

Network Working Group  
Internet Draft  
Intended status: Standard  
Expires: October 26, 2022

L. Dunbar  
Futurewei  
S. Hares  
Hickory Hill Consulting  
R. Raszuk  
NTT Network Innovations  
K. Majumdar  
CommScope  
Gyan Mishra  
Verizon  
April 26, 2022

**BGP UPDATE for SDWAN Edge Discovery**  
**draft-ietf-idr-sdwan-edge-discovery-02**

Abstract

The document describes the encoding of BGP UPDATE messages for the SDWAN edge node discovery.

In the context of this document, BGP Route Reflector (RR) is the component of the SDWAN Controller that receives the BGP UPDATE from SDWAN edges and in turns propagates the information to the intended peers that are authorized to communicate via the SDWAN overlay network.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

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>

This Internet-Draft will expire on Dec 25, 2022.

## Copyright Notice

Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the [Trust Legal Provisions](#) and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

|                      |                                                                             |                    |
|----------------------|-----------------------------------------------------------------------------|--------------------|
| <a href="#">1.</a>   | <a href="#">Introduction.....</a>                                           | <a href="#">3</a>  |
| <a href="#">2.</a>   | <a href="#">Conventions used in this document.....</a>                      | <a href="#">3</a>  |
| <a href="#">3.</a>   | <a href="#">Framework of SDWAN Edge Discovery.....</a>                      | <a href="#">5</a>  |
| <a href="#">3.1.</a> | <a href="#">The Objectives of SDWAN Edge Discovery.....</a>                 | <a href="#">5</a>  |
| <a href="#">3.2.</a> | <a href="#">Comparing with Pure IPsec VPN.....</a>                          | <a href="#">5</a>  |
| <a href="#">3.3.</a> | <a href="#">Client Route UPDATE and Hybrid Underlay Tunnel UPDATE.....</a>  | <a href="#">7</a>  |
| <a href="#">3.4.</a> | <a href="#">Edge Node Discovery.....</a>                                    | <a href="#">9</a>  |
| <a href="#">4.</a>   | <a href="#">BGP UPDATE to Support SDWAN Segmentation.....</a>               | <a href="#">10</a> |
| <a href="#">4.1.</a> | <a href="#">SDWAN Segmentation, SDWAN Virtual Topology and Client VPN10</a> |                    |
| <a href="#">4.2.</a> | <a href="#">Constrained Propagation of Edge Capability.....</a>             | <a href="#">11</a> |
| <a href="#">5.</a>   | <a href="#">Client Route UPDATE.....</a>                                    | <a href="#">12</a> |
| <a href="#">5.1.</a> | <a href="#">SDWAN VPN ID in Client Route Update.....</a>                    | <a href="#">13</a> |
| <a href="#">5.2.</a> | <a href="#">SDWAN VPN ID in Data Plane.....</a>                             | <a href="#">13</a> |
| <a href="#">6.</a>   | <a href="#">Hybrid Underlay Tunnel UPDATE.....</a>                          | <a href="#">13</a> |
| <a href="#">6.1.</a> | <a href="#">NLRI for Hybrid Underlay Tunnel Update.....</a>                 | <a href="#">13</a> |
| <a href="#">6.2.</a> | <a href="#">SDWAN-Hybrid Tunnel Encoding.....</a>                           | <a href="#">15</a> |
| <a href="#">6.3.</a> | <a href="#">IPsec-SA-ID Sub-TLV.....</a>                                    | <a href="#">15</a> |

|        |                                                           |                    |
|--------|-----------------------------------------------------------|--------------------|
| 6.3.1. | Encoding example #1 of using IPsec-SA-ID Sub-TLV....      | <a href="#">16</a> |
| 6.3.2. | Encoding Example #2 of using IPsec-SA-ID Sub-TLV....      | <a href="#">17</a> |
| 6.4.   | Extended Port Tunnel Encapsulation Attribute Sub-TLV..... | <a href="#">18</a> |
| 6.5.   | Underlay Network Properties Sub-TLV.....                  | <a href="#">20</a> |
| 7.     | IPsec SA Property Sub-TLVs.....                           | <a href="#">21</a> |
| 7.1.   | IPsec SA Nonce Sub-TLV.....                               | <a href="#">21</a> |
| 7.2.   | IPsec Public Key Sub-TLV.....                             | <a href="#">22</a> |
| 7.3.   | IPsec SA Proposal Sub-TLV.....                            | <a href="#">23</a> |
| 7.4.   | Simplified IPsec Security Association sub-TLV.....        | <a href="#">23</a> |
| 7.5.   | IPsec SA Encoding Examples.....                           | <a href="#">24</a> |
| 8.     | Error & Mismatch Handling.....                            | <a href="#">25</a> |
| 9.     | Manageability Considerations.....                         | <a href="#">26</a> |
| 10.    | Security Considerations.....                              | <a href="#">27</a> |
| 11.    | IANA Considerations.....                                  | <a href="#">27</a> |
| 11.1.  | Hybrid (SDWAN) Overlay SAFI.....                          | <a href="#">27</a> |
| 11.2.  | Tunnel Encapsulation Attribute Type.....                  | <a href="#">27</a> |
| 12.    | References.....                                           | <a href="#">28</a> |
| 12.1.  | Normative References.....                                 | <a href="#">28</a> |
| 12.2.  | Informative References.....                               | <a href="#">28</a> |
| 13.    | Acknowledgments.....                                      | <a href="#">30</a> |

## 1. Introduction

[SDWAN-BGP-USAGE] illustrates how BGP [[RFC4271](#)] is used as a control plane for a SDWAN network. SDWAN network refers to a policy-driven network over multiple heterogeneous underlay networks to get better WAN bandwidth management, visibility, and control.

The document describes BGP UPDATE messages for an SDWAN edge node to announce its properties to its RR which then propagates that information to the authorized peers.

## 2. Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [BCP14](#) [[RFC2119](#)] [[RFC8174](#)] when, and only when, they appear in all capitals, as shown here.

The following acronyms and terms are used in this document:

Cloud DC: Off-Premise Data Centers that usually host applications and workload owned by different organizations or tenants.

Controller: Used interchangeably with SDWAN controller to manage SDWAN overlay path creation/deletion and monitor the path conditions between sites.

CPE: Customer (Edge) Premises Equipment.

CPE-Based VPN: Virtual Private Secure network formed among CPEs. This is to differentiate such VPNs from most commonly used PE-based VPNs discussed in [[RFC4364](#)].

MP-NLRI: Multi-Protocol Network Layer Reachability Information [MP\_REACH\_NLRI] Path Attribute defined in [RFC4760](#).

SDWAN End-point: can be the SDWAN edge node address, a WAN port address (logical or physical) of a SDWAN edge node, or a client port address.

OnPrem: On Premises data centers and branch offices.

RR Route Reflector.

SDWAN: Software Defined Wide Area Network. In this document, "SDWAN" refers to policy-driven transporting IP packets over multiple different underlay networks to get better WAN bandwidth management, visibility and control.

SDWAN Segmentation: Segmentation is the process of dividing the network into logical sub-networks.

SDWAN VPN: refers to the Client's VPN, which is like the VRF on the PEs of a MPLS VPN. One SDWAN client VPN can be mapped one or multiple SD-WAN virtual topologies. How Client VPN is mapped to a SDWAN virtual topology is governed by policies.

SDWAN Virtual Topology: Since SDWAN can connect any nodes, whereas MPLS VPN connects a fixed number of PEs, one SDWAN

Virtual Topology refers to a set of edge nodes and the tunnels (including both IPsec tunnels and/or MPLS tunnels) interconnecting those edge nodes.

|     |                                      |
|-----|--------------------------------------|
| VPN | Virtual Private Network.             |
| VRF | VPN Routing and Forwarding instance. |
| WAN | Wide Area Network.                   |

### **3. Framework of SDWAN Edge Discovery**

#### **3.1. The Objectives of SDWAN Edge Discovery**

The objectives of SDWAN edge discovery are for an SDWAN edge node to discover its authorized peers and their associated properties to establish secure tunnels. The attributes to be propagated includes:

- the SDWAN (client) VPNs information,
- the attached routes under the SDWAN VPNs,
- the properties of the underlay networks over which the client routes can be carried, and potentially more.

Some SDWAN peers are connected by both trusted VPNs and untrusted public networks. Some SDWAN peers are connected only by untrusted public networks. For the traffic over untrusted networks, IPsec Security Associations (IPsec SA) must be established and maintained. If an edge node has network ports behind a NAT, the NAT properties need to be discovered by the authorized SDWAN peers.

Like any VPN networks, the attached client's routes belonging to specific SDWAN VPNs can only be exchanged with the SDWAN peer nodes authorized to communicate.

#### **3.2. Comparing with Pure IPsec VPN**

A pure IPsec VPN has IPsec tunnels connecting all edge nodes over public networks. Therefore, it requires stringent authentication and authorization (i.e., IKE Phase 1) before other properties of IPsec SA can be exchanged. The IPsec Security Association (SA) between two untrusted nodes typically requires the following configurations and message exchanges:

- IPsec IKE to authenticate with each other
- Establish IPsec SA
  - o Local key configuration
  - o Remote Peer address (192.10.0.10<->172.0.01)
  - o IKEv2 Proposal directly sent to peer
    - o Encryption method, Integrity sha512
  - o Transform set
- Attached client prefixes discovery
  - o By running routing protocol within each IPsec SA
  - o If multiple IPsec SAs between two peer nodes are established to achieve load sharing, each IPsec tunnel needs to run its own routing protocol to exchange client routes attached to the edges.
- Access List or Traffic Selector)
  - o Permit Local-IP1, Remote-IP2

In a BGP-controlled SDWAN network over hybrid MPLS VPN and public internet underlay networks, all edge nodes and RRs are already connected by private secure paths. The RRs have the policies to manage the authentication of all peer nodes. More importantly, when an edge node needs to establish multiple IPsec tunnels to many edge nodes, all the management information can be multiplexed into the secure management tunnel between RR and the edge node. Therefore, the amount of authentication in a BGP-Controlled SDWAN network can be significantly reduced.

Client VPNs are configured via VRFs, just like the configuration of the existing MPLS VPN. The IPsec equivalent traffic selectors for local and remote routes are achieved by importing/exporting VPN Route Targets. The binding of client routes to IPsec SA is dictated by policies. As a result, the IPsec configuration for a BGP controlled SDWAN (with mixed MPLS VPN) can be simplified:

- The SDWAN controller has the authority to authenticate edges and peers. Remote Peer association is controlled by the SDWAN Controller (RR)
- The IKEv2 proposals, including the IPsec Transform set, can be sent directly to peers or incorporated in a BGP UPDATE.
- BGP UPDATE: Announces the client route reachability for all permitted parallel tunnels/paths.
  - o There is no need to run multiple routing protocols in each IPsec tunnel.

- Importing/exporting Route Targets under each client VPN (VRF) achieves the traffic selection (or permission) among clients' routes attached to multiple edge nodes.

### **3.3. Client Route UPDATE and Hybrid Underlay Tunnel UPDATE**

As described in [[SDWAN-BGP-USAGE](#)], two separate BGP UPDATE messages are used for SDWAN Edge Discovery:

- UPDATE U1 for advertising the attached client routes,  
This UPDATE is precisely the same as the BGP edge client route UPDATE. It uses the Encapsulation Extended Community and the Color Extended Community to link with the Underlay Tunnels UPDATE Message as specified in [section 8 of \[RFC9012\]](#).

A new Tunnel Type (SDWAN-Hybrid) is added and used by the Encapsulation Extended Community or the Tunnel-Encap Path Attribute [[RFC9012](#)] to indicate mixed underlay networks.

- UPDATE U2 advertises the properties of the various tunnels, including IPsec, terminated at the edge node.  
This UPDATE is for an edge node to advertise the properties of directly attached underlay networks, including the NAT information, pre-configured IPsec SA identifiers, and/or the underlay network ISP information. This UPDATE can also include the detailed IPsec SA attributes, such as keys, nonce, encryption algorithms, etc.

In the following figure: there are four types underlay paths between C-PE1 and C-PE2:

- a) MPLS-in-GRE path.
- b) node-based IPsec tunnel [2.2.2.2<->1.1.1.1].
- c) port-based IPsec tunnel [192.0.0.1 <-> 192.10.0.10]; and
- d) port-based IPsec tunnel [172.0.0.1 <-> 160.0.0.1].

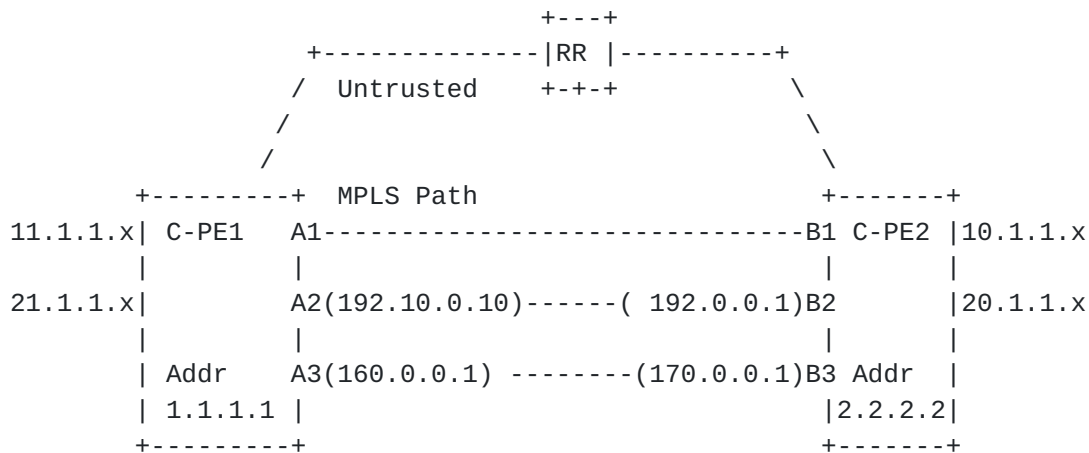


Figure 1: Hybrid SDWAN

C-PE2 uses UPDATE U1 to advertise the attached client routes:

UPDATE U1:

```

Extended community: RT for SDWAN VPN 1
NLRI: AFI=? & SAFI = 1/1
  Prefix: 10.1.1.x; 20.1.1.x
  NextHop: 2.2.2.2 (C-PE2)
Encapsulation Extended Community: tunnel-type=SDWAN-Hybrid
Color Extended Community: RED

```

The UPDATE U1 is recursively resolved to the UPDATE U2 which specifies the detailed hybrid WAN underlay tunnels terminated at the C-PE2:

UPDATE U2:

```

NLRI: SAFI = SDWAN-Hybrid
(With Color RED encoded in the NLRI Site-Property field)
Prefix: 2.2.2.2
Tunnel encapsulation Path Attribute [type=SDWAN-Hybrid]
  IPsec SA for 192.0.0.1
  Tunnel-End-Point Sub-TLV for 192.0.0.1 [RFC9012]
  IPsec-SA-ID sub-TLV [See the Section 6]
Tunnel encapsulation Path Attribute [type=SDWAN-Hybrid]
  IPsec SA for

```



```
Tunnel-End-Point Sub-TLV /* for 170.0.0.1 */  
IPsec-SA-ID sub-TLV  
Tunnel Encap Attr MPLS-in-GRE [type=SDWAN-Hybrid]  
Sub-TLV for MPLS-in-GRE [Section 3.2.6 of RFC9012]
```

Note: [\[RFC9012\] Section 11](#) specifies that each Tunnel Encap Attribute can only have one Tunnel-End-Point sub-TLV. Therefore, two separate Tunnel Encap Attributes are needed to indicate that the client routes can be carried by either one.

### **[3.4.](#) Edge Node Discovery**

The basic scheme of SDWAN edge node discovery using BGP consists of the following:

- Secure connection to a SDWAN controller (i.e., RR in this context):  
For an SDWAN edge with both MPLS and IPsec paths, the edge node should already have a secure connection to its controller, i.e., RR in this context. For an SDWAN edge that is only accessible via Internet, the SDWAN edge, upon power-up, establishes a secure tunnel (such as TLS or SSL) with the SDWAN central controller whose address is preconfigured on the edge node. The central controller informs the edge node of its local RR. The edge node then establishes a transport layer secure session with the RR (such as TLS or SSL).
- The Edge node will advertise its own properties to its designated RR via the secure connection.
- The RR propagates the received information to the authorized peers.
- The authorized peers can establish the secure data channels (IPsec) and exchange more information among each other.

For an SDWAN deployment with multiple RRs, it is assumed that there are secure connections among those RRs. How secure connections are established among those RRs is out of the scope of this document. The existing BGP UPDATE propagation mechanisms control the edge properties propagation among the RRs.

For some environments where the communication to RR is highly secured, [\[RFC9016\]](#) IKE-less can be deployed to simplify IPsec SA establishment among edge nodes.

#### **4. BGP UPDATE to Support SDWAN Segmentation**

##### **4.1. SDWAN Segmentation, SDWAN Virtual Topology and Client VPN**

In SDWAN deployment, "SDWAN Segmentation" is a frequently used term, referring to partitioning a network into multiple sub-networks, just like MPLS VPNs. "SDWAN Segmentation" is achieved by creating SDWAN virtual topologies and SDWAN VPNs. An SDWAN virtual topology consists of a set of edge nodes and the tunnels (a.k.a. underlay paths), including both IPsec tunnels and/or MPLS VPN tunnels, interconnecting those edge nodes.

An SDWAN VPN is the same as a client VPN, which is configured in the same way as the VRFs on PEs of an MPLS VPN. One SDWAN client VPN can be mapped to multiple SD-WAN virtual topologies. SDWAN Controller governs the policies of mapping a client VPN to SDWAN virtual topologies.

Each SDWAN edge node may need to support multiple VPNs. Just as a Route Target is used to distinguish different MPLS VPNs, an SDWAN VPN ID is used to differentiate the SDWAN VPNs. For example, in the picture below, the "Payment-Flow" on C-PE2 is only mapped to the virtual topology of C-PEs to/from Payment Gateway, whereas other flows can be mapped to a multipoint-to-multipoint virtual topology.

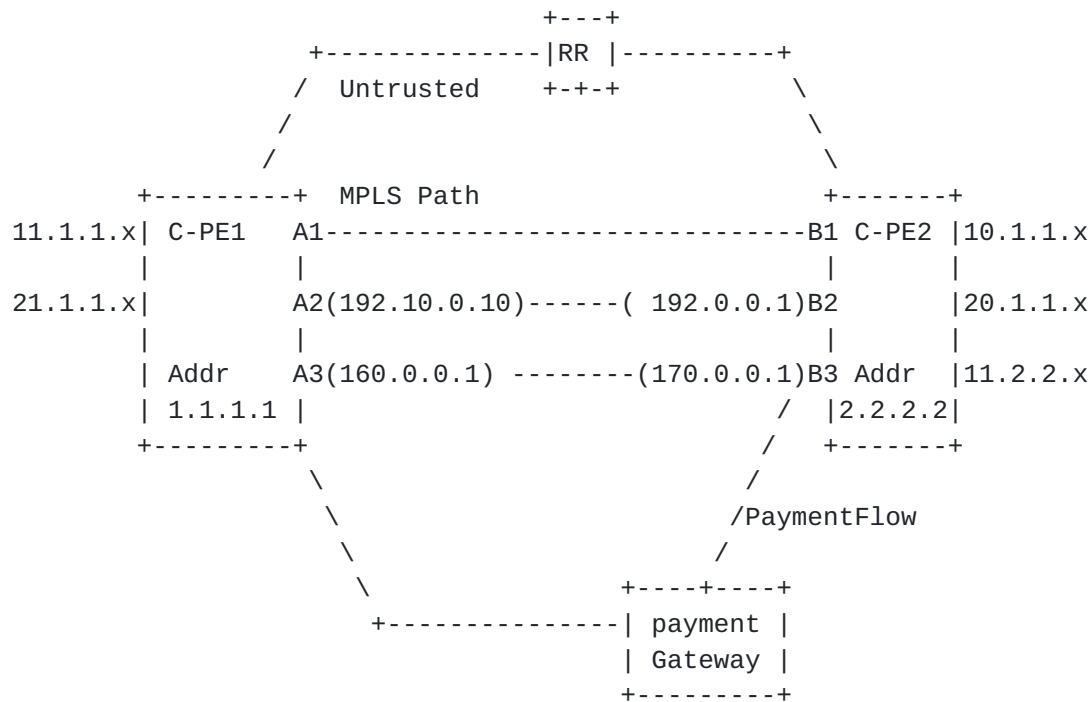


Figure 2: SDWAN Virtual Topology &amp; VPN

#### 4.2. Constrained Propagation of Edge Capability

BGP has a built-in mechanism to dynamically achieve the constrained distribution of edge information. [RFC4684] describes the BGP RT constrained distribution. In a nutshell, an SDWAN edge sends RT Constraint (RTC) NLRI to the RR for the RR to install an outbound route filter, as shown in the figure below:

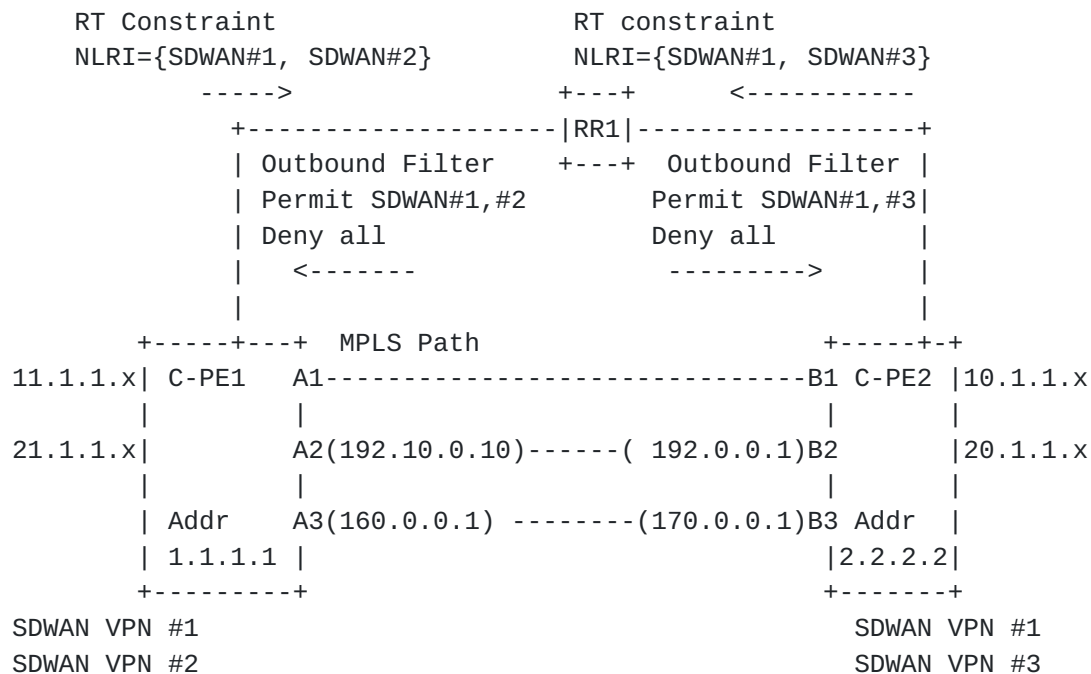


Figure 3: Constraint propagation of Edge Property

However, a SDWAN overlay network can span across untrusted networks, RR can't trust the RT Constraint (RTC) NLRI BGP UPDATE from any nodes. RR can only process the RTC NLRI from authorized peers for a SDWAN VPN.

It is out of the scope of this document on how RR is configured with the policies to filter out unauthorized nodes for specific SDWAN VPNs.

When the RR receives BGP UPDATE from an edge node, it propagates the received UPDATE message to the nodes that are in the Outbound Route filter for the specific SDWAN VPN.

## 5. Client Route UPDATE

The SDWAN network's Client Route UPDATE message is the same as the MPLS VPN client route UPDATE message. The SDWAN Client Route UPDATE message uses the Encapsulation Extended Community and the Color Extended Community to link with the Underlay Tunnels UPDATE Message.

### **5.1. SDWAN VPN ID in Client Route Update**

An SDWAN VPN is same as a client VPN in a BGP controlled SDWAN network. The Route Target Extended Community should be included in a Client Route UPDATE message to differentiate the client routes from routes belonging to other VPNs.

### **5.2. SDWAN VPN ID in Data Plane**

For an SDWAN edge node which can be reached by both MPLS and IPsec paths, the client packets reached by MPLS network will be encoded with the MPLS Labels based on the scheme specified by [[RFC8277](#)].

For GRE Encapsulation within an IPsec tunnel, the GRE key field can be used to carry the SDWAN VPN ID. For network virtual overlay (VxLAN, GENEVE, etc.) encapsulation within the IPsec tunnel, the Virtual Network Identifier (VNI) field is used to carry the SDWAN VPN ID.

## **6. Hybrid Underlay Tunnel UPDATE**

The hybrid underlay tunnel UPDATE is to advertise the detailed properties of hybrid types of tunnels terminated at a SDWAN edge node.

A client route UPDATE is recursively tied to an underlay tunnel UPDATE by the Color Extended Community included in client route UPDATE.

### **6.1. NLRI for Hybrid Underlay Tunnel Update**

A new NLRI is introduced within the MP\_REACH\_NLRI Path Attribute of [RFC4760](#), for advertising the detailed properties of hybrid types of tunnels terminated at the edge node, with SAFI=SDWAN (code = 74):

```

+-----+
|  NLRI Length  | 1 octet
+-----+
|  Site-Type    | 2 Octet
+-----+
|  Port-Local-ID | 4 octets
+-----+
|  SDWAN-Color   | 4 octets
+-----+
|  SDWAN-Node-ID | 4 or 16 octets
+-----+

```

where:

- NLRI Length: 1 octet of length expressed in bits as defined in [\[RFC4760\]](#).
- Site Type: 2 octet value. The SDWAN Site Type defines the different types of Site IDs to be used in the deployment. This document defines the following types:
  - Site-Type = 1: For a simple deployment, such as all edge nodes under one SDWAN management system, the node ID is enough for the SDWAN management to map the site to its precise geolocation.
  - Site-Type = 2: For large SDWAN heterogeneous deployment where a Geo-Loc Sub-TLV [LISP-GEOLoc] is needed to fully describe the accurate location of the node.
- Port local ID: SDWAN edge node Port identifier, which is locally significant. If the SDWAN NLRI applies to multiple ports, this field is NULL.
- SDWAN-Color: to correlate with the Color-Extended-community included in the client routes UPDATE.
- SDWAN Edge Node ID: The node's IPv4 or IPv6 address.

## 6.2. SDWAN-Hybrid Tunnel Encoding

A new BGP Tunnel-Type=SDWAN-Hybrid (code point TBD1) is specified for the Tunnel Encapsulation Attribute to indicate hybrid underlay networks.

```

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
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| Tunnel-Type(=SDWAN-Hybrid ) | Length (2 Octets) |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                               Sub-TLVs                               |
|                               |                                       |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
SDWAN Hybrid Underlay network Sub-TLV Value Field

```

## 6.3. IPsec-SA-ID Sub-TLV

IPsec-SA-ID Sub-TLV for the Hybrid Underlay Tunnel UPDATE indicates one or more preestablished IPsec SAs by using their identifiers, instead of listing all the detailed attributes of the IPsec SAs.

Using an IPsec-SA-ID Sub-TLV not only greatly reduces the size of BGP UPDATE messages, but also allows the pairwise IPsec rekeying process to be performed independently.

The following is the structure of the IPsec-SA-ID sub-TLV:

```

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
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|Type= IPsec-SA-ID subTLV (TBD2)| Length (2 Octets) |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                               IPsec SA Identifier #1                               |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                               IPsec SA Identifier #2                               |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

If the client traffic needs to be encapsulated in a specific way within the IPsec ESP Tunnel, such as GRE or VxLAN, etc., the corresponding Tunnel-Encap Sub-TLV needs to be prepended right before the IPsec-SA-ID Sub-TLV.

### 6.3.1. Encoding example #1 of using IPsec-SA-ID Sub-TLV

This section provides an encoding example for the following scenario:

- There are four IPsec SAs terminated at the same WAN Port address (or the same node address)
- Two of the IPsec SAs use GRE (value =2) as Inner Encapsulation within the IPsec Tunnel
- two of the IPsec SA uses VxLAN (value = 8) as the Inner Encapsulation within its IPsec Tunnel.

Here is the encoding for the scenario:

```

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
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| Tunnel-Type =SDWAN-Hybrid      |      Length =          |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|      Tunnel-end-Point Sub-TLV          |                  |
~                                         ~
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
~                                         ~
| subTLV-Type = IPsec-SA-ID      |      Length =          |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|      IPsec SA Identifier = 1          |                  |
+-----+-----+-----+-----+-----+-----+-----+-----+
|      IPsec SA Identifier = 2          |                  |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
~                                         ~
| subTLV-Type = IPsec-SA-ID      |      Length=          |
+-----+-----+-----+-----+-----+-----+-----+-----+
|      IPsec SA Identifier = 3          |                  |
+-----+-----+-----+-----+-----+-----+-----+-----+
|      IPsec SA Identifier = 4          |                  |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

The Length of the Tunnel-Type = SDDWAN-Hybrid is the sum of the following:

- Tunnel-end-point sub-TLV total length
- The GRE Sub-TLV total length,



- The IPsec-SA-ID Sub-TLV length,
- The VxLAN sub-TLV total length, and
- The IPsec-SA-ID Sub-TLV length.

### 6.3.2. Encoding Example #2 of using IPsec-SA-ID Sub-TLV

For IPsec SAs terminated at different endpoints, multiple Tunnel Encap Attributes must be included. This section provides an encoding example for the following scenario:

- there is one IPsec SA terminated at the WAN Port address 192.0.0.1; and another IPsec SA terminated at WAN Port 170.0.0.1;
- Both IPsec SAs use GRE (value =2) as Inner Encapsulation within the IPsec Tunnel

```

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
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| Tunnel-Type =SDWAN-Hybrid      |      Length =      |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|      Tunnel-end-Point Sub-TLV      |      |
~      for 192.0.0.1      ~
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
~      GRE Sub-TLV      ~
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
~      IPsec-SA-ID sub-TLV #1      ~
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| Tunnel-Type =SDWAN-Hybrid      |      Length =      |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|      Tunnel-end-Point Sub-TLV      |      |
~      for 170.0.0.1      ~
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
~      GRE sub-TLV      ~
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
~      IPsec-SA-ID sub-TLV #2      ~
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

#### 6.4. Extended Port Tunnel Encapsulation Attribute Sub-TLV

When a SDWAN edge node is connected to an underlay network via a port behind NAT devices, traditional IPsec uses IKE for NAT negotiation. The location of a NAT device can be such that:

- Only the initiator is behind a NAT device. Multiple initiators can be behind separate NAT devices. Initiators can also connect to the responder through multiple NAT devices.
- Only the responder is behind a NAT device.
- Both the initiator and the responder are behind a NAT device.

The initiator's address and/or responder's address can be dynamically assigned by an ISP or when their connection crosses a dynamic NAT device that allocates addresses from a dynamic address pool.

Because one SDWAN edge can connect to multiple peers via one underlay network, the pair-wise NAT exchange as IPsec's IKE is not efficient. In BGP Controlled SDWAN, NAT information of a WAN port is advertised to its RR in the BGP UPDATE message. It is encoded as an Extended sub-TLV that describes the NAT property if the port is behind a NAT device.

An SDWAN edge node can ask a STUN Server (Session Traversal of UDP Through Network Address Translation [[RFC3489](#)]) to get the NAT properties, the public IP address and the Public Port number to pass to peers.

```

 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
+-+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|Port Ext Type | EncapExt subTLV Length          |I|O|R|R|R|R|R|
+-+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
| NAT Type     | Encap-Type   |Trans networkID|   RD ID   |
+-+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|
|               Local  IP Address                  |
|               32-bits for IPv4, 128-bits for Ipv6
|               ~~~~~~
+-+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|               Local  Port                        |
+-+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+

```

```

|          Public IP          |
| 32-bits for IPv4, 128-bits for Ipv6 |
| ~~~~~~ |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|          Public Port          |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|          ISP-Sub-TLV          |
| ~~~~~~ |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

Where:

- o Port Ext Type (TBD3): indicating it is the Port Ext SubTLV.
- o PortExt subTLV Length: the length of the subTLV.
- o Flags:
  - I bit (CPE port address or Inner address scheme)
    - If set to 0, indicate the inner (private) address is IPv4.
    - If set to 1, it indicates the inner address is IPv6.
  - O bit (Outer address scheme):
    - If set to 0, indicate the public (outer) address is IPv4.
    - If set to 1, it indicates the public (outer) address is IPv6.
  - R bits: reserved for future use. Must be set to 0 now.
- o NAT Type.without NAT; 1:1 static NAT; Full Cone; Restricted Cone; Port Restricted Cone; Symmetric; or Unknown (i.e. no response from the STUN server).
- o Encap Type.the supported encapsulation types for the port facing public network, such as IPsec+GRE, IPsec+VxLAN, IPsec without GRE, GRE (when packets don't need encryption)
- o Transport Network ID.Central Controller assign a global unique ID to each transport network.
- o RD ID.Routing Domain ID.need to be global unique.
- o Local IP.The local (or private) IP address of the port.
- o Local Port.used by Remote SDWAN edge node for establishing IPsec to this specific port.

- o Public IP.The IP address after the NAT. If NAT is not used, this field is set to NULL.
- o Public Port.The Port after the NAT. If NAT is not used, this field is set to NULL.

#### 6.5. Underlay Network Properties Sub-TLV

The purpose of the Underlay Network Sub-TLV is to carry the WAN port properties with SDWAN SAFI NLRI. It would be treated as optional Sub-TLV. The BGP originator decides whether to include this Sub-TLV along with the SDWAN NLRI. If this Sub-TLV is present, it would be processed by the BGP receiver and to determine what local policies to apply for the remote end point of the underlay tunnel.

The format of this Sub-TLV is as follows:

```

  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
  +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
  | Type=TBD4      |      Length      |      Flag      |      Reserved      |
  +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
  |Connection Type|      Port Type      |      Port Speed      |
  +---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

Where:

Type: TBD4.

Length: always 6 bytes.

Flag: a 1 octet value.

Reserved: 1 octet of reserved bits. It SHOULD be set to zero on transmission and MUST be ignored on receipt.

Connection Type: There are two different types of WAN Connectivity. They are listed below as:

```

Wired - 1
WIFI - 2
LTE - 3
5G - 4

```

Port Type: There are different types of ports. They are listed Below as:

Ethernet - 1  
Fiber Cable - 2  
Coax Cable - 3  
Cellular - 4

Port Speed: The port seed is defined as 2 octet value. The values are defined as Gigabit speed.

## **7. IPsec SA Property Sub-TLVs**

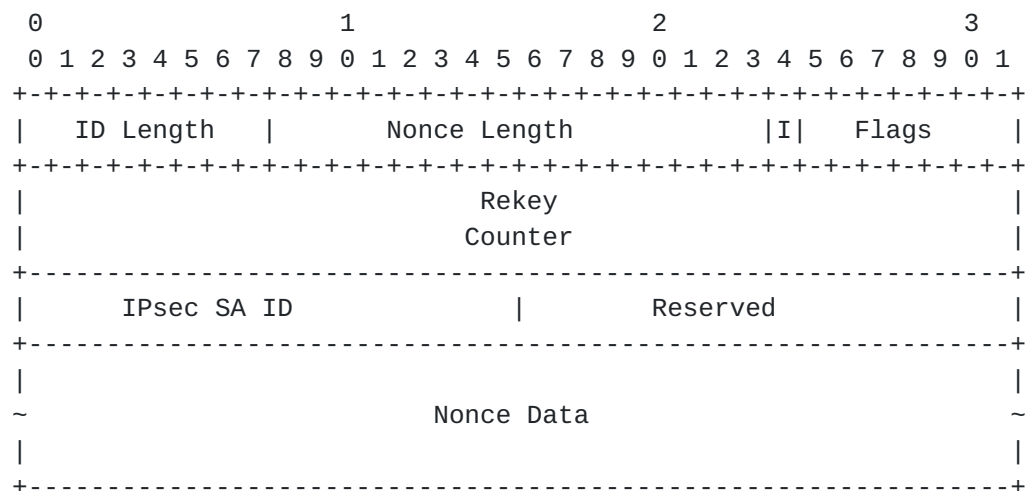
This section describes the detailed IPsec SA properties sub-TLVs.

### **7.1. IPsec SA Nonce Sub-TLV**

The Nonce Sub-TLV is based on the Base DIM sub-TLV as described the Section 6.1 of [[SECURE-EVPN](#)]. The IPsec SA ID is included in the sub-TLV, which is to be referenced by the client route NLRI Tunnel Encapsulation Path Attribute for the IPsec SA. The following fields are removed because:

- the Originator ID is carried by the NLRI,
- the Tenant ID is represented by the SDWAN VPN ID Extended Community, and
- the Subnet ID are carried by the BGP route UPDATE.

The format of this Sub-TLV is as follows:

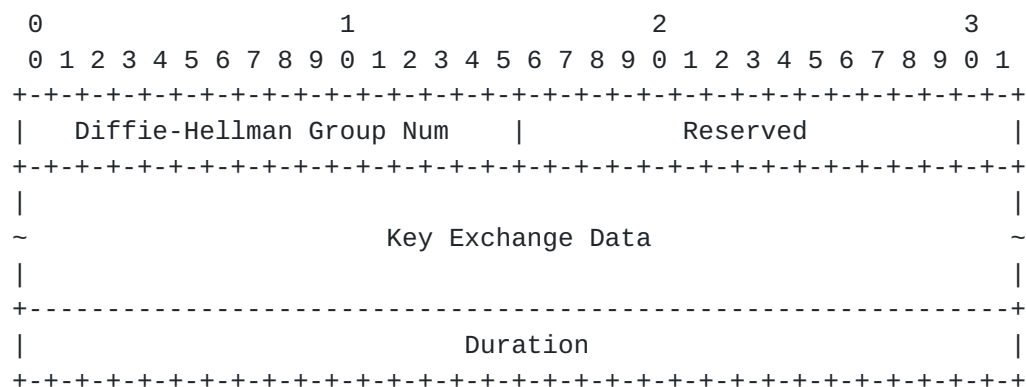


IPsec SA ID - The 2 bytes IPsec SA ID could 0 or non-zero values. It is cross referenced by client route's IPsec Tunnel Encapsulation IPsec-SA-ID SubTLV in [Section 6](#). When there are multiple IPsec SAs terminated at one address, such as WAN port address or the node address, they are differentiated by the different IPsec SA IDs.

## 7.2. IPsec Public Key Sub-TLV

The IPsec Public Key Sub-TLV is derived from the Key Exchange Sub-TLV described in [\[SECURE-EVPN\]](#) with an addition of Duration field to define the IPsec SA life span. The edge nodes would pick the shortest duration value between the SDWAN SAFI pairs.

The format of this Sub-TLV is as follows:



### 7.3. IPsec SA Proposal Sub-TLV

The IPsec SA Proposal Sub-TLV is to indicate the number of Transform Sub-TLVs. This Sub-TLV aligns with the sub-TLV structure from [SECURE-VPN]

The Transform Sub-sub-TLV will following the [section 3.3.2 of RFC7296](#).

### 7.4. Simplified IPsec Security Association sub-TLV

For a simple SDWAN network with edge nodes supporting only a few pre-defined encryption algorithms, a simple IPsec sub-TLV can be used to encode the pre-defined algorithms, as below:

```

 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
+-+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|IPsec-simType |IPsecSA Length          | Flag          |
+-+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
| Transform    | Mode                    | AH algorithms |ESP algorithms |
+-+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|              ReKey Counter (SPI)              |
+-+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
| key1 length  |          Public Key          ~
+-+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
| key2 length  |          Nonce              ~
+-+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|              Duration                      |
+-+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+

```

Where:

- o IPsec-SimType: The type value has to be between 128~255 because IPsec-SA subTLV needs 2 bytes for length to carry the needed information.
- o IPsec-SA subTLV Length (2 Byte): 25 (or more)
- o Flags: 1 octet of flags. None are defined at this stage. Flags SHOULD be set to zero on transmission and MUST be ignored on receipt.
- o Transform (1 Byte): the value can be AH, ESP, or AH+ESP.

- o IPsec Mode (1 byte): the value can be Tunnel Mode or Transport mode
- o AH algorithms (1 byte): AH authentication algorithms supported, which can be md5 | sha1 | sha2-256 | sha2-384 | sha2-512 | sm3. Each SDWAN edge node can have multiple authentication algorithms; send to its peers to negotiate the strongest one.
- o ESP (1 byte): ESP authentication algorithms supported, which can be md5 | sha1 | sha2-256 | sha2-384 | sha2-512 | sm3. Each SDWAN edge node can have multiple authentication algorithms; send to its peers to negotiate the strongest one. Default algorithm is AES-256.
  - o When node supports multiple authentication algorithms, the initial UPDATE needs to include the "Transform Sub-TLV" described by [\[SECURE-EVPN\]](#) to describe all of the algorithms supported by the node.
- o Rekey Counter (Security Parameter Index): 4 bytes
- o Public Key: IPsec public key
- o Nonce: IPsec Nonce
- o Duration: SA life span.

### 7.5. IPsec SA Encoding Examples

For the Figure 1 in [Section 3](#), C-PE2 needs to advertise its IPsec SA associated attributes, such as the public keys, the nonce, the supported encryption algorithms for the IPsec tunnels terminated at 192.0.0.1, 170.1.1.1 and 2.2.2.2 respectively.

Using the IPsec Tunnel [ISP4: 160.0.0.1 <-> ISP2:170.0.0.1] as an example: C-PE1 needs to advertise the following attributes for establishing the IPsec SA:

- SDWAN Node ID

- SDWAN Color

- Tunnel Encap Attr (Type=SDWAN-Hybrid)

- Extended Port Sub-TLV for information about the Port

- (including ISP Sub-TLV for information about the ISP2)

- IPsec SA Nonce Sub-TLV,

- IPsec SA Public Key Sub-TLV,

- IPsec SA Sub-TLV for the supported transforms



```
{Transforms Sub-TLV - Trans 2,  
Transforms Sub-TLV - Trans 3}
```

C-PE2 needs to advertise the following attributes for establishing IPsec SA:

- SDWAN Node ID

- SDWAN Color

- Tunnel Encap Attr (Type=SDWAN-Hybrid)

  - Extended Port Sub-TLV (including ISP Sub-TLV for information about the ISP2)

  - IPsec SA Nonce Sub-TLV,

  - IPsec SA Public Key Sub-TLV,

  - IPsec SA Sub-TLV for the supported transforms

    - {Transforms Sub-TLV - Trans 2,  
Transforms Sub-TLV - Trans 4}

As both end points support Transform #2, the Transform #2 will be used for the IPsec Tunnel [ISP4: 160.0.0.1 <-> ISP2:170.0.0.1].

## **8. Error & Mismatch Handling**

Each C-PE device advertises a SDWAN SAFI Underlay NLRI to the other C-PE devices via a BGP Route Reflector to establish pairwise SAs between itself and every other remote C-PEs. During the SAFI NLRI advertisement, the BGP originator would include either simple IPsec Security Association properties defined in IPsec SA Sub-TLV based on IPsec-SA-Type = 1 or full-set of IPsec Sub-TLVs including Nonce, Public Key, Proposal and number of Transform Sub-TLVs based on IPsec-SA-Type = 2.

The C-PE devices compare the IPsec SA attributes between the local and remote WAN ports. If there is a match on the SA Attributes between the two ports, the IPsec Tunnel is established.

The C-PE devices would not try to negotiate the base IPsec-SA parameters between the local and the remote ports in the case of simple IPsec SA exchange or the Transform sets between local and remote ports if there is a mismatch on the Transform sets in the case of full-set of IPsec SA Sub-TLVs.

As an example, using the Figure 1 in [Section 3](#), to establish IPsec Tunnel between C-PE1 and C-PE2 WAN Ports A2 and B2 [A2: 192.10.0.10 <-> B2:192.0.0.1]:

C-PE1 needs to advertise the following attributes for establishing the IPsec SA:

- NH: 192.10.0.10
- SDWAN Node ID
- SDWAN-Site-ID
- Tunnel Encap Attr (Type=SDWAN)
  - ISP Sub-TLV for information about the ISP3
  - IPsec SA Nonce Sub-TLV,
  - IPsec SA Public Key Sub-TLV,
  - Proposal Sub-TLV with Num Transforms = 1
  - {Transforms Sub-TLV - Trans 1}

C-PE2 needs to advertise the following attributes for establishing IPsec SA:

- NH: 192.0.0.1
- SDWAN Node ID
- SDWAN-Site-ID
- Tunnel Encap Attr (Type=SDWAN)
  - ISP Sub-TLV for information about the ISP1
  - IPsec SA Nonce Sub-TLV,
  - IPsec SA Public Key Sub-TLV,
  - Proposal Sub-TLV with Num Transforms = 1
  - {Transforms Sub-TLV - Trans 2}

As there is no matching transform between the WAN ports A2 and B2 in C-PE1 and C-PE2 respectively, there will be no IPsec Tunnel be established.

## 9. Manageability Considerations

TBD - this needs to be filled out before publishing

## **10. Security Considerations**

The document describes the encoding for SDWAN edge nodes to advertise its properties to their peers to its RR, which propagates to the intended peers via untrusted networks.

The secure propagation is achieved by secure channels, such as TLS, SSL, or IPsec, between the SDWAN edge nodes and the local controller RR.

[More details need to be filled in here]

## **11. IANA Considerations**

### **11.1. Hybrid (SDWAN) Overlay SAFI**

IANA has assigned SAFI = 74 as the Hybrid (SDWAN)SAFI.

### **11.2. Tunnel Encapsulation Attribute Type**

IANA is requested to assign a type from the BGP Tunnel Encapsulation Attribute Tunnel Types as follows:

| Value | Description  | Reference       |
|-------|--------------|-----------------|
| ----- | -----        | -----           |
| TBD1  | SDWAN-Hybrid | [this document] |

### **11.3 Tunnel Encapsulation Attribute Sub-TLV Types**

IANA is requested to assign three Types, as follows, in the BGP Tunnel Encapsulation Attribute Sub-TLVs registry:

| Value | Description             | Reference       |
|-------|-------------------------|-----------------|
| ----- | -----                   | -----           |
| TBD2  | IPSEC-SA-ID             | [this document] |
| TBD3  | Port Extension          | [this document] |
| TBD4  | Underlay ISP Properties | [this document] |

## [12. References](#)

### 12.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [RFC4271] Rekhter, Y., Ed., Li, T., Ed., and S. Hares, Ed., "A Border Gateway Protocol 4 (BGP-4)", [RFC 4271](#), DOI 10.17487/RFC4271, January 2006, <<https://www.rfc-editor.org/info/rfc4271>>.
- [RFC4760] Bates, T., Chandra, R., Katz, D., and Y. Rekhter, "Multiprotocol Extensions for BGP-4", [RFC 4760](#), DOI 10.17487/RFC4760, January 2007, <<https://www.rfc-editor.org/info/rfc4760>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in [RFC 2119](#) Key Words", [BCP 14](#), [RFC 8174](#), DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC9012] Patel, K., Van de Velde, G., Sangli, S., and J. Scudder, "The BGP Tunnel Encapsulation Attribute", [RFC 9012](#), DOI 10.17487/RFC9012, April 2021, <<https://www.rfc-editor.org/info/rfc9012>>.

### 12.2. Informative References

- [RFC8192] S. Hares, et al, "Interface to Network Security Functions (I2NSF) Problem Statement and Use Cases", July 2017
- [RFC5521] P. Mohapatra, E. Rosen, "The BGP Encapsulation Subsequent Address Family Identifier (SAFI) and the BGP Tunnel Encapsulation Attribute", April 2009.

- [RFC9061] Marin-Lopez, R., Lopez-Millan, G., and F. Pereniguez-Garcia, "A YANG Data Model for IPsec Flow Protection Based on Software-Defined Networking (SDN)", [RFC 9061](#), DOI 10.17487/RFC9061, July 2021, <<https://www.rfc-editor.org/info/rfc9061>>.
- [CONTROLLER-IKE] D. Carrel, et al, "IPsec Key Exchange using a Controller", [draft-carrel-ipsecme-controller-ike-01](#), work-in-progress.
- [LISP-GEOLoc] D. Farinacci, "LISP Geo-Coordinate Use-Case", [draft-farinacci-lisp-geo-09](#), April 2020.
- [SDN-IPSEC] R. Lopez, G. Millan, "SDN-based IPsec Flow Protection", [draft-ietf-i2nsf-sdn-ipsec-flow-protection-07](#), Aug 2019.
- [SECURE-EVPN] A. Sajassi, et al, "Secure EVPN", [draft-sajassi-bess-secure-evpn-02](#), July 2019.
- [VPN-over-Internet] E. Rosen, "Provide Secure Layer L3VPNs over Public Infrastructure", [draft-rosen-bess-secure-l3vpn-00](#), work-in-progress, July 2018
- [DMVPN] Dynamic Multi-point VPN:  
<https://www.cisco.com/c/en/us/products/security/dynamic-multipoint-vpn-dmvpn/index.html>
- [DSVPN] Dynamic Smart VPN:  
<http://forum.huawei.com/enterprise/en/thread-390771-1-1.html>
- [ITU-T-X1036] ITU-T Recommendation X.1036, "Framework for creation, storage, distribution and enforcement of policies for network security", Nov 2007.
- [Net2Cloud-Problem] L. Dunbar and A. Malis, "Dynamic Networks to Hybrid Cloud DCs Problem Statement", [draft-ietf-rtgwg-net2cloud-problem-statement-12](#), March 7, 2022.

[Net2Cloud-gap] L. Dunbar, A. Malis, and C. Jacquenet, "Networks Connecting to Hybrid Cloud DCs: Gap Analysis", [draft-ietf-rtgwg-net2cloud-gap-analysis-07](#), July, 2020.

[RFC9012] K. Patel, et al "The BGP Tunnel Encapsulation Attribute", [RFC9012](#), April 2021.

### **13. Acknowledgments**

Acknowledgements to Wang Haibo, Hao Weiguo, and ShengCheng for implementation contribution; Many thanks to Yoav Nir, Graham Bartlett, Jim Guichard, John Scudder, and Donald Eastlake for their review and suggestions.

This document was prepared using 2-Word-v2.0.template.dot.

## Authors' Addresses

Linda Dunbar  
Futurewei  
Email: ldunbar@futurewei.com

Sue Hares  
Hickory Hill Consulting  
Email: shares@ndzh.com

Robert Raszuk  
NTT Network Innovations  
Email: robert@raszuk.net

Kausik Majumdar  
CommScope  
Email: Kausik.Majumdar@commscope.com

Gyan Mishra  
Verizon Inc.  
Email: gyan.s.mishra@verizon.com

## Contributors' Addresses

Donald Eastlake  
Futurewei  
Email: d3e3e3@gmail.com