                    Generic Routing Encapsulation (GRE)

Status of this Memo

Abstract

   This document specifies a protocol for performing encapsulation of an
   arbitrary network layer protocol over another arbitrary network layer
   protocol.
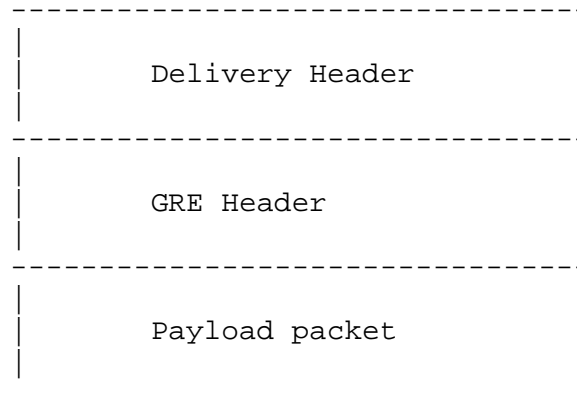
Introduction

   A number of different proposals [RFC 1234, RFC 1226] currently exist
   for the encapsulation of one protocol over another protocol. Other
   types of encapsulations [RFC 1241, SDRP, RFC 1479] have been proposed
   for transporting IP over IP for policy purposes. This memo describes
   a protocol which is very similar to, but is more general than, the
   above proposals. In attempting to be more general, many protocol
   specific nuances have been ignored. The result is that this proposal
   is may be less suitable for a situation where a specific "X over Y"
   encapsulation has been described. It is the attempt of this protocol
   to provide a simple, general purpose mechanism which is reduces the
   problem of encapsulation from its current O(n^2) problem to a more
   manageable state. This proposal also attempts to provide a
   lightweight encapsulation for use in policy based routing. This memo
   explicitly does not address the issue of when a packet should be
   encapsulated. This memo acknowledges, but does not address problems
   with mutual encapsulation [RFC 1326].

   In the most general case, a system has a packet that needs to be
   encapsulated and routed. We will call this the payload packet. The
   payload is first encapsulated in a GRE packet, which possibly also
   includes a route. The resulting GRE packet can then be encapsulated
   in some other protocol and then forwarded. We will call this outer

protocol the delivery protocol. The algorithms for processing this
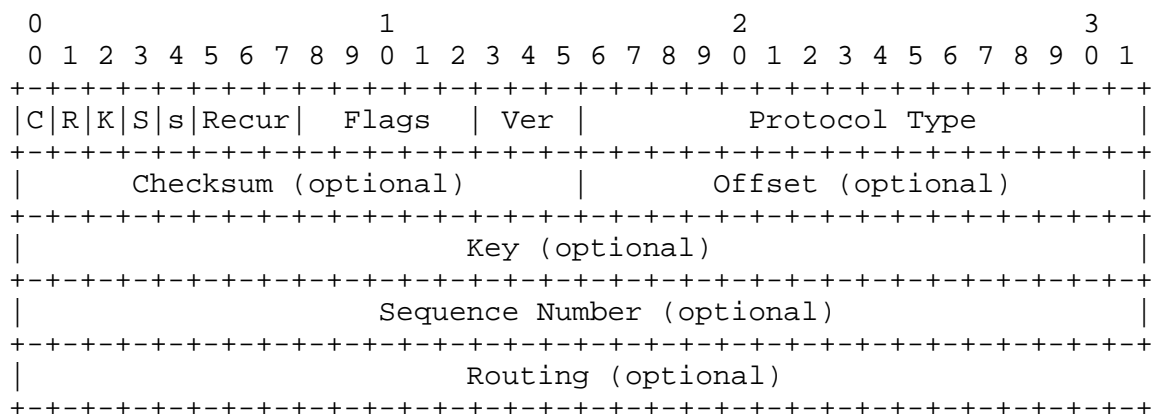packet are discussed later.

Overall packet

   The entire encapsulated packet would then have the form:

```
                 --------------------------------
                 |                              |
                 |       Delivery Header        |
                 |                              |
                 --------------------------------
                 |                              |
                 |        GRE Header            |
                 |                              |
                 --------------------------------
                 |                              |
                 |       Payload packet         |
                 |                              |
                 --------------------------------
```

Packet header

   The GRE packet header has the form:

```
       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
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |C|R|K|S|s|Recur|  Flags  | Ver |         Protocol Type         |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |      Checksum (optional)      |       Offset (optional)       |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                         Key (optional)                       |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                  Sequence Number (optional)                  |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                        Routing (optional)                    |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

   Flags and version (2 octets)

   The GRE flags are encoded in the first two octets. Bit 0 is the
   most significant bit, bit 15 is the least significant bit. Bits
   13 through 15 are reserved for the Version field. Bits 5 through
   12 are reserved for future use and MUST be transmitted as zero.

Checksum Present (bit 0)

If the Checksum Present bit is set to 1, then the Checksum field
is present and contains valid information.

If either the Checksum Present bit or the Routing Present bit are
set, BOTH the Checksum and Offset fields are present in the GRE
packet.

Routing Present (bit 1)

If the Routing Present bit is set to 1, then it indicates that the
Offset and Routing fields are present and contain valid
information.

If either the Checksum Present bit or the Routing Present bit are
set, BOTH the Checksum and Offset fields are present in the GRE
packet.

Key Present (bit 2)

If the Key Present bit is set to 1, then it indicates that the Key
field is present in the GRE header. Otherwise, the Key field is
not present in the GRE header.

Sequence Number Present (bit 3)

If the Sequence Number Present bit is set to 1, then it indicates
that the Sequence Number field is present. Otherwise, the
Sequence Number field is not present in the GRE header.

Strict Source Route (bit 4)

The meaning of the Strict Source route bit is defined in other
documents. It is recommended that this bit only be set to 1 if
all of the the Routing Information consists of Strict Source
Routes.

Recursion Control (bits 5-7)

Recursion control contains a three bit unsigned integer which
contains the number of additional encapsulations which are
permissible. This SHOULD default to zero.

Version Number (bits 13-15)

The Version Number field MUST contain the value 0. Other values
are outside of the scope of this document.

Protocol Type (2 octets)

The Protocol Type field contains the protocol type of the payload
packet. In general, the value will be the Ethernet protocol type
field for the packet. Currently defined protocol types are listed
below. Additional values may be defined in other documents.

Offset (2 octets)

The offset field indicates the octet offset from the start of the
Routing field to the first octet of the active Source Route Entry
to be examined. This field is present if the Routing Present or
the Checksum Present bit is set to 1, and contains valid
information only if the Routing Present bit is set to 1.

Checksum (2 octets)

The Checksum field contains the IP (one's complement) checksum of
the GRE header and the payload packet. This field is present if
the Routing Present or the Checksum Present bit is set to 1, and
contains valid information only if the Checksum Present bit is set
to 1.

Key (4 octets)

The Key field contains a four octet number which was inserted by
the encapsulator. It may be used by the receiver to authenticate
the source of the packet. The techniques for determining
authenticity are outside of the scope of this document. The Key
field is only present if the Key Present field is set to 1.

Sequence Number (4 octets)

The Sequence Number field contains an unsigned 32 bit integer
which is inserted by the encapsulator. It may be used by the
receiver to establish the order in which packets have been
transmitted from the encapsulator to the receiver. The exact
algorithms for the generation of the Sequence Number and the
semantics of their reception is outside of the scope of this
document.

Routing (variable)

The Routing field is optional and is present only if the Routing
Present bit is set to 1.

The Routing field is a list of Source Route Entries (SREs). Each
SRE has the form:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Address Family          |  SRE Offset  |  SRE Length  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Routing Information ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The routing field is terminated with a "NULL" SRE containing an
address family of type 0x0000 and a length of 0.

Address Family (2 octets)

The Address Family field contains a two octet value which indicates
the syntax and semantics of the Routing Information field. The
values for this field and the corresponding syntax and semantics for
Routing Information are defined in other documents.

SRE Offset (1 octet)

The SRE Offset field indicates the octet offset from the start of the
Routing Information field to the first octet of the active entry in
Source Route Entry to be examined.

SRE Length (1 octet)

The SRE Length field contains the number of octets in the SRE. If
the SRE Length is 0, this indicates this is the last SRE in the
Routing field.

Routing Information (variable)

The Routing Information field contains data which may be used in
routing this packet. The exact semantics of this field is defined in
other documents.

Forwarding of GRE packets

   Normally, a system which is forwarding delivery layer packets will
   not differentiate GRE packets from other packets in any way.
   However, a GRE packet may be received by a system. In this case, the
   system should use some delivery-specific means to determine that this
   is a GRE packet. Once this is determined, the Key, Sequence Number
   and Checksum fields if they contain valid information as indicated by
   the corresponding flags may be checked. If the Routing Present bit

is set to 1, then the Address Family field should be checked to
determine the semantics and use of the SRE Length, SRE Offset and
Routing Information fields. The exact semantics for processing a SRE
for each Address Family is defined in other documents.

Once all SREs have been processed, then the source route is complete,
the GRE header should be removed, the payload's TTL MUST be
decremented (if one exists) and the payload packet should be
forwarded as a normal packet. The exact forwarding method depends on
the Protocol Type field.

Current List of Protocol Types

   The following are currently assigned protocol types for GRE. Future
   protocol types must be taken from DIX ethernet encoding. For
   historical reasons, a number of other values have been used for some
   protocols. The following table of values MUST be used to identify
   the following protocols:

```
        Protocol Family                  PTYPE
        ---------------                  -----
        Reserved                         0000
        SNA                              0004
        OSI network layer                00FE
        PUP                              0200
        XNS                              0600
        IP                               0800
        Chaos                            0804
        RFC 826 ARP                      0806
        Frame Relay ARP                  0808
        VINES                            0BAD
        VINES Echo                       0BAE
        VINES Loopback                   0BAF
        DECnet (Phase IV)                6003
        Transparent Ethernet Bridging    6558
        Raw Frame Relay                  6559
        Apollo Domain                    8019
        Ethertalk (Appletalk)            809B
        Novell IPX                       8137
        RFC 1144 TCP/IP compression      876B
        IP Autonomous Systems            876C
        Secure Data                      876D
        Reserved                         FFFF
```

   See the IANA list of Ether Types for the complete list of these
   values.

   URL = ftp://ftp.isi.edu/in-notes/iana/assignments/ethernet-numbers.

References

RFC 1479
    Steenstrup, M. "Inter-Domain Policy Routing Protocol
    Specification: Version 1", RFC1479, BBN Systems and Technologies,
    July 1993.

RFC 1226
    Kantor, B. "Internet Protocol Encapsulation of AX.25 Frames", RFC
    1226, University of California, San Diego, May 1991.

RFC 1234
    Provan, D. "Tunneling IPX Traffic through IP Networks", RFC 1234,
    Novell, Inc., June 1991.

RFC 1241
    Woodburn, R., and D. Mills, "Scheme for an Internet Encapsulation
    Protocol: Version 1", RFC 1241, SAIC, University of Delaware, July
    1991.

RFC 1326
    Tsuchiya, P., "Mutual Encapsulation Considered Dangerous", RFC
    1326, Bellcore, May 1992.

SDRP
    Estrin, D., Li, T., and Y. Rekhter, "Source Demand Routing
    Protocol Specification (Version 1)", Work in Progress.

RFC 1702
    Hanks, S., Li, T., Farinacci, D., and P. Traina, "Generic Routing
    Encapsulation over IPv4 networks", RFC 1702, NetSmiths, Ltd.,
    cisco Systems, October 1994.

Security Considerations

    Security issues are not discussed in this memo.

Acknowledgements

   The authors would like to acknowledge Yakov Rekhter (IBM) and Deborah
   Estrin (USC) for their advice, encouragement and insightful comments.

Authors' Addresses

   Stan Hanks
   NetSmiths, Ltd.
   2025 Lincoln Highway
   Edison NJ, 08817

   EMail: stan@netsmiths.com

   Tony Li
   cisco Systems, Inc.
   1525 O'Brien Drive
   Menlo Park, CA 94025

   EMail: tli@cisco.com

   Dino Farinacci
   cisco Systems, Inc.
   1525 O'Brien Drive
   Menlo Park, CA 94025

   EMail: dino@cisco.com

   Paul Traina
   cisco Systems, Inc.
   1525 O'Brien Drive
   Menlo Park, CA 94025

   EMail: pst@cisco.com