
   in-IP tunnels. These considerations and procedures also apply in the
   presence of MPLS-in-IP or MPLS-in-GRE tunnels.

   Accordingly, when a tunnel head is about to send an MPLS packet into
   an MPLS-in-IP or MPLS-in-GRE tunnel, the setting of the DS field of
   the encapsulating IPv4 or IPv6 header MAY be determined (at least
   partially) by the "Behavior Aggregate" of the MPLS packet.
   Procedures for determining the Behavior Aggregate of an MPLS packet
   are specified in [RFC3270].

   Similarly, at the tunnel tail, the DS field of the encapsulating IPv4
   or IPv6 header MAY be used to determine the Behavior Aggregate of the
   encapsulated MPLS packet. [RFC3270] specifies the relation between
   the Behavior Aggregate and the subsequent disposition of the packet.

6. Applicability

   The MPLS-in-IP encapsulation is the more efficient, and it would
   generally be regarded as preferable, other things being equal. There
   are, however, some situations in which the MPLS-in-GRE encapsulation
   may be used:

      - Two routers are "adjacent" over a GRE tunnel that exists for
         some reason that is outside the scope of this document, and
         those two routers have to send MPLS packets over that
         adjacency. As all packets sent over this adjacency must have a
         GRE encapsulation, the MPLS-in-GRE encapsulation is more
         efficient than the alternative, that would be an MPLS-in-IP
         encapsulation which is then encapsulated in GRE.

      - Implementation considerations may dictate the use of MPLS-in-
         GRE. For example, some hardware device might only be able to
         handle GRE encapsulations in its fastpath.

7. IANA Considerations

   The IANA has allocated IP Protocol Number 137 for MPLS-in-IP
   encapsulation, as described in section 3. No future IANA actions
   will be required. The MPLS-in-GRE encapsulation does not require any
   IANA action.

8. Security Considerations

   The main security problem faced when IP or GRE tunnels are used is
   the possibility that the tunnel's receive endpoint will get a packet
   that appears to be from the tunnel, but that was not actually put
   into the tunnel by the tunnel's transmit endpoint. (The specified
   encapsulations do not by themselves enable the decapsulator to
   authenticate the encapsulator.) A second problem is the possibility
   that the packet will be altered between the time it enters the tunnel
   and the time it leaves. (The specified encapsulations do not by
   themselves assure the decapsulator of the packet's integrity.) A
   third problem is the possibility that the packet's contents will be
   seen while the packet is in transit through the tunnel. (The
   specification encapsulations do not ensure privacy.) How significant
   these issues are in practice depends on the security requirements of
   the applications whose traffic is being sent through the tunnel. For
   example, lack of privacy for tunneled packets is not a significant
   issue if the applications generating the packets do not require
   privacy.

   Because of the different potential security requirements, deployment
   scenarios, and performance considerations of different applications
   using the described encapsulation mechanism, this specification
   defines IPsec support as OPTIONAL. Basic implementation requirements
   if IPsec is implemented are described in section 8.1. If IPsec is
   not implemented, additional mechanisms may have to be implemented and
   deployed. Those are discussed in section 8.2.

8.1. Securing the Tunnel with IPsec

   All of these security issues can be avoided if the MPLS-in-IP or
   MPLS-in-GRE tunnels are secured with IPsec. Implementation
   requirements defined in this section apply if IPsec is implemented.

   When IPsec is used, the tunnel head and the tunnel tail should be
   treated as the endpoints of a Security Association. For this
   purpose, a single IP address of the tunnel head will be used as the
   source IP address, and a single IP address of the tunnel tail will be
   used as the destination IP address. The means by which each node
   knows the proper address of the other is outside the scope of this
   document. If a control protocol is used to set up the tunnels (e.g.,

to inform one tunnel endpoint of the IP address of the other), the
control protocol MUST have an authentication mechanism, and this MUST
be used when the tunnel is set up. If the tunnel is set up
automatically as the result of, for example, information distributed
by BGP, then the use of BGP's MD5-based authentication mechanism is
satisfactory.

The MPLS-in-IP or MPLS-in-GRE encapsulated packets should be viewed
as originating at the tunnel head and as being destined for the
tunnel tail; IPsec transport mode SHOULD thus be used.

The IP header of the MPLS-in-IP packet becomes the outer IP header of
the resulting packet when the tunnel head uses IPsec transport mode
to secure the MPLS-in-IP packet. This is followed by an IPsec
header, followed by the MPLS label stack. The IPsec header has to
set the payload type to MPLS by using the IP protocol number
specified in section 3. If IPsec transport mode is applied on a
MPLS-in-GRE packet, the GRE header follows the IPsec header.

At the tunnel tail, IPsec outbound processing recovers the contained
MPLS-in-IP/GRE packet. The tunnel tail then strips off the
encapsulating IP/GRE header to recover the MPLS packet, which is then
forwarded according to its label stack.

Note that the tunnel tail and the tunnel head are LSP adjacencies,
which means that the topmost label of any packet sent through the
tunnel must be one that was distributed by the tunnel tail to the
tunnel head. The tunnel tail MUST know precisely which labels it has
distributed to the tunnel heads of IPsec-secured tunnels. Labels in
this set MUST NOT be distributed by the tunnel tail to any LSP
adjacencies other than those that are tunnel heads of IPsec-secured
tunnels. If an MPLS packet is received without an IPsec
encapsulation, and if its topmost label is in this set, then the
packet MUST be discarded.

An IPsec-secured MPLS-in-IP or MPLS-in-GRE tunnel MUST provide
authentication and integrity. (Note that the authentication and
integrity will apply to the entire MPLS packet, including the MPLS
label stack.) Thus, the implementation MUST support ESP will null
encryption. ESP with encryption MAY be supported if a source
requires confidentiality. If ESP is used, the tunnel tail MUST check
that the source IP address of any packet received on a given SA is
the one expected.

Key distribution may be done either manually or automatically by
means of IKE [RFC2409]. Manual keying MUST be supported. If
automatic keying is implemented, IKE in main mode with preshared keys

MUST be supported. A particular application may escalate this
requirement and request implementation of automatic keying.

Manual key distribution is much simpler, but also less scalable, than
automatic key distribution. Therefore, which method of key
distribution is appropriate for a particular tunnel has to be
carefully considered by the administrator (or pair of administrators)
responsible for the tunnel endpoints. If replay protection is
regarded as necessary for a particular tunnel, automatic key
distribution should be configured.

If the MPLS-in-IP encapsulation is being used, the selectors
associated with the SA would be the source and destination addresses
mentioned above, plus the IP protocol number specified in section 3.
If it is desired to secure multiple MPLS-in-IP tunnels between a
given pair of nodes separately, each tunnel must have unique pair of
IP addresses.

If the MPLS-in-GRE encapsulation is being used, the selectors
associated with the SA would be the source and destination addresses
mentioned above, and the IP protocol number representing GRE (47).
If it is desired to secure multiple MPLS-in-GRE tunnels between a
given pair of nodes separately, each tunnel must have unique pair of
IP addresses.

8.2. In the Absence of IPsec

If the tunnels are not secured with IPsec, then some other method
should be used to ensure that packets are decapsulated and forwarded
by the tunnel tail only if those packets were encapsulated by the
tunnel head. If the tunnel lies entirely within a single
administrative domain, address filtering at the boundaries can be
used to ensure that no packet with the IP source address of a tunnel
endpoint or with the IP destination address of a tunnel endpoint can
enter the domain from outside.

However, when the tunnel head and the tunnel tail are not in the same
administrative domain, this may become difficult, and filtering based
on the destination address can even become impossible if the packets
must traverse the public Internet.

Sometimes only source address filtering (but not destination address
filtering) is done at the boundaries of an administrative domain. If
this is the case, the filtering does not provide effective protection
at all unless the decapsulator of an MPLS-in-IP or MPLS-in-GRE
validates the IP source address of the packet. This document does
not require that the decapsulator validate the IP source address of
the tunneled packets, but it should be understood that failure to do

   so presupposes that there is effective destination-based (or a
   combination of source-based and destination-based) filtering at the
   boundaries.

9. Acknowledgements

   This specification combines prior work on encapsulating MPLS in IP,
   by Tom Worster, Paul Doolan, Yasuhiro Katsube, Tom K. Johnson, Andrew
   G. Malis, and Rick Wilder, with prior work on encapsulating MPLS in
   GRE, by Yakov Rekhter, Daniel Tappan, and Eric Rosen. The current
   authors wish to thank all these authors for their contribution.

   Many people have made valuable comments and corrections, including
   Rahul Aggarwal, Scott Bradner, Alex Conta, Mark Duffy, Francois Le
   Feucheur, Allison Mankin, Thomas Narten, Pekka Savola, and Alex
   Zinin.

10. Normative References

   [RFC791]   Postel, J., "Internet Protocol", STD 5, RFC 791, September
              1981.

   [RFC792]   Postel, J., "Internet Control Message Protocol", STD 5, RFC
              792, September 1981.

   [RFC1191] Mogul, J. and S. Deering, "Path MTU discovery", RFC 1191,
              November 1990.

   [RFC1981] McCann, J., Deering, S., and J. Mogul, "Path MTU Discovery
              for IP version 6", RFC 1981, August 1996.

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

   [RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6
              (IPv6) Specification", RFC 2460, December 1998.

   [RFC2463] Conta, A. and S. Deering, "Internet Control Message
              Protocol (ICMPv6) for the Internet Protocol Version 6
              (IPv6) Specification", RFC 2463, December 1998.

   [RFC2473] Conta, A. and S. Deering, "Generic Packet Tunneling in IPv6
              Specification", RFC 2473, December 1998.

   [RFC2784] Farinacci, D., Li, T., Hanks, S., Meyer, D., and P. Traina,
              "Generic Routing Encapsulation (GRE)", RFC 2784, March
              2000.

   [RFC3031] Rosen, E., Viswanathan, A., and R. Callon, "Multiprotocol
             Label Switching Architecture", RFC 3031, January 2001.

   [RFC3032] Rosen, E., Tappan, D., Fedorkow, G., Rekhter, Y.,
             Farinacci, D., Li, T., and A. Conta, "MPLS Label Stack
             Encoding", RFC 3032, January 2001.

11. Informative References

   [RFC2401] Kent, S. and R. Atkinson, "Security Architecture for the
             Internet Protocol", RFC 2401, November 1998.

   [RFC2402] Kent, S. and R. Atkinson, "IP Authentication Header", RFC
             2402, November 1998.

   [RFC2406] Kent, S. and R. Atkinson, "IP Encapsulating Security
             Payload (ESP)", RFC 2406, November 1998.

   [RFC2409] Harkins, D. and D. Carrel, "The Internet Key Exchange
             (IKE)", RFC 2409, November 1998.

   [RFC2475] Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z.,
             and W. Weiss, "An Architecture for Differentiated Service",
             RFC 2475, December 1998.

   [RFC2890] Dommety, G., "Key and Sequence Number Extensions to GRE",
             RFC 2890, September 2000.

   [RFC2983] Black, D., "Differentiated Services and Tunnels", RFC 2983,
             October 2000.

   [RFC3260] Grossman, D., "New Terminology and Clarifications for
             Diffserv", RFC 3260, April 2002.

   [RFC3270] Le Faucheur, F., Wu, L., Davie, B., Davari, S., Vaananen,
             P., Krishnan, R., Cheval, P., and J. Heinanen, "Multi-
             Protocol Label Switching (MPLS) Support of Differentiated
             Services", RFC 3270, May 2002.

Authors' Addresses

   Tom Worster
   Motorola, Inc.
   120 Turnpike Road
   Southborough, MA 01772

   EMail: tom.worster@motorola.com

   Yakov Rekhter
   Juniper Networks, Inc.
   1194 N. Mathilda Ave.
   Sunnyvale, CA 94089

   EMail: yakov@juniper.net

   Eric Rosen
   Cisco Systems, Inc.
   1414 Massachusetts Avenue
   Boxborough, MA 01719

   EMail: erosen@cisco.com