Internet Engineering Task Force
INTERNET DRAFT
draft-macrae-policy-cops-vpn-00.txt
Expiration Date: August 1999

M. MacRae
S. Ayandeh
Nortel Networks
February 1999

Using COPS for VPN Connectivity

Status of this Memo

Abstract

This document proposes a solution for establishing connectivity for a network routed VPN. It makes use of a Policy Server to get information on the establishment of tunnels between the Edge Devices of a Service Provider's network. Use of a Policy Server allows a network operator to enable the tunnel topology which is most suited to the VPN; provides a mechanism to learn about the current state of the VPN connectivity; and enables the addition and removal of tunnels to reflect current network demands. The proposed extensions are independent of the specific tunneling technology deployed in the core of the network, i.e. MPLS, ATM, FR or IP.

The document also defines the extensions required to the COPS (Common Open Policy Service) protocol being developed in the IETF for communication between the Edge Devices and the Policy Server to exchange VPN connectivity information.

Table of Contents

1. Introduction

There are a variety of VPN types which can be supported over IP and non-IP facilities and which can operate at different layers of a protocol stack. This work focuses on what is called a Virtual Private Routed Network (VPRN), defined as an emulation of a multi-site wide area routed network using IP facilities. These network based VPNs are operated by Internet Service Providers (ISPs), and are implemented within the provider network rather than on the CPE equipment.

VPRNs operate by establishing tunnels between the Edge Devices (EDs) supporting the sites of a VPN. Currently, there is no scalable and interoperable method of dynamically setting up these tunnels. This work proposes a solution for establishing the connectivity between all sites of a routed VPN. It uses policy to determine which EDs require tunnels between them, thus allowing support of various tunnel topologies. An off-network Policy Server (PS) owns information regarding the configuration for the VPRN tunnels, and informs the ED. The proposed ED - PS communication is the COPS (Common Open Policy Service) protocol [1] which is being developed by the IETF RAP (RSVP Admission Protocol) WG. Tunnels are likely to be subjected to QOS considerations, and the use of COPS as the common mechanism for communicating both VPN and QOS related information is advantageous (see COPS extensions for DiffServ [2] and RSVP [3]).

This document starts by defining a VPRN and the tunnels used, makes a case for using policy and COPS, gives an overview of the RAP policy architecture, and highlights the COPS operation. It then gives various scenarios for establishing and updating the VPN connectivity before describing in detail the COPS extensions required for VPN connectivity support.

2. VPRNs and Tunnels

2.1 VPRN Reference Network

A VPRN, pictured in Figure 1 below, consists of a mesh of IP tunnels between the Edge Devices in an ISP network. Attached to these EDs are CPE routers connected via one or more links, termed "stub" links. The CPE could be any piece of equipment connecting a VPN site to the ISP network; it is usually a router, but it could be, for example, a LAN Switch sitting in front of a LAN. CPEs can be multihomed to the ISP network; in this case, one stub link may be designated as a "backup" link to be used in case of failure, or both links can be active. A link between 2 CPEs outside of the ISP network is termed a "backdoor" link. Although not shown in the diagram, an ED could support multiple VPNs from the same or different stub link.

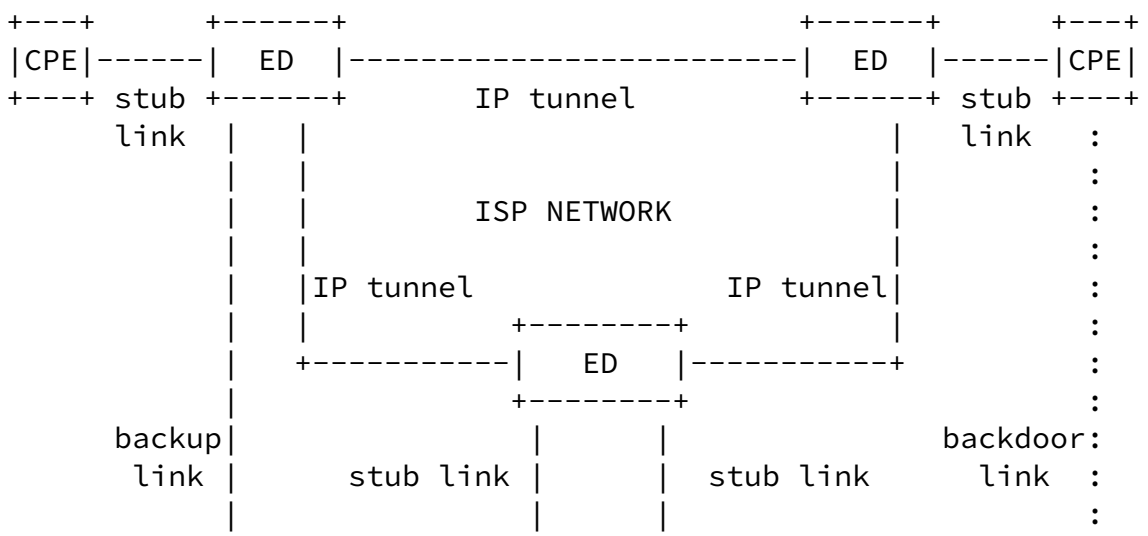


Figure 1 - Virtual Private Routed Network - Reference Network

2.2 Tunnels in a VPRN

Tunnels have entry-points, transit points, and exit points. The entry and exit points for VPRN tunnels are the EDs; the transit points (not shown in the figure) are network devices in the provider network. Packets traveling on a tunnel are encapsulated at the entry point ED and forwarded towards the designated exit point ED, where the packet is decapsulated. Transit points are unaware that a tunnel is traversing them: tunnel packets are forwarded just as any other packet. However, one might want to assign a particular QOS (e.g. guarantee some

bandwidth or give higher/lower priority) to the packets flowing through a tunnel, in which case, the transit points need to be configured appropriately. Therefore, tunnel establishment needs to be coordinated at entry, transit and exit points.

Various encapsulation methods (or tunnel technology) for VPRN tunnels exist: IP-in-IP, IPsec, GRE, L2TP, ATM, FR or MPLS. This document does not advocate the use of any particular tunnel technology, and allows for the support of all. The specific information needed by an ED to perform tunnel installation for various technologies is not the concern of this document.

Tunnels can be established in a full-mesh, partial mesh, or arbitrary topology, depending on the VPRN traffic patterns and policy considerations. Tunnels can support uni or bi-directional traffic, requiring sometimes more than one tunnel to achieve full duplex communication between EDs. For example, an ED can send and receive frames on the same Frame Relay DLCI, but it can only receive on one GRE tunnel and send on another.

2.3 Establishing Tunnels

A tunnel endpoint can be installed when an ED initializes, upon operator intervention, or possibly when a new site joins a VPRN. Tunnel installation has to be performed at both endpoints (and possibly the transit points) and should be coordinated by the PS.

The ED requires parameters to control the installation. This information may be technology specific and it may vary between tunnels of the same technology. These include, but are probably not limited to:

Rate of retrying tunnel installation:
- interval to wait after initial installation failure
- interval between each subsequent retry (for linear rate)
- factor by which to increment the previous interval to compute the current interval (for exponential rate)

- maximum interval

Reporting tunnel installation failures to PS:
- report initial failure only
- report every failure
- report each "n" failure
- never report failures

(More work is required to define / refine these parameters.)

Tunnel installation is meant to reserve the resources and setup the appropriate structures in the ED for that tunnel. However, the ED still needs to know when the tunnel can be used, for example, when both ends of an IP-in-IP tunnel are installed. Since the PS has a view of the entire tunnel, it informs the ED when the tunnel can be activated.

2.4 Removing Tunnels

Tunnels are removed upon failure of a tunnel component, operator intervention, or possibly when a site leaves a VPN. Depending on the tunnel technology used, it may not be possible for an ED to detect a tunnel failure. For example, failure of the exit point ED of an IP-in-IP tunnel cannot be directly detected at the entry point ED due to the connectionless nature of the tunnel, but routing information may convey the loss of a path to the failed ED; on the other hand, failure of an ATM VC is easily detected by the EDs at both endpoints. If a failure can be detected by an ED, it should report it to the PS which may take steps to remove the tunnel at all points.

A mechanism to dampen the reporting of flapping tunnels is needed. This requires further study.

3. Using COPS for VPN Connectivity

Policy is what controls the behaviour of a network under different conditions. Rather than offering a uniform or best-effort service to all users based on static or automated technical considerations, a policy enabled network takes into account the business priorities of the users and their applications, and dynamically determines the treatment to give each packet.

For VPRNs, policy is essential in determining the tunnels needed to establish a topology that reflects the expected traffic patterns between sites, respects the criticalness of certain paths or applications, and allows packets to be routed based on non-technical considerations (e.g. all packets from a certain VPN site must go through a firewall). No automatic tool for the discovery of EDs and tunnel establishment could take into consideration these business aspects. Policy dictates which tunnels are required, which EDs and transit points are involved in installing a tunnel, which tunnel technology to use, and any other information which is necessary to establish the tunnel.

exchanging QOS information (such as filters) between a PS (or Bandwidth Broker) and an ED, using COPS for VPN connectivity enables a PS to coordinate tunnel installation and the QOS behaviour of those tunnels.

COPS allows for dynamic updating of the connectivity between the VPN sites. The PS can oversee the end-to-end management of tunnels (provided the policy architecture supports inter-domain PS-PS communication). Changes to the tunnel topology can be done at different times of day by installing tunnels which are only valid during certain periods of the day, week, month or year.

COPS provides a feedback mechanism where the status of the tunnels from all the affected EDs for a given VPN can be correlated at the PS. (Note that the interaction between policy management and network management is out of the scope of this document.) A network operator can therefore add and remove tunnels to better reflect the current network demands.

Without COPS, tunnel information would need to be configured for each ED individually and downloaded through a network management or command line interface to each affected ED. This lacks the central control needed to establish a coherent topology and the dynamism required for monitoring and updating the topology.

4. Policy Server for VPN Connectivity

The Policy Server requirements and its implementation are not dealt with in this document. How the VPN connectivity policy information is obtained, stored, and its exact data representation is not discussed. However, there are certain assumptions being made that are worth mentioning.

The Policy Server has access to information required to determine and establish connectivity between the EDs for all the VPNs in its domain. Identifiers are used for each VPN and these may or may not have global significance. The VPN Identifiers are not yet standardized in the IETF, we assume a generic coding capable of accepting many formats (i.e. type - length - value coding). A stub link Identifier is also kept by the PS, the format of which can be varied.

For each VPN, the PS has a list of all EDs with links to the VPN sites, and has technology specific information on the tunnels to be setup for each VPN. This may include a membership list of the VPNs attributed to a stub link or configuration information for tunnel such as routing or QOS parameters.

We assume in this document that a Policy Information Base (PIB)

contains all the policy rules (akin to tunnel information) for VPN connectivity, and that each instance of a policy rule can be identified with a Policy Rule IDentifier (PRID). The policy information in the PIB is defined using the ASN.1 data definition language [5]. The representation of the policy information and the PRID on the wire

follows the Basic Encoding Rules (BER) for ASN.1 [6]. The acronym "BPD" refers to the BER-coded Policy Data in the COPS messages / objects. This model is also used for COPS for Diffserv [2].

It is the PS's responsibility to extract the information that is relevant to one ED for a given VPN and send that information when requested by the ED or when a change occurs. The PS receives information on the status of the tunnels (installed or removed) from the ED. What it does with this information and it's relationship with the domain's network or service management system(s) is out of the purview of this document.

It is also possible for the PS to coordinate the policies for tunnel establishment with other policy managed activities, such as DiffServ's Bandwidth Broker, IntServ's RSVP resource reservation system, or IPsec security policy.

5. RAP and COPS Background Information

5.1 Overview of RAP Policy Architecture

The RAP WG has proposed [4] two main architectural elements for policy control: (1) the PEP (Policy Enforcement Point) component running on a network node, such as an ED, and (2) the PDP (Policy Decision Point) off-network entity which typically resides at a Policy Server. COPS is used as the protocol between the PDP and the PEP for exchanging policy information.

The PEP is the point at which policy decisions are actually enforced or implemented. Policy decisions are made primarily at the PDP which may make use of other mechanisms and protocols to achieve additional functionality. For example, the PDP may use an LDAP-based directory service for storage and retrieval of policy information which has been populated by a management tool. How the PDP obtains the policy information and it's representation (or schema) is out of the scope of this document.

In this document, we will continue to refer to the ED and the PS rather than the equivalent PEP and PDP terminology used by RAP.

5.2 Overview of COPS for VPN Connectivity

Please refer to the base COPS protocol description [1] for a complete understanding of the protocol capabilities.

COPS runs on a TCP connection using a well-known port number. The ED opens the TCP session and then initiates a COPS session with a new client type of "vpn".

Tunnel installation information is requested by the ED by sending a Request (REQ) message. The PS replies with the information required for tunnels to be installed in one or more Decision (DEC) messages. The

protocol requires that the ED send a Report (RPT) message to the PS when the tunnel is installed (or not installed and the reason). Tunnels are activated by the PS by sending unsolicited DEC messages.

When a change occurs in the policy data for VPN connectivity, the PS can push the new configuration to the ED in an unsolicited DEC message. The RPT message is also used by the ED to periodically report back to the PS on the current status of the various tunnels.

To provide fault tolerance, COPS supports keep-alive, redirects and re-synchronization procedures to allow for the PS to be backed up, or for the load to be shared between more than one PS.

6. Scenarios for Managing VPN Connectivity

This chapter describes various scenarios for establishing, maintaining, updating, and tearing down the VPN tunnels using COPS protocol exchanges. The scenarios are:

- opening a COPS session
- requesting VPN connectivity information
- receiving VPN connectivity information
- reporting on tunnel installation
- adding a new tunnel
- removing a tunnel
- changing tunnel parameters
- failure of a COPS session

- failure of a tunnel

All the messages described below contain a Client Handle which is computed by the ED to identify a transaction. We have defined the unit of transaction as a VPN, and therefore, there is a unique handle computed for each VPN supported on the ED.

The Client Specific Information (ClientSI), and the VPN specific objects mentioned in this section are detailed in Chapters 7 and 8.

6.1 Opening a COPS Session

When the ED is brought into service, it activates all of its links. The device learns about its ED Id (or the PEPID in RAP parlance) and the VPN Ids of the sites supported on each of its own stub links. How this information is obtained (e.g. static provisioning or a dynamic discovery mechanism) is out of the scope of this document. At this point, the ED is ready to get the information to establish the tunnels for all the VPNs that it supports.

The ED locates the primary PS (and possibly one or more backup PSs) using provisioned information, or a service location protocol. The ED then establishes a TCP connection (if one is not already established) with the PS on a well-known TCP port number. Once the TCP connection with the PS is established, the ED initiates the COPS session by

sending a COPS Client Open (OPN) message to the PS containing the new client type for VPN. Refer to [1] for a complete description of the opening and closing of a COPS session.

6.2 Requesting VPN Connectivity Information

When an ED's stub link is activated, the ED must send COPS Request (REQ) message(s) to the PS asking for the connectivity information associated with each VPN Id supported on a stub link. If a VPN Id is already supported on another of its own stub link, there is still a need to ask for the tunnel information since the PS may want to change some the tunnel parameters (e.g. bandwidth) when a new stub link is comes into service, and the ED must be told which existing tunnels to use. Therefore the tunnel information is always specific to the stub link. The REQ message contains:

<Common Header> with the "vpn" client type

<Client Handle> unique handle computed by the ED for this VPN
<Context> to explain the request
- request type flag (r-type) = "configuration request"
- message type (m-type) indicating new VPN to support
<ClientSI> with a VPN Id and stub link Id - defined in section 8.3.2

6.3 Receiving VPN Connectivity Information

As a response to the ED request for the configuration for a VPN, the PS sends one or more Decision (DEC) messages containing information on the tunnels to be installed or used by the ED for one VPN. In some cases, a tunnel may need to be removed in order to install a replacement tunnel (e.g. one with different parameters). The PS can also return a "null decision" to indicate that the ED is not responsible for the establishment of any tunnel for the VPN; in this case, the PS informs the ED of the existing tunnels to use for that VPN.

The DEC message can contain information for multiple tunnels, and many DEC messages can be used to contain all the tunnel information required by an ED. It is up to the PS to determine the best way to package the information since the content is self-describing. One DEC message contains:

<Common Header> with the "vpn" client type
<Client Handle> for that VPN - same as in the REQ message
<Decision(s)> one or more decision objects
<Context> same as REQ message
<Decision: Flags> = "install", "remove", or "null decision"
<Decision: Named Data> as defined in section 8.2.3

In the case where the PS finds a COPS protocol error in the REQ message, a single <Error> object is returned in one DEC message instead of a (multiple) <Decision> object(s). The ED should take appropriate action depending on the error as indicated in the COPS protocol

specification [1]. The Error_Subcode field for VPNs is defined in section 8.2.4.

6.4 Reporting on Tunnel Installation

Once tunnel information is received in a DEC message, the ED proceeds to install each tunnel. Tunnel installation is technology

specific, as is the determination of its success or failure.

Parameters needed by the ED to guide tunnel installation and the reporting back to the PS are included in the tunnel setup information given by the PS (refer to section 2.2). We refer to these as "retry" and "reporting" parameters.

When a tunnel is installed, the ED sends to the PS a Report (RPT) message to indicate a successful installation. The status of more than one tunnel for the same VPN can be included in one RPT message, but a tunnel installation status should be sent to the PS without delay. The RPT message for successful tunnel installation contains:

<Common Header> with the "vpn" client type
<Client Handle> computed by the ER for the VPN Id
<Report Type> = "installed"
<ClientSI> PRID of installed tunnels (section 8.3.3)

In due time, the PS informs the ED to start using the tunnel just installed by sending an unsolicited DEC message with an "activate" decision flag.

<Common Header> with the "vpn" client type
<Client Handle> for that VPN
<Decision(s)> one or more decision objects
  <Context> indicating "unsolicited"
  <Decision: Flags> = "activate"
  <Decision: Named Data> PRID of tunnels to activate (section 8.2.3)

(Is there a need to send an RPT message to indicate that the tunnel was indeed activated?)

If the tunnel installation failed, the ED should periodically attempt the installation again using the retry parameters provided. If a failure needs to be reported (according the reporting parameters), the ED sends the PS an RPT message as follows:

<Common Header> with the "vpn" client type
<Client Handle> computed by the ER for the VPN Id
<Report Type> indicating "not installed"
<ClientSI> is a tunnel report with a status defined in section 8.3.3

6.5 Summary of Tunnel Establishment

The summary of the entire procedure of requesting and receiving

tunnel information, reporting on the installation, and activating the tunnel(s) (sections 6.2, 6.3 and 6.4) is depicted in Figure 2 below; only the successful case is represented.

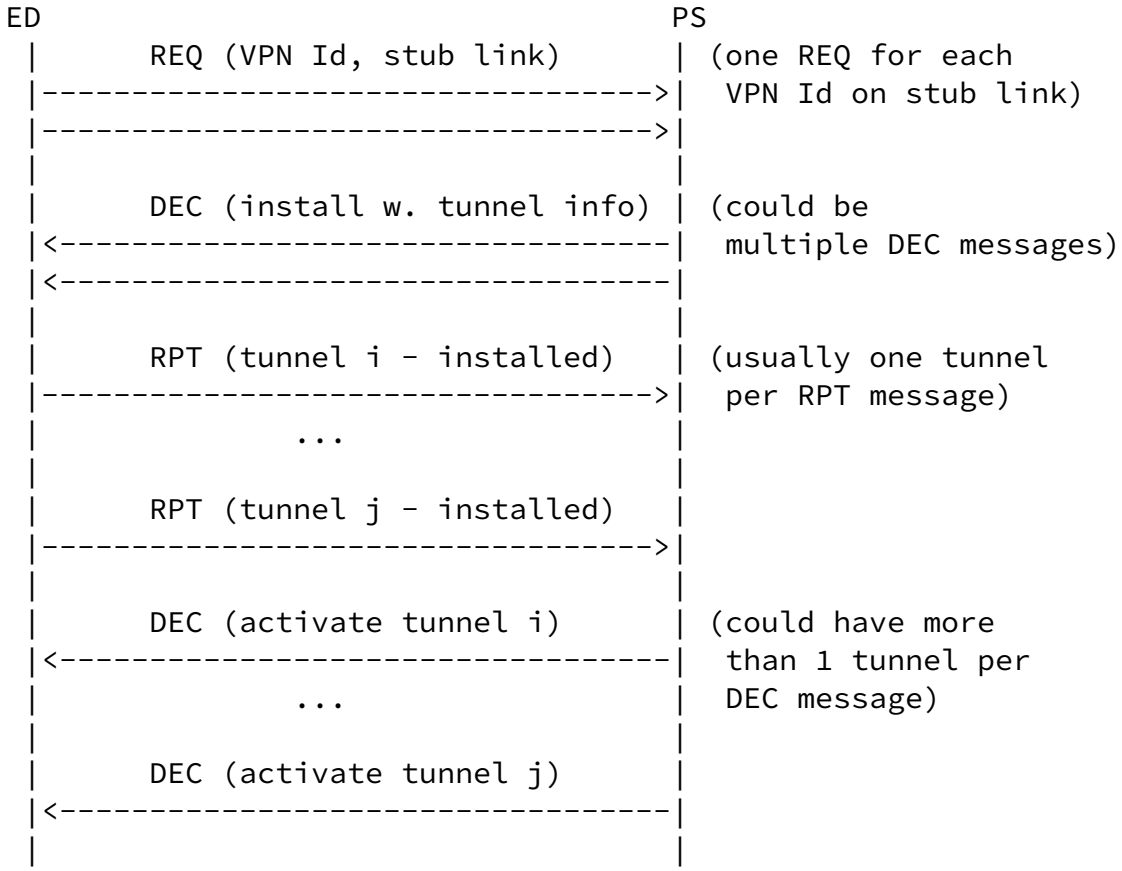


Figure 2 - COPS Message Exchange for Tunnel Establishment

6.6 Adding a New Tunnel

A tunnel may be added to the established topology under the following circumstances:

1. A new stub link is activated.
2. A new VPN Id is added to an active stub link (e.g., a new VPN site is added behind an existing stub link).
3. It is determined that a new tunnel is required between existing EDs (e.g., for engineering purposes).
4. A new tunnel is required by the PS at a specific time (e.g., the policy validity period is reached).

In the first 2 cases, the ED is made aware via a configuration update or otherwise that a new VPN Id is to be supported on a stub link. The ED then requests the tunnel information associated with the VPN Id on that stub link (see section 6.2), even if the VPN Id is already supported on

another stub link. The PS must have already been populated with the information to give to the ED.

In the last 2 cases, the PS is given the information on the new tunnel in some manner (irrelevant to this document). The PS identifies

the affected EDs and pushes the tunnel information to those EDs in unsolicited DEC messages (see section 6.3).

<Common Header> with the "vpn" client type and "unsolicited" flag
<Client Handle> computed by the ER for the VPN Id
<Decision(s)> one or more decision objects
  <Context>
    - request type flag (r-type) = "unsolicited"
  <Decision: Flags> = "install"
  <Decision: Named Data> is tunnel information (see section 8.2.3)

The DEC message elicits RPT messages (as in section 6.4) for reporting on the tunnel installation(s). When appropriate, the PS will also send DEC messages to activate the tunnel(s).

6.7 Removing a Tunnel

Tunnel removal may be required under the following circumstances:

1. A stub link is deactivated or fails.
2. A VPN Id is removed from an active stub link.
3. It is determined that a tunnel is no longer required between existing EDs (e.g., for engineering purposes).
4. The PS determines that a tunnel is no longer required (e.g., a policy validity period expires).

In the first 2 cases, the ED detects that a change in the VPNs supported on a stub link. It cannot release the resources of the tunnel(s) associated with the VPN Id for that stub link because the tunnel(s) may still be needed by another stub link. The PS must be informed because another ED may need to remove the other end of the tunnel. The ED sends a request for new configuration information because of the unsupported VPN Ids on the stub link, and waits for the PS decision to determine its course of action. The REQ message is as follows:

<Common Header> with the "vpn" client type

<Client Handle> unique handle computed by the ED for this VPN
<Context> to explain the request
- request type flag (r-type) = "configuration request"
- message type (m-type) indicating VPN no longer supported
<ClientSI> with a VPN Id and stub link - defined in section 8.3.2

The PS responds with a DEC message indicating either to "deactivate" the tunnel(s), "remove" the tunnels(s), or "install" new tunnel(s).

In the third and fourth cases, the PS sends an unsolicited DEC message with a "deactivate" or "remove" decision, indicating that the tunnel is no longer to be used or required by the ED. This is sent to the EDs at the tunnel entry and exit points.

<Common Header> with the "vpn" client type and "unsolicited" flag

<Client Handle> for the VPN Id
<Decision(s)> one or more decision objects
<Context>
- request type flag (r-type) = "unsolicited"
<Decision: Flags> = "deactivate" or "remove"
<Decision: Named Data> PRIDs of affected tunnels (see 8.2.3)

The ED then stops using a tunnel (deactivate) or releases the resources (remove) of all the affected tunnels. The ED sends RPT messages for each tunnel indicating that it was "deactivated" or "removed".

It is possible to fail to deactivate or remove a tunnel, for example, if there is not enough buffers for inter-process communication. This however is an extreme situation, and we will assume that tunnel removal and deactivation is always successful.

6.8 Changing Tunnel Parameters

Sometimes, it may be necessary to change the parameters of a tunnel. The PS is given the new parameters and uses a two step process of tunnel removal and addition. This can be done in one of 2 ways, at the PS's discretion:

Case 1: The PS first sends an unsolicited DEC message to remove the tunnel. Upon reception of the successful RPT message from the ED, the PS sends the DEC message to install and activate the tunnel with the new parameters. Operation in this way may leave a window when there is

no tunnel operating between 2 EDs for a given VPN.

Case 2: The PS first sends a DEC message to install the new tunnel. Upon reception of the successful RPT message from the ED, the PS sends the DEC message to deactivate the old tunnel and activate the new tunnel (sent in same message to be executed as one atomic operation by the ED). This way causes additional resources to be used in the network during the changeover time, but ensures continuous transmission between EDs.

The capability to change tunnel parameters while keeping the tunnel active is not considered at this point. Connection oriented tunnel technology does not support such a change.

6.9 Failure of a COPS Session

Should the communication between the PS and the ED fail, the COPS document [1] explains how the ED tries to re-establish the communication with the primary PS, and this failing, with a backup PS. The ED operates with the cached VPN connectivity information only for a specified period of time, after which it will remove the established tunnels. The time to wait before removing (or deactivating?) the tunnels could be communicated to the ED in various ways:

- As part of the configuration information received from the network

management system of the ED.

- As part of the Client Accept (CAT) message from the PS; this would require the definition of a new object to be included in the CAT message; the "policy data lifetime after communication failure" would be applicable to all COPS client types for all the policy data received from the PS; this means a change to the base COPS protocol.

- As part of the information for each tunnel; this value would override the configured value or the value received in the CAT message.

The exact way to communicate this information to the ED requires further study.

6.10 Failure of a Tunnel

When a tunnel fails, it is desirable to inform the PS to insure that the tunnel at both endpoints is removed or deactivated, and, if at all possible, that new tunnels be added to compensate for the loss of a tunnel.

Detecting a tunnel failure is technology dependent, and it is not always possible for the ED to do so. Nonetheless, any detectable failure should be reported to the PS. It is left to the ED implementation to detect the failures if it can.

When tunnel failure is detected, the ED sends to the PS a RPT message containing:

<Common Header> with the "vpn" client type
<Client Handle> computed by the ER for the VPN Id
<Report Type> indicating "removed"
<ClientSI> is a tunnel report with a status of
"removed - failed tunnel" and a technology specific reason (see section 8.3.3)

7. New COPS Objects for VPN Connectivity

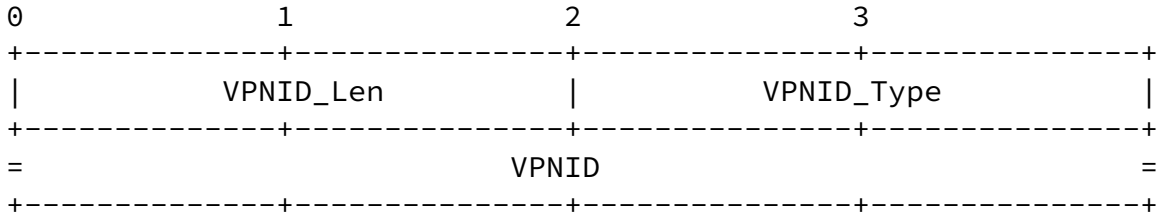
The following objects are added to COPS objects to support VPN Connectivity:

- VPN Identifier
- Stub Link Identifier
- Policy Rule Identifier (PRID)
- BER (Basic Encoding Rule) encoded Policy Data (BPD)
- Tunnel status

Please note that all objects described below are a multiple of 4 octets to align them on a 32-bit word boundary. Shorter objects are padded with zeros.

7.1 VPN Identifier (VPNID) Object

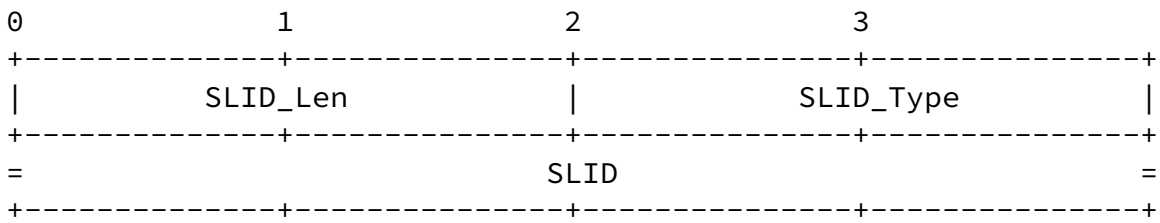
This is a PS-unique identifier for the VPN. Its exact format has not been standardized yet, although some proposals have been put forth. We will assume at this time that more than one format can be supported.



- VPNID_Len = length in octets of the entire object
- VPNID_Type = identifies the format of the VPN Id that follows (values tbd)
- VPNID = variable length VPN identifier

7.2 Stub Link IDentifier (SLID) Object

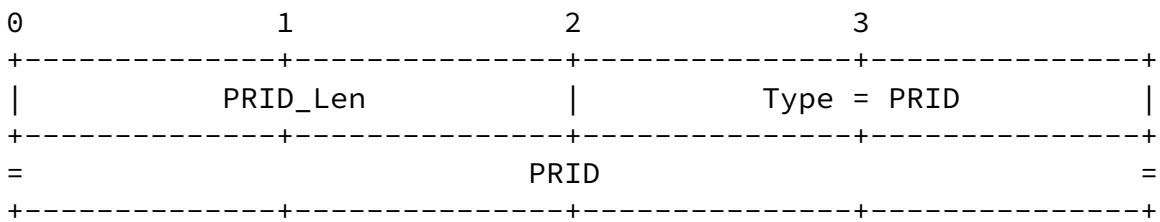
This is an identifier for the stub link of the ER. Since a format has not been standardized, we will assume at this time that more than one format can be supported.



- SLID_Len = length in octets of the entire object
- SLID_Type = identifies the format of the SLID that follows (values tbd)
- SLID = variable length Stub Link identifier

7.3 Policy Rule IDentifier (PRID) Object

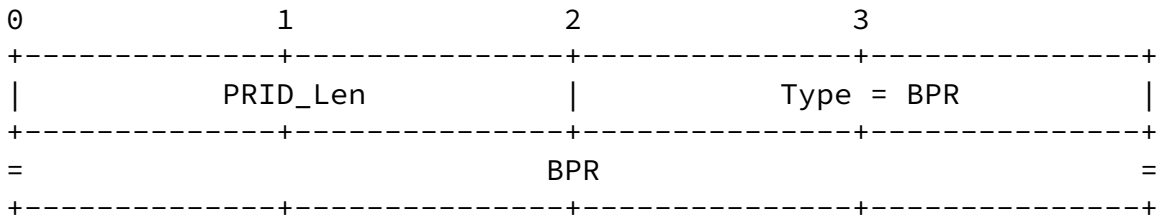
This object carries the identifier for the instance of a policy rule. The exact format of the PRID is to be determined.



- PRID_Len = length in octets of the entire object
- Type = identifies that a PRID follows
- PRID = variable length Policy Rule identifier

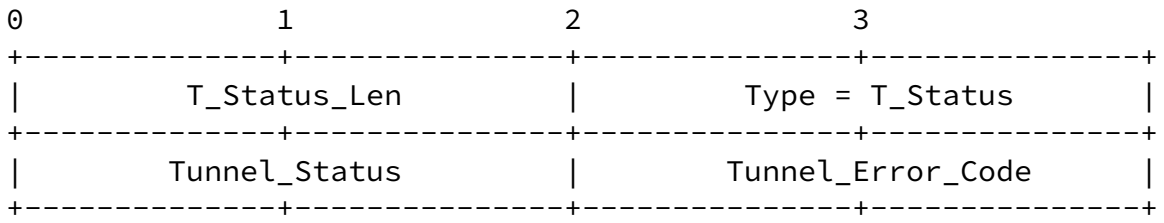
7.4 BER Encoded Policy Data (BPD) Object

This object carries the value of the instance of a policy rule which is encoded using BER (Basic Encoding Rules) for ASN.1 [6]. The content of the BPD is still to be determined (i.e. the policy schema).



7.5 Tunnel Status Object

The tunnel status object is used by the ED to report on the installation and removal of the tunnel.



Tunnel_Status
- 0 = installed
- 1 = not installed - doing periodic retries
- 2 = not installed - all periodic retries failed
- 3 = removed - PS request
- 4 = removed - failed tunnel

Tunnel_Error_Code = technology specific reason why the tunnel is not installed or being removed (values tbd)

8. COPS Extensions for VPN Connectivity

A COPS message contains a common header followed by a number of typed objects. The base COPS protocol provides for the definition of client specific information (ClientSI) in its messages and in the objects included within these messages. The extensions required to COPS messages and objects to support the VPN client type are described in detail in this chapter.

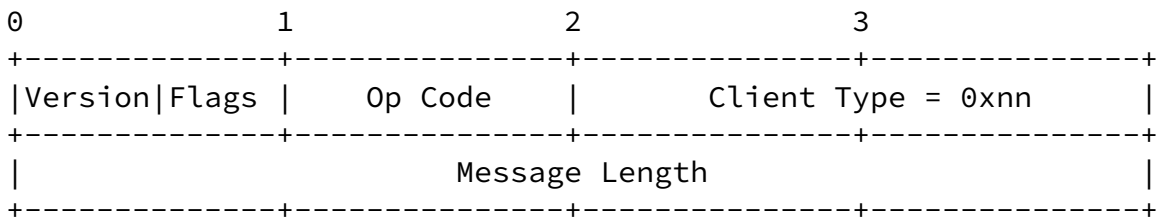
Note that there are no new messages required for VPN support in COPS. Refer to the base COPS specification [1] for the complete message and object descriptions. In case of any discrepancy between this document

and [1], the latter should be considered the authoritative text.

8.1 Common Header for VPN

The common header is used to identify the message type and client

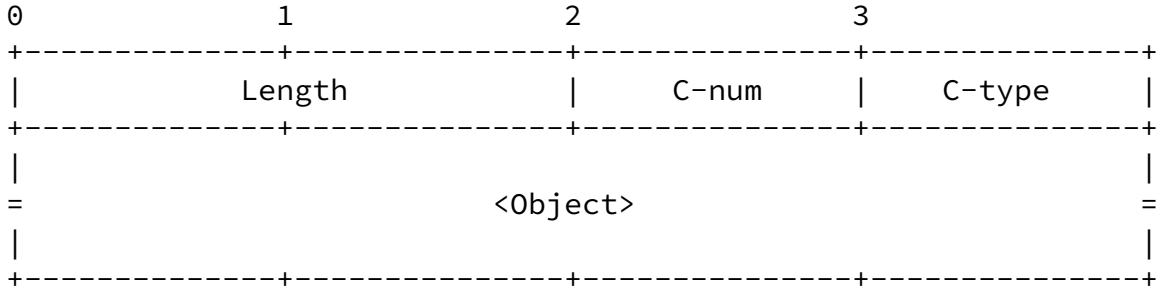
type. A new COPS client type is defined to support VPNs.



- Client Type = nn for VPN client type (need IANA reservation)

8.2 Existing COPS Objects with VPN Information

COPS objects appear after the common header and are formatted as follows:



- Length = length in octets of object header and content
- C-num = object class number (identifies the object)
- C-type = type of information specific to the C-num
- <Object> = variable length object content; specific to the C-num and C-type

Some COPS objects are defined with fields which can contain information tailored for client specific purposes. These objects are:

- Context object: the M-type field can contain VPN message types
- Reason Code object: the error sub-codes can be VPN specific
- Decision object: the ClientSI field can be defined for VPNs

- Error object: can contain error sub-codes for VPN purposes

Furthermore, the Decision object contains a command-code field which requires new commands to support VPN connectivity.

8.2.1 Context Object: Message Type Field

The Context object (c-num=2, c-type=1) is present in the Request (REQ) and Decision (DEC) messages. It contains a Request Type (R-type) field to indicate the type of request sent to the PS, and a Message Type (M-type) field to indicate a request specific to the COPS client type.

M-type values for the VPN client type are defined for a "configuration request" R-type. These are:

0x01 = new VPN supported on stub link
0x02 = VPN no longer supported on stub link

The current values in the COPS base protocol for the R-type field found in the Context object refers to the R-type of the REQ message that provoked the DEC message. For an unsolicited DEC message, where there is no prior request, a new value of "unsolicited" would make more sense than using "configuration request".

8.2.2 Reason Object: Sub-Code Field

The Reason object (c-num=5, c-type=1) is present in the Delete Request State (DRQ) message to indicate the reason why a previous request is being deleted. It contains a Reason-subcode field which can be used to indicate VPN specific information to clarify the generic Reason-code.

There are no Reason-subcode values defined for the VPN client type at this time; the field should be set to zero.

8.2.3 Decision Object

The Decision (DEC) message contains one or more Decision objects (c-num=6). Each of these objects has a c-type defined for carrying various types of decision data. There is a mandatory Decision Flags object (c-type=1) which includes a command-code field to indicate the action to

perform on the decision data:

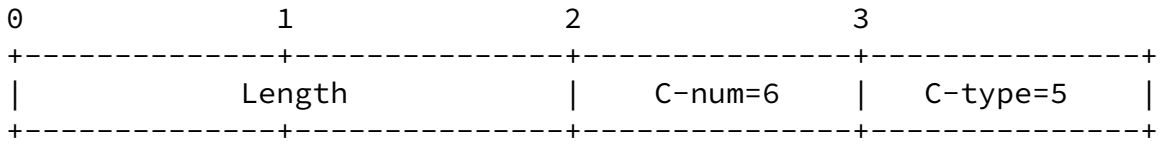
- "null decision": to indicate to use existing tunnel(s)
- "install": to indicate tunnels to install (i.e. get resources)
- "remove": to releases the tunnel resources

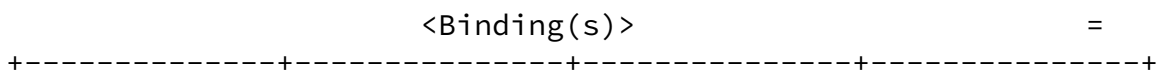
For VPN connectivity purposes, there is a need to add 3 more commands:

- "activate": to instruct the ED to start using the tunnel(s)
- "deactivate: to instruct the ED to stop using the tunnel(s)
- "deactivate & activate" to instruct the ED to swap from using one tunnel (deactivate) to using another one (activate)

Another Decision object (c-type=5) carries Named decision data. The Named data is specific to the decision flag which indicates the type of configuration data given in the decision.

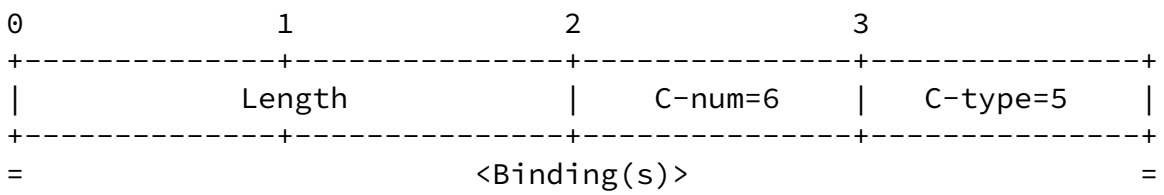
The following information is carried in the Decision object for Named data when the decision flag is "install":





<Binding(s)> ::= <Binding> | <Binding(s)> <Binding>
<Binding> ::= <PRID> <BPD>
<PRID> defined in section 7.3
<BPD> defined in section 7.4

The following information is carried in the Decision object for Named data for the decision flag values of "null decision", "remove", "activate" and "deactivate":



```
+--------------+--------------+--------------+---------------+
```

```
<Binding(s)> ::= <Binding> | <Binding(s)> <Binding>
<Binding> ::= <PRID>
```

The following information is carried in the Decision object for Named data for the "deactivate & activate" decision flag:

```
0              1               2               3
+--------------+---------------+---------------+---------------+
|            Length            |    C-num=6    |   C-type=5    |
+--------------+---------------+---------------+---------------+
=                          <Binding>                            =
+--------------+---------------+---------------+---------------+
```

```
   <Binding> ::= <PRID to deactivate> <PRID to activate>
```

8.2.4 Error Object: Error Subcode Field

The Error object (c-num=8, c-type=1) is found in the DEC message when the PS cannot respond to the ED's REQ message because of a protocol error. The Error-subcode field in this object is used to further define the Error field included in the object.

The are no Error-subcode values defined for the VPN client type at this time. The field should contain zeros.

8.3 Existing COPS Messages with ClientSI Objects

There are Client Specific Information (ClientSI) objects defined in various COPS messages which can be used to carry VPN data. These are found in the:

- Client Open (OPN) message
- Request (REQ) message

- Report State (RPT) message

The ClientSI object for VPN uses the following field settings:

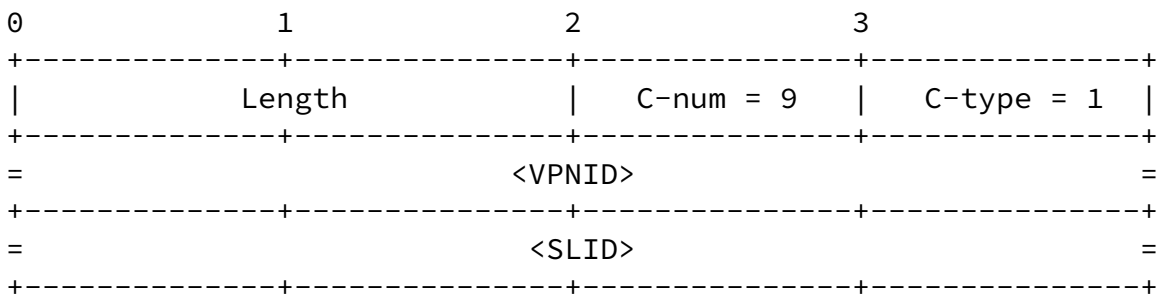
- C-num = 9; ClientSI object
- C-type = 1; Signaled ClientSI (VPN objects)
  = 2; Named ClientSI (named configuration information)

8.3.1 ClientSI Object in Open Message

The VPN ClientSI object in the Open (OPN) message is not used at this time.

8.3.2 ClientSI Object in Request Message

The VPN ClientSI object in the Request (REQ) message contains the VPN Id and the stub link Id for which configuration is requested; its format is as follows:

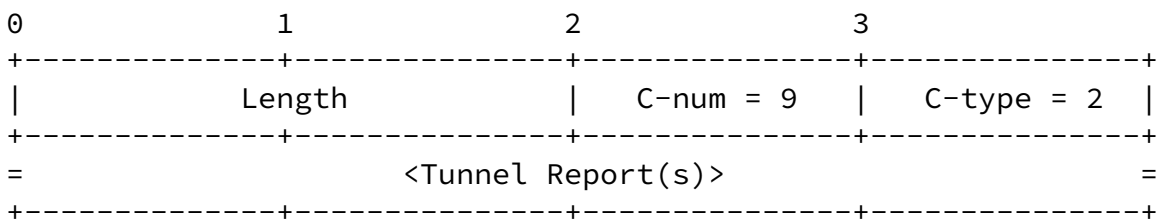


<VPNID> defined in section 7.1
<SLID> defined in section 7.2

8.3.3 ClientSI Object in Report State Message

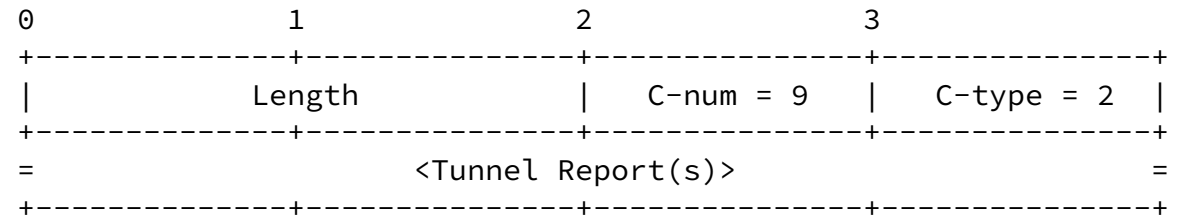
The VPN ClientSI object in the Report (RPT) message for reporting tunnel installation or removal status carries named data which is dependent on the report type.

To indicate a successful tunnel installation, the report type of "installed" is used with the following ClientSI:




```
<Tunnel Report(s)> ::= <Tunnel Report> |
                       <Tunnel Report(s)> <Tunnel Report>
<Tunnel Report> ::= <PRID>
<PRID> defined in section 7.3
```

A report type of "removed" indicates that a tunnel's resources have been released, while "not installed" or "not removed" report types indicate that the action requested by the PS to install or remove a tunnel was not successful. The ClientSI is then:




```
<Tunnel Report(s)> ::= <Tunnel Report> |
                       <Tunnel Report(s)> <Tunnel Report>
<Tunnel Report> ::= <PRID> <Tunnel Status>
<PRID> defined in section 7.3
<Tunnel Status> defined in section 7.5
```

The content of the ClientSI object when the report type is "accounting" is still to be determined.

Further study is required to determine if new report types of "activated" and "deactivated" are required to support VPNs.

9. Security Considerations

Extending COPS to provide VPN Connectivity is subject to the same security considerations as the base protocol. COPS [1] advocates using IPsec for PS-ED communication.

10. References

[1] Boyle, J. et al, "The COPS (Common Open Policy Service) Protocol", draft-ietf-rap-cops-05.txt, January 18, 1999, work in progress.

[2] Reichmeyer, F. et al, "COPS Usage for Differentiated Services", draft-ietf-rap-cops-ds-01.txt, December, 1998, work in progress.

[3] Boyle, J. et al, "COPS Usage for RSVP", draft-ietf-rap-cops-rsvp-03.txt, February 18, 1999, work in progress.

[4] Yavatkar R. et al. "A Framework for Policy-based Admission Control", draft-ietf-rap-framework-01.txt, November, 1998, work in progress.

[5] Information Processing Systems - Open Systems Interconnection - "Specification of Abstract Syntax Notation One (ASN.1)", International

Organization for Standardization. International Standard 8824, December, 1987.

[6] Information Processing Systems - Open Systems Interconnection - "Specification of the Basic Encoding Rules for Abstract Syntax Notation

One (ASN.1)", International Organization for Standardization. International Standard 8825, December, 1987.

11. Author Information

Michelle MacRae
Nortel Networks
PO Box 3511 Station C
Ottawa, Ontario K1Y 4H7
Canada
phone: (613) 763-5607
email: crm57a@nortelnetworks.com

Siamack Ayandeh
Nortel Networks
PO Box 3511 Station C
Ottawa, Ontario K1Y 4H7
Canada
phone: (613) 763-3645
email: ayandeh@nortelnetworks.com

12. Full Copyright Statement