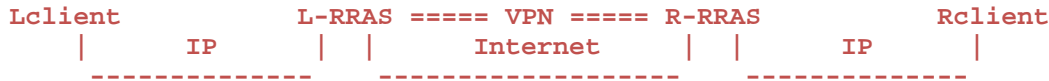


# VPN Tunnels

## GRE Protocol 47 Packet Description and Use (Microsoft Corporation)

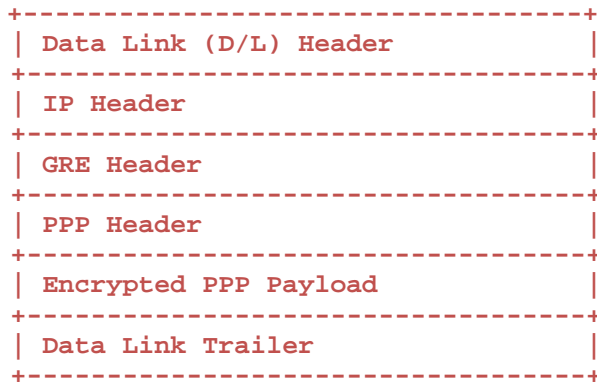
The Generic Route Encapsulation (GRE) protocol is used in conjunction with Point-to-Point Tunneling Protocol (PPTP) to create virtual private networks (VPNs) between clients or between clients and servers. One popular implementation is to use Microsoft's VPN technology between two Routing and Remote Access Services (RRAS) servers that are configured for LAN-to-LAN routing, as shown below:



To better understand the use of GRE in the creation and use of VPNs, it is helpful to understand the packet structure. After the PPTP control session has been established, GRE is used to encapsulate the data or payload in a secure manner.

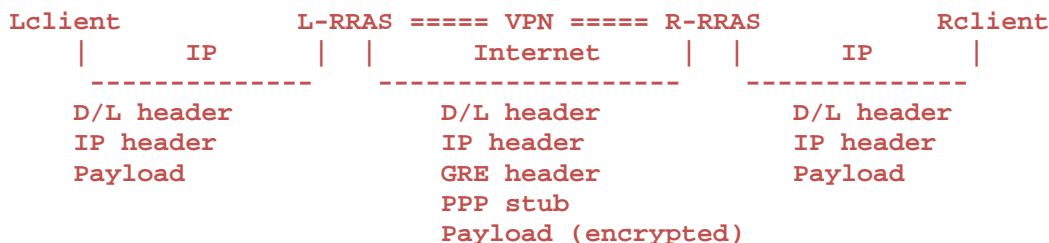
### VPN Tunnels - PPTP Protocol Packet Description and Use

The GRE packet format that Microsoft uses for encapsulating data has the following general form:



The data or payload that is going to pass through the tunnel is given a Point-to-Point Protocol (PPP) header and then placed inside a GRE packet. The GRE packet carries the data between the two tunnel endpoints. After the GRE packet has arrived at the final destination (the endpoint of the tunnel), it is discarded and the encapsulated packet is then transmitted to its final destination.

Using the diagram at the top of this section, an Internet Protocol (IP) packet from Lclient is first transmitted to the L-RRAS server. The IP packet is encrypted, given an additional PPP header, and then placed inside a GRE packet. The diagram below says "PPP stub" and not "PPP header" because the PPP header is also encrypted along with the data. Although it cannot see it, the GRE protocol is configured to know that a PPP header exists. The GRE packet with the encapsulated and encrypted data is sent across the Internet with a final destination of "R-RRAS server." The R-RRAS server strips off the GRE header and PPP header, and then transmits the decrypted data (IP packet) to Rclient.

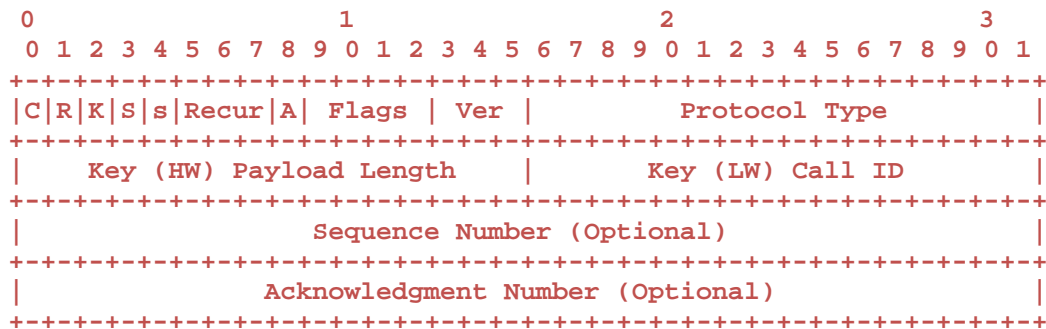


# VPN Tunnels

## GRE Protocol 47 Packet Description and Use (Microsoft Corporation)

### The Protocol Header

To understand how the GRE protocol works as an encapsulating protocol, you need to review the header format of the protocol. The GRE packet header as implemented by Microsoft has the following form:



The following table lists each field with a more detailed explanation of its function and the parameters that can be used.

C	(Bit 0) Checksum Present. Set to zero (0).
R	(Bit 1) Routing Present. Set to zero (0).
K	(Bit 2) Key Present. Set to one (1).
S	(Bit 3) Sequence Number Present. Set to one (1) if a payload (data) packet is present. Set to zero (0) if payload is not present (GRE packet is an Acknowledgment only).
s	(Bit 4) Strict source route present. Set to zero (0).
Recur	(Bits 5-7) Recursion control. Set to zero (0).
A	(Bit 8) Acknowledgment sequence number present. Set to one (1) if packet contains Acknowledgment Number to be used for acknowledging previously transmitted data.
Flags	(Bits 9-12) Must be set to zero (0).
Ver	(Bits 13-15) Must contain 1 (enhanced GRE).
=====	
Protocol Type	Set to hex 880B (for PPP).
Key (HW) Payload Length	(High 2 octets of Key) Size of the payload, not including the GRE header.
Key (LW) Call ID	(Low 2 octets) Contains the Peer's Call ID for the session to which this packet belongs.
Sequence Number	Contains the sequence number of the payload. Present if S bit (Bit 3) is one (1).

# VPN Tunnels

## GRE Protocol 47 Packet Description and Use

(Microsoft Corporation)

Acknowledgment Number	Contains the sequence number of the highest numbered GRE packet received by the sending peer for this user session. Present if A bit (Bit 8) is one (1).
-----------------------	--

### Enhancements

The GRE protocol has several noteworthy enhancements. These are from Request for Comments (RFC) 2637.

- An Acknowledgment Number field. This is used to determine whether a particular GRE packet or set of packets has arrived at the remote end of the tunnel. This acknowledgment capability is not used in conjunction with any retransmission of user data packets. It is used instead to determine the rate at which user data packets are to be transmitted over the tunnel for a given user session.
- Tunneling portability. The payload section contains a PPP data packet without any media-specific framing elements.
- Sequence number tracking. The sequence numbers involved are per-packet sequence numbers. The sequence number for each user session is set to zero at session startup. Each packet sent for a given user session that contains a payload (and has the S bit, or Bit 3, set to one) is assigned the next consecutive sequence number for that session.
- Use of piggyback Acks. This protocol allows acknowledgments to be carried with the data and makes the overall protocol more efficient, which in turn requires less buffering of packets.

### Network Monitor Traces

You should note several things when you are looking at a Network Monitor trace. The flags summary is made up of the hexadecimal value of the first 16 bits. In the sample packet below, the flags summary is 12,417 or 0x3081h. The Microsoft Network Monitor parser does not represent the version number in the Flags Summary bit field, but it is there. For example, assume the following sample packet:

```
GRE: Flags Summary = 12417 (0x3081)
    GRE: 0..... = Checksum Absent
    GRE: .0..... = Routing Absent
    GRE: ..1..... = Key Present
    GRE: ...1..... = Sequence Number Present
    GRE: ....0..... = Strict Source Route Absent
    GRE: .....1..... = Acknowledge Sequence Number Present
GRE: Recursion Control = 0 (0x0)
GRE: Ver = 1 (0x1)
GRE: Protocol Type = 0x880B
GRE: Key Length = 90 (0x5A)
GRE: Key Call ID = 32768 (0x8000)
GRE: Sequence Number = 16 (0x10)
GRE: Ack Number = 15 (0xF)
```

**VPN Tunnels**  
**GRE Protocol 47 Packet Description and Use**  
**(Microsoft Corporation)**

The first 8 bits are 00110000, which represents a hexadecimal value of 30. The next 8 bits are 10000001, which represents a hexadecimal value of 81. Therefore, the flags summary is 0x3081h, or 12,417 in decimal.