Network Working Group
INTERNET DRAFT

Elwin Stelzer
Corona Networks, Inc.

July 2001
Expires: Jan 2002

VPN Tunneling Protocol
<draft-elwin-ppvpn-vtp-00.txt>

1.0 Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of RFC2026.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet- Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at:
http://www.ietf.org/ietf/1id-abstracts.txt

The list of Internet-Draft Shadow Directories can be accessed at:
http://www.ietf.org/shadow.html.

2.0 Abstract

This draft describes a VPN Tunnel encapsulation shim header (VTE) to be used in VPN tunnels and also defines a simple tunnel signaling protocol, called VPN Tunneling Protocol (VTP). It provides a simple mechanism for carrying VPN traffic over provider backbone. The protocol is expected to be run at each of the PE routers in the provider backbone. Remote access nodes can also join the VPN using this protocol.

3.0 Table of Contents

4.0 Introduction

4.1 Terminology

This draft uses the terms defined in [PPVPN-RQ] and [PPVPN-FW], and defines the following new terms.

TVI - Tunnel Virtual Interface

VTE - VPN Tunnel Encapsulation

VTP - VPN Tunneling Protocol

4.2 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].

4.3 Comparison with other approaches

TBD

5.0 VTP Overview

VTP has two components:

- VTE (VPN Tunnel Encapsulation): header used in datapath

- VTP Signaling

5.1 VTE

The VTE encapsulation is over IP, and the IP protocol field uses the protocol-type value TBD to indicate this. This shim header is used to identify VPNs and other tunnel parameters.

This is a 16 octet header and is formatted as shown in Fig 6.1.

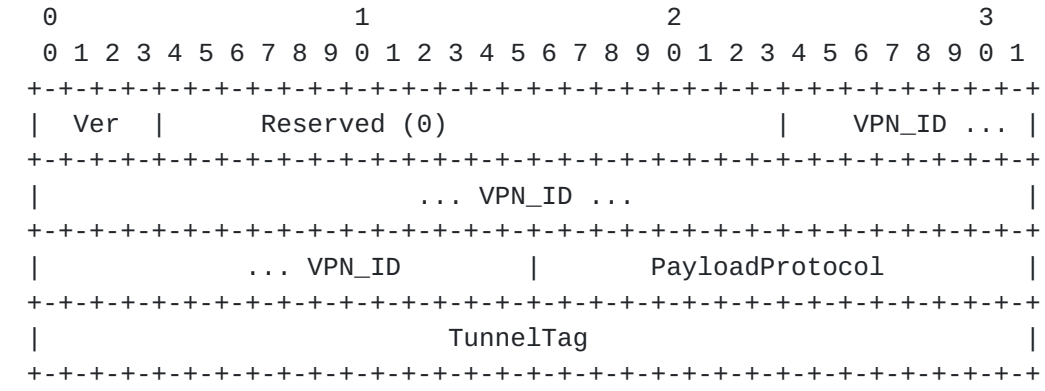

Figure 5.1 VTE Format

Where:

Ver = 0001, for this format of VRTE

Reserved = all bits 0, till more extensions are added

VPN_ID = the 7 octet VPN ID, corresponding to the VPN traffic being tunneled. Refer [RFC 2685].

PayloadProtocol = protocol corresponding to the payload. Mostly it is expected to be IP.

TunnelTag = a 4 octet value for additional tagging.

Special keepalive packets can also be sent with this encapsulation to check the data-path tunnel operational status.

5.2 VTP Signaling

A VTP session is established between each pair of participating PEs.

Each PE router, when it becomes operational, learns all the VPNs

existing in each of the PE peers. The VpnInfo message is used for this

purpose. The message is sent with the list of VPNs a PE router supports, and it also has a request flag, that denotes if the peers needs to respond with VpnInfo message. VpnInfo is also sent to other PEs when a new VPN is enabled in a PE router. This delta information is conveyed using another flag in the VpnInfo message.

Once the VPN information is obtained about the peers, the PE router initiates the establishment of VTP tunnels. With the tunnel being established, packets in the datapath are encapsulated with VTE.

The VTP runs on top of TCP, and listens to the port-number TBD. The relationship between VTP, VTE and other modules are shown in Fig 5.2.

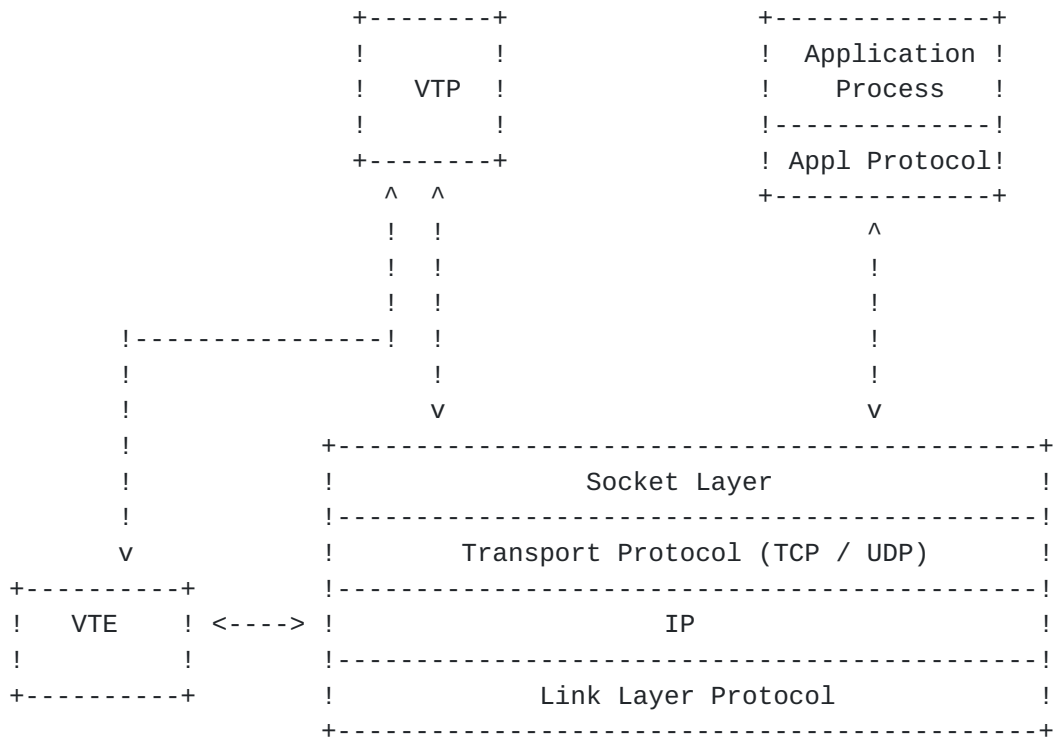

Figure 5.2 VTP Relationships

6.0 Protocol Specification

6.1 VTP protocol header format

Each VTP PDU is a VTP header followed by a VTP messages.

The VTP header is a 12 octet value and is shown in Fig 6.1.

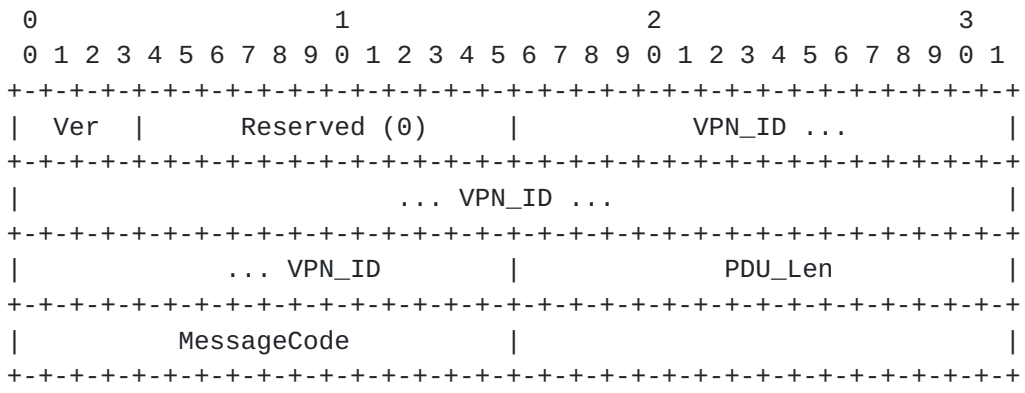

Figure 6.1 VTP Header

Where:

Ver = 0001, for this format of VTE

Reserved = all bits 0, till more extensions are added

VPN_ID = the 7 octet VPN ID, corresponding to the VPN traffic
being tunneled. Refer [RFC 2685].

PDU_Len = Two octet integer specifying the total length of this PDU
in octets starting from version field.

MessageCode = This identifies what message is carried in this PDU.

Following are the messages that are defined for VTP:

VpnInfo message (msgCode = 1)
TunStpRq message (msgCode = 11)
TunStpRp message (msgCode = 12)
TunClrRq message (msgCode = 13)
TunUpdRq message (msgCode = 14)
TunUpdRp message (msgCode = 15)

6.2 AVPs

VTP uses a Attribute-Value-Pair (AVP) encoding scheme to encode the information carried in VTP messages.

Each AVP is encoded as:

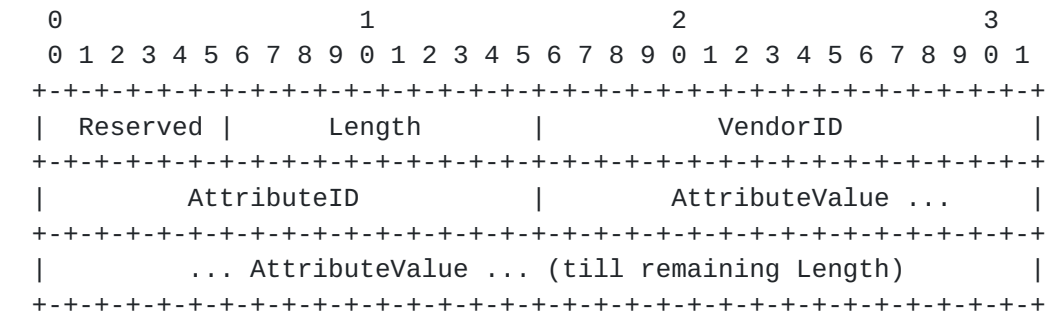

Figure 6.2 AVP Format

Where:

Reserved = all bits 0, till more extensions are added

Length = number of octets in this AVP, starting from 'reserved'.

VendorID = IANA assigned enterprise code.

AttributeID = Identifier for the Attribute

AttributeValue = Value for the Attribute

Following are the different AVPs defined for VTP:

ResultCode AVP - has code denoting errors, or reason for tunnel clearing, etc.

Capability AVP - has a capability bitmap denoting the optional features supported.

Keepalive AVP - used for controlling keepalive functions.

VpnList AVP - has a list of VPNs supported, also has additional flags

TunnelTag AVP - the tunnel tag value

ProtocolVersion AVP - protocol version supported.

6.3 VTP State Machine

The following state machine represent the state corresponding to each VPN tunnels in a device. As soon as the TCP connection is established, all the VPN tunnels start with a IDLE state.

State	Event	Action	New State
IDLE	Trigger for setting up tunnel	Tx TunStpRq	TXWAIT
IDLE	Rx TunStpRq	Tx TunStpRp estab tunnel	ESTAB
IDLE	Rx any other VRTP msg	Tx TunClrRq	IDLE
TXWAIT	Rx TunStpRp	estab tunnel	ESTAB
TXWAIT	Rx TunStpRq	TxTunStpRp estab tunnel	ESTAB
TXWAIT	Rx any other VRTP msg	Tx TunClrRq close tunnel	IDLE
ESTAB	Rx TunClrRq	close tunnel	IDLE
ESTAB	Rx TunStpRq	Tx TunClrRq close tunnel	IDLE
ESTAB	Rx TunStpRp	Discard msg	ESTAB
ESTAB	Rx any other VRTP msg	Process msg	ESTAB
ESTAB	Trigger for clearing up tunnel	Tx TunClrRq close tunnel	IDLE

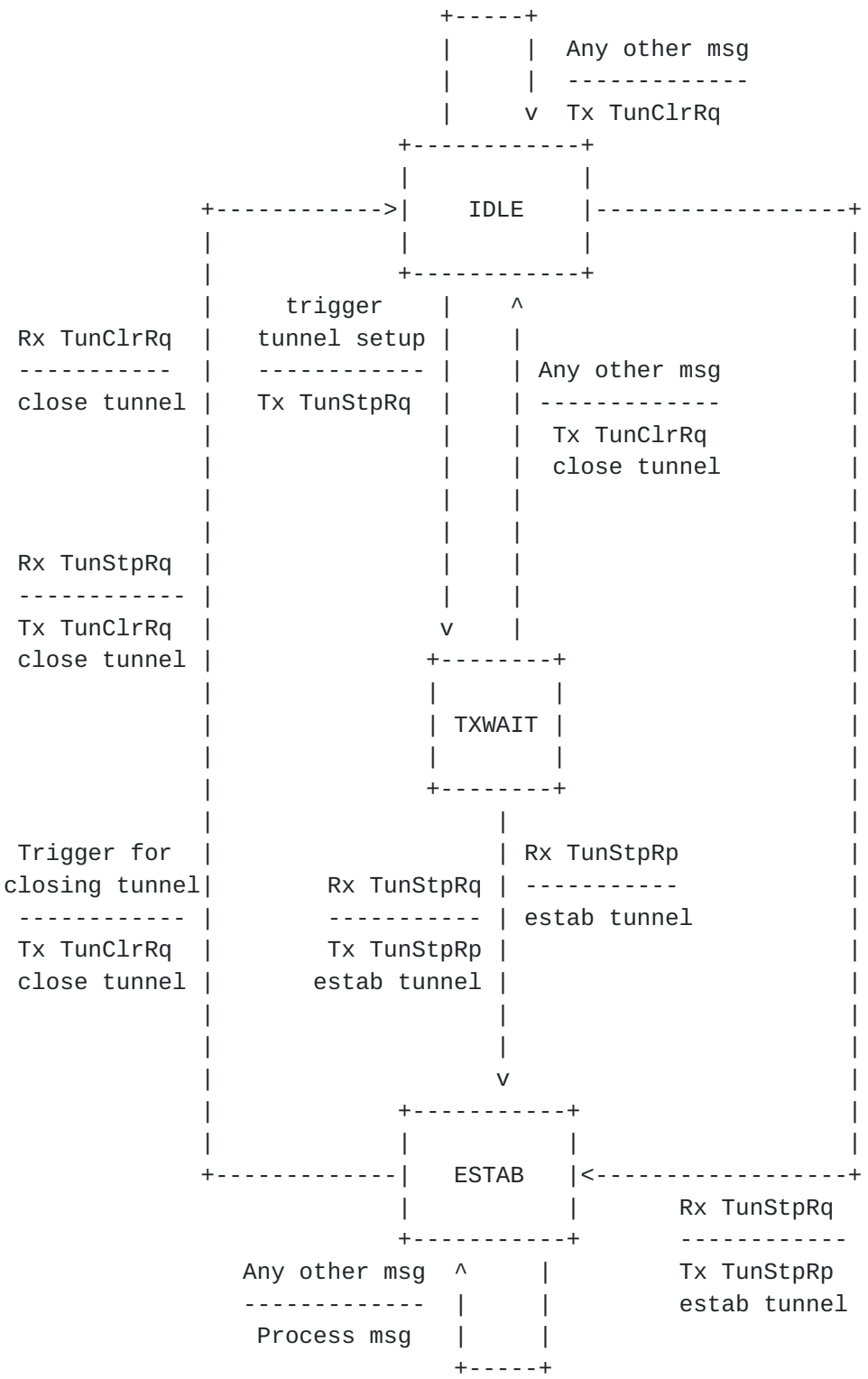
Any other msg
Tx TunClrRq
IDLE
trigger
tunnel setup
Tx TunStpRq
Rx TunClrRq
close tunnel
Any other msg
Tx TunClrRq
close tunnel
Rx TunStpRq
Tx TunClrRq
close tunnel
TXWAIT
Rx TunStpRp
estab tunnel
Trigger for
closing tunnel
Rx TunStpRq
Tx TunStpRp
estab tunnel
Tx TunClrRq
close tunnel
ESTAB
Rx TunStpRq
Tx TunStpRp
estab tunnel
Any other msg
Process msg

Figure 6.3 VTP State Machine

7.0 Protocol Operation

The protocol operation is illustrated with the sample scenario shown in Fig 7.1. This section is just used for illustration and does not constrain any variation in implementation.

Here V1, V2 and V3 are three VPNs considered, and PE1, PE2, PE3 and PE4 are four PE routers. The intermediate P routers are not shown. The lines shown represent VTP tunnels.

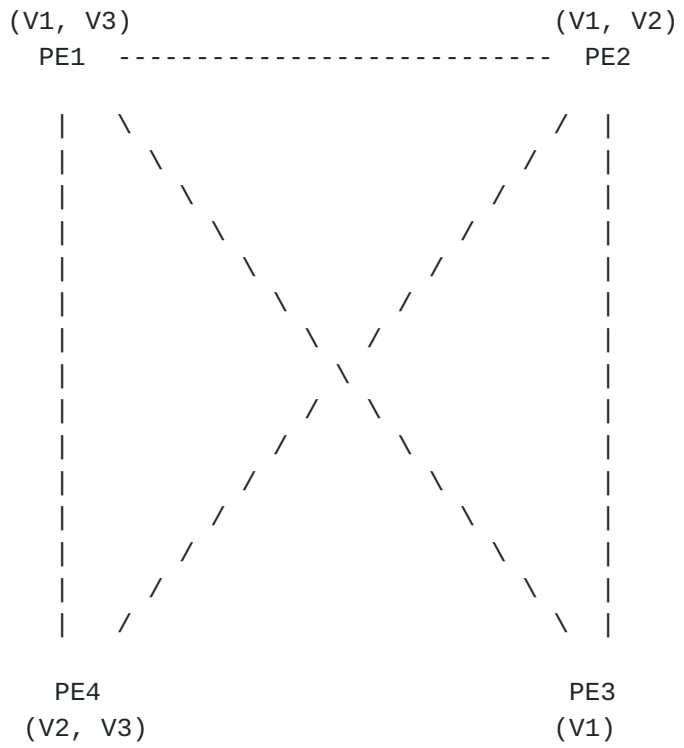

Figure 7.1 Sample Scenario

7.1 Control connection operation

- Consider a scenario in which the routers PE1, PE2 and PE3 are up and operational, and the router PE4 is started.
- PE4 understands the PE1 and PE2 are the routers with which it needs to establish a tunnel, through VTP.
- PE4 establishes these tunnels, using VTP.

7.2 Datapath Operation

- Take for example a packet in VPN V1 to reach PE3 from PE1.

- Packet is received in PE1 from a V1 CE, attached to this PE.
- Forwarding lookup done with the V1 VFI forwarding table in PE1.

- Egress interface is obtained which is a TVI.
- The TVI Tx function, encapsulates the packet with VTE and outer IP header. The source IP address is that of PE1, and destination IP address of the remote PE is determined based on TVI.
- Backbone forwarding table lookup is done with the encapsulated packet, and eventually the packet is sent out.
- The remote PE receives this packet and finds it as a local packet.
- VTE decapsulation is done, during which the appropriate VFI is determined.
- Forwarding lookup is done with this VPN forwarding table, and eventually the packet is transmitted out.

7.3 VPN Route Exchanges

- IGP routing protocols are instantiated per VFI.
- The dynamic TVIs are also added as interfaces to the appropriate routing protocol instance.
- Thus routing protocol PDUs get encapsulated in VTE and reach the remote PE, where it gets decapsulated and is handed over to the appropriate routing instance.

8.0 Multipoint Tunnels

TBD

9.0 IANA Considerations

- TCP port number for VTP to be assigned.
- IP protocol type value for VTE to be assigned.

10.0 Security Considerations

The VTP as such does not have any authentication or encryption in it. Rather it relies on other protocols, like IPSec to do this.

11.0 Summary and Conclusions

The draft defines a mechanism to tunnel VPN traffic over a backbone network. The solution described does not need MPLS or IPSec as a prerequisite to achieve this. For cases where security is needed IPSec can still be used along with this, in transport mode.

12.0 Acknowledgments

13.0 References

[RFC 2026] S. Bradner, "The Internet Standards Process - Revision 3", RFC 2026, Oct 1996.

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

[RFC 2685] B. Fox, B. Gleeson, "Virtual Private Networks Identifier", RFC 2685, Sep 1999.

[PPVPN-RQ] M. Carugi et al, "Service Requirements for Provider Provisioned Virtual Private Networks", Work in progress, draft-ietf-ppvpn-requirements-01.txt, Jun 2001

[PPVPN-FW] R. Callon et al, "A Framework for Provider Provisioned Virtual Private Networks", Work in progress, draft-ietf-ppvpn-framework-00.txt, Feb 2001.

14.0 Author's Address

Elwin Stelzer Eliazer
Corona Networks, Inc.
630 Alder Drive
Milpitas, CA 95035
Phone: 408-519-3832
Email: elwinietf@yahoo.com

APPENDIX A: Summary for Sub-IP Area

This draft defines a tunneling and signaling mechanism to achieve Provider Provisioned VPNs.

A.1 Where does it fit in the Picture of the Sub-IP Work

This work fits in the PPVPN Working Group.

A.2 Why is it Targeted at this WG

The WG is chartered with developing Provider Provisioned VPN solutions. This draft contributes to this.

A.3 Justification

The WG should consider this document since it provides a means to achieve PPVPNs.