Network Working Group
Internet Draft

Dino Farinacci
Tony Li
Procket Networks
Stan Hanks
Enron Communications
David Meyer
Cisco Systems
Paul Traina
Juniper Networks

Category
draft-meyer-gre-update-02.txt

Standards Track
January, 2000

# Generic Routing Encapsulation (GRE)
# <draft-meyer-gre-update-02.txt>

## 1. Status of this Memo

## 2. Abstract

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

## 3. Copyright Notice



## 4. Introduction

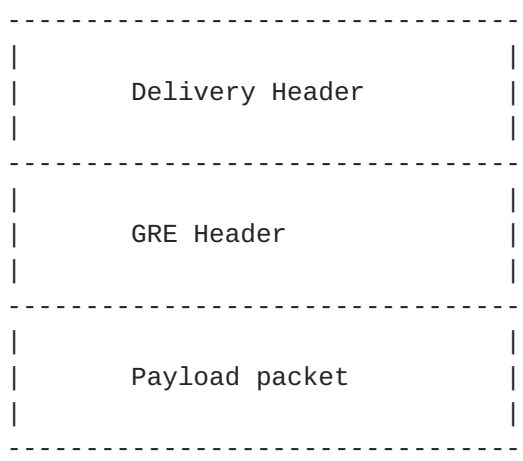
A number of different proposals [RFC1234, RFC1226] currently exist for the encapsulation of one protocol over another protocol. Other types of encapsulations [RFC1241, RFC1479] have been proposed for transporting IP over IP for policy purposes. This memo describes a protocol which is very similar to, 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 may be less suitable for a situation where a specific "X over Y" problem of encapsulation from its current O(n^2) problem to a more manageable state. This memo purposely does not address the issue of when a packet should be encapsulated. This memo acknowledges, but does not address problems such as mutual encapsulation [RFC1326].

In the most general case, a system has a packet that needs to be encapsulated and delivered to some destination. We will call this the payload packet. The payload is first encapsulated in a GRE packet. 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.

The keywords MUST, MUST NOT, MAY, OPTIONAL, REQUIRED, RECOMMENDED, SHALL, SHALL NOT, SHOULD, SHOULD NOT are to be interpreted as defined in RFC 2119 [RFC2119].

## 5. Structure of a GRE Encapsulated Packet

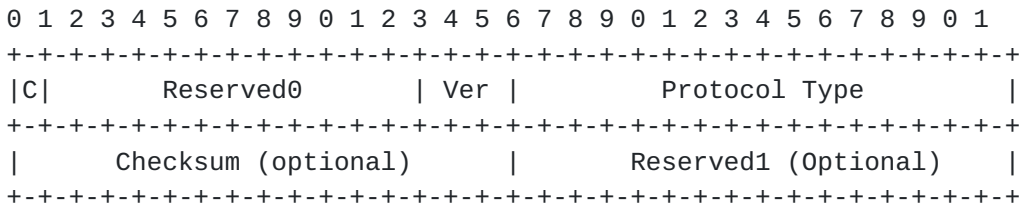
A GRE encapsulated packet has the form:




This specification is generally concerned with the structure of the GRE header, although special consideration is given to some of the issues surrounding IPv4 payloads.

### 5.1. GRE Header

The GRE packet header has the form:




### 5.2. Checksum Present (bit 0)

If the Checksum Present bit is set to one, then the Checksum and the Reserved1 fields are present and the Checksum field contains valid information. Note that a compliant implementation MUST accept and process this field.

### 5.3. Reserved0 (bits 1-12)

Bits 1 through 12 are reserved for future use. A sender MUST set them to zero while a recipient MUST be prepared to deal with non-zero data as specified in section 7.

#### 5.3.1. Version Number (bits 13-15)

The Version Number field MUST contain the value zero.

### 5.4. Protocol Type (2 octets)

The Protocol Type field contains the protocol type of the payload packet. These Protocol Types are defined in [RFC1700] as "ETHER TYPES" and in [ETYPES]. An implementation receiving a packet containing a Protocol Type which is not listed in [RFC1700] or [ETYPES] SHOULD discard the packet.

### 5.5. Checksum (2 octets)

The Checksum field contains the IP (one's complement) checksum sum of the all the 16 bit words in the GRE header and the payload packet. For purposes of computing the checksum, the value of the checksum field is zero. This field is present only if the Checksum Present bit is set to one.

### 5.6. Reserved1 (2 octets)

The Reserved1 field is reserved for future use, and if present, MUST be transmitted as zero. The Reserved1 field is present only when the Checksum field is present (that is, Checksum Present bit is set to one).

## 6. IPv4 as a Payload

When IPv4 is being carried as the GRE payload, the Protocol Type field MUST be set to 0x800.

### 6.1. Forwarding Decapsulated IPv4 Payload Packets

When a tunnel endpoint decapsulates a GRE packet which has an IPv4 packet as the payload, the destination address in the IPv4 payload packet header MUST be used to forward the packet and the TTL of the payload packet MUST be decremented. Care should be taken when forwarding such a packet, since if the destination address of the payload packet is the encapsulator of the packet (i.e., the other end of the tunnel), looping can occur. In this case, the packet MUST be discarded.

## 7. IPv4 as a Delivery Protocol

The IPv4 protocol 47 [RFC1700] is used when GRE packets are enapsulated in IPv4. See [RFC1122] for requirements relating to the delivery of packets over IPv4 networks.

## 8. Interoperation with RFC 1701 Compliant Implementations

In RFC 1701, the field described here as Reserved0 contained a number of flag bits which this specification deprecates. In particular, the Routing Present, Key Present, Sequence Number Present, and Strict Source Route bits have been deprecated, along with the Recursion Control field. As a result, the GRE header will never contain the Key, Sequence Number or Routing fields specified in RFC 1701.

There are, however, existing implementations of the RFC 1701. The following sections describe correct interoperation with such implementations.

### 8.1. RFC 1701 Compliant Receiver

An implementation complying to this specification will transmit the Reserved0 field set to zero. An RFC 1701 compliant receiver will interpret this as having the Routing Present, Key Present, Sequence Number Present, and Strict Source Route bits set to zero, and will not expect the RFC 1701 Key, Sequence Number or Routing fields to be present.

### 8.2. RFC 1701 Compliant Transmitter

An RFC 1701 transmitter may set any of the Routing Present, Key Present, Sequence Number Present, and Strict Source Route bits set to one, and thus may transmit the RFC 1701 Key, Sequence Number or Routing fields in the GRE header. In this case, an implementation compliant with this specification MAY discard the packet.

## 9. Security Considerations

Security in a network using GRE should be relatively similar to security in a normal IPv4 network, as routing using GRE follows the same routing that IPv4 uses natively. Route filtering will remain unchanged. However packet filtering requires either that a firewall look inside the GRE packet or that the filtering is done on the GRE tunnel endpoints. In those environments in which this is considered to be a security issue it may be desirable to terminate the tunnel at the firewall.

## 10. IANA Considerations for Assignment of Protocol Types

New ETHER TYPES as assigned by Xerox Systems Institute [RFC1700]. The IANA SHOULD NOT encourage the assignment of additional ETHER TYPES (GRE Protocol Types) for use with GRE.

## 11. Acknowledgments

This document is derived from the original ideas of the authors of RFC 1701 and RFC 1702. Hitoshi Asaeda, Scott Bradner, Randy Bush, Brian Carpenter, Bill Fenner, Andy Malis, Thomas Narten, and Dave Thaler and provided many constructive and insightful comments.

## 12. Appendix -- Known Issues

This document specifies the behavior of currently deployed GRE implementations. As such, it does not attempt to address the following known issues:

### 12.1. Interaction Path MTU Discovery (PMTU) [RFC1191]

Existing implementations of GRE, when using IPv4 as the Delivery Header, do not implement Path MTU discovery and do not set the Don't Fragment bit in the Delivery Header. This can cause large packets to become fragmented within the tunnel and reassembled at the tunnel exit (independent of whether the payload packet is using PMTU). If a tunnel entry point were to use Path MTU discovery, however, that tunnel entry point would also need to relay ICMP unreachable error messages (in particular the "fragmentation needed and DF set" code) back to the originator of the packet, which is not a requirement in this specification. Failure to properly relay Path MTU information to an originator can result in the following behavior: the originator sets the don't fragment bit, the packet gets dropped within the tunnel, but since the originator doesn't receive proper feedback, it retransmits with the same PMTU, causing subsequently transmitted packets to be dropped.

### 12.2. IPv6 as Delivery and/or Payload Protocol

This specification describes the intersection of GRE currently deployed by multiple vendors. IPv6 as delivery and/or payload protocol is not included in the currently deployed versions of GRE.

### 12.3. Interaction with ICMP

### 12.4. Interaction with the Differentiated Services Architecture

### 12.5. Multiple and Looping Encapsulations

## 13. REFERENCES

[ETYPES] ftp://ftp.isi.edu/in-notes/iana/assignments/ethernet-numbers

[RFC1122] R.T. Braden, "Requirements for Internet hosts - communication layers", RFC1122, Octber 1989

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

[RFC1226] Kantor, B. "Internet Protocol Encapsulation of AX.25 Frames", RFC 1226, University of California, San Diego, May 1991.

[RFC1234] Provan, D. "Tunneling IPX Traffic through IP Networks", RFC 1234, Novell, Inc., June 1991.

[RFC1241] Woodburn, R., and D. Mills, "Scheme for an Internet Encapsulation Protocol: Version 1", RFC 1241, SAIC, University of Delaware, July 1991.

[RFC1326] Tsuchiya, P., "Mutual Encapsulation Considered Dangerous", RFC 1326, Bellcore, May 1992.

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

[RFC1700] J. Reynolds and J. Postel, "Assigned Numbers", RFC 1700, October 1994.

[RFC1701] Hanks, S., Li, T, Farinacci, D., and P. Traina, "Generic Routing Encapsulation", RFC 1701, NetSmiths, Ltd., and cisco Systems, October 1994.

[RFC1702] Hanks, S., Li, T., Farinacci, D., and P. Traina, "Generic Routing Encapsulation over IPv4 networks", RFC 1702, NetSmiths, Ltd., cisco Systems, October 1994.

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

[RFC2408] Maughan, D., Schertler, M., Schneider, M., and J. Turner, "Internet Security Association and Key Management Protocol (ISAKMP)", RFC 2408, November 1998.

## 14. Authors' Addresses

Dino Farinacci
Procket Networks
3850 No. First St., Ste. C
San Jose, CA 95134
Email: dino@procket.com

Tony Li
Procket Networks
3850 No. First St., Ste. C
San Jose, CA 95134
+1 408 954 7903 (w)
+1 408 987 6166 (f)
Email: tony1@home.net

Stan Hanks
Enron Communications
Email: stan_hanks@enron.net

David Meyer
Cisco Systems, Inc.
170 Tasman Drive
San Jose, CA, 95134
Email: dmm@cisco.com

Paul Traina
Juniper Networks
Email: pst@juniper.net