FlexVPN Spoke in Redundant Hub Design with FlexVPN Client Block Configuration Example

Document ID: 116413

Contributed by Marcin Latosiewicz, Cisco TAC Engineer.
Sep 16, 2013

Contents

Introduction

This document describes how to configure a spoke in a FlexVPN network with use of the FlexVPN client configuration block in a scenario where multiple hubs are available.

Prerequisites

Requirements

Cisco recommends that you have knowledge of these topics:

- FlexVPN
- Cisco Routing Protocols

Components Used

The information in this document is based on these software and hardware versions:

- Cisco G2 Series Integrated Service Router (ISR)
- Cisco IOS® Version 15.2M

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, make sure that you understand the potential impact of any command.

Background Information

For redundancy purposes, a spoke might need to connect to multiple hubs. Redundancy on the spoke−side allows continuous operation without a single point of failure on the hub−side.

The two most common FlexVPN redundant hub designs that use the spoke configuration are:

- ***Dual cloud approach***, where a spoke has two separate tunnels active to both hubs at all times.
- ***Failover approach***, where a spoke has an active tunnel with one hub at any given point in time.

Both approaches have a unique set of pros and cons.

Approach	***Pros***	***Cons***
Dual cloud	• Faster recovery in a failure, based on routing protocol timers • More possibilties to distribute traffic among hubs, since the connections to both hubs are active	• Spoke maintains session to both hubs at the same time, which consumes resources on both hubs
Failover	• Easy configuration − built into FlexVPN • Does not rely on routing protocol in a failure	• Slower recovery time − based on Dead Peer Detection (DPD) or (optionally) object tracking • All traffic is forced to travel to one hub at a time

This document describes the second approach.

Configure

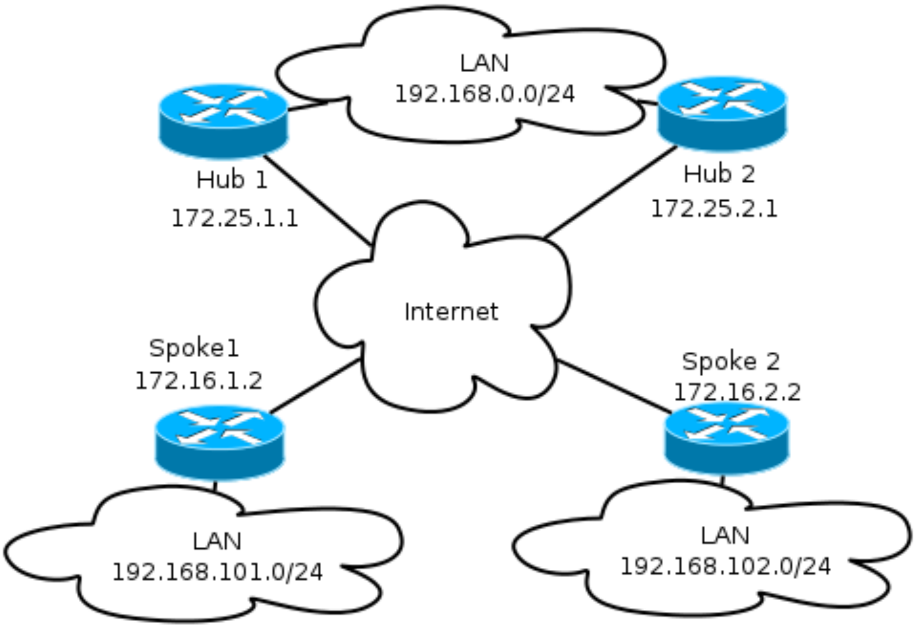
Note: Use the Command Lookup Tool (registered customers only) in order to obtain more information on the commands used in this section.

Network Diagrams

These diagrams show both the transport and overlay topology diagrams.

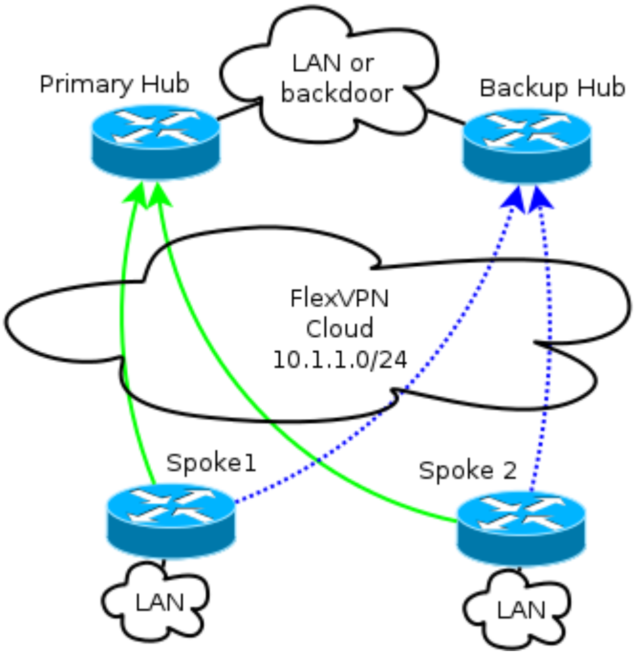
Transport Network

This diagram illustrates the basic transport network that is typically used in FlexVPN networks.



Overlay Network

This diagram illustrates the overlay network with logical connectivity that shows how the failover should work. During normal operation, Spoke 1 and Spoke 2 maintain a relationship with one hub only.



Note: In the diagram, the solid green lines show the connection and direction of primary Internet Key Exchange Version 2 (IKEv2)/Flex sessions, and the dotted blue lines indicate the backup connection should the Internet Key Exchange (IKE) session to the primary hub fail.

The ***/24*** addressing represents the pool of addresses allocated for this cloud, and not the actual interface addressing. This is because the FlexVPN hub typically allocates a dynamic IP address for the spoke interface, and relies on routes inserted dynamically via route commands in the FlexVPN authorization block.

Basic Configuration of Spoke and Hub

The basic configuration of the hub and spoke is based on migration documents from Dynamic Multipoint VPN (DMVPN) to FlexVPN. This configuration is described in the FlexVPN Migration: Hard Move from DMVPN to FlexVPN on Same Devices article.

Spoke Configuration Adjustment

Spoke Configuration − Client Configuration Block

Spoke configuration must be extended by the client configuration block.

In the basic configuration, multiple peers are specified. The peer with the highest preference (lowest number) is considered before others.

```
crypto ikev2 client flexvpn Flex_Client
 peer 1 172.25.1.1
 peer 2 172.25.2.1
 client connect Tunnel1
```

The tunnel configuration must change in order to allow the tunnel destination to be chosen dynamically, based on the FlexVPN client configuration block.

```
interface Tunnel1
  tunnel destination dynamic
```

It is crucial to remember that the FlexVPN client configuration block is tied to an interface, and not to IKEv2 or the Internet Protocol Security (IPsec) profile.

The client configuration block provides multiple options in order to adjust the failover time and operations, which include tracking objects usage, dial backup, and backup groups functionalities.

With basic configuration, the spoke relies on DPDs in order to detect whether a spoke is unresponsive, and it triggers a change once the peer is declared dead. The option to use DPD is not a fast one, because of how DPDs work. An administrator might want to enhance the configuration with object tracking or similar enhancements.

For more information, refer to the ***FlexVPN Client Configuration*** chapter of the Cisco IOS configuration guide, which is linked in the ***Related Information*** section at the end of this document.

Full Spoke Configuration − Reference

```
crypto logging session

crypto ikev2 keyring Flex_key
 peer Spokes
  address 0.0.0.0 0.0.0.0
  pre−shared−key local cisco
  pre−shared−key remote cisco

crypto ikev2 profile Flex_IKEv2
 match identity remote address 0.0.0.0
 authentication remote pre−share
 authentication local pre−share
 keyring local Flex_key
 aaa authorization group psk list default default
 virtual−template 1
```

```
crypto ikev2 dpd 30 5 on-demand

crypto ikev2 client flexvpn Flex_Client
  peer 1 172.25.1.1
  peer 2 172.25.2.1
  client connect Tunnel1

crypto ipsec transform-set IKEv2 esp-gcm
 mode transport

crypto ipsec profile default
 set ikev2-profile Flex_IKEv2

interface Tunnel1
 description FlexVPN tunnel
 ip address negotiated
 ip mtu 1400
 ip nhrp network-id 2
 ip nhrp shortcut virtual-template 1
 ip nhrp redirect
 ip tcp adjust-mss 1360
 delay 2000
 tunnel source Ethernet0/0
 tunnel destination dynamic
 tunnel path-mtu-discovery
 tunnel protection ipsec profile default
```

Hub Configuration

While the majority of the hub configuration remains the same, several aspects must be addressed. Most of them pertain to a situation in which one or more spokes are connected to one hub, while others remain in relationship to another hub.

Spoke Addresses

Since spokes obtain IP addresses from hubs, it is normally desired that hubs assign addresses from different subnets or a different part of a subnet.

For example:

Hub1

```
ip local pool FlexSpokes 10.1.1.100 10.1.1.175
```

Hub2

```
ip local pool FlexSpokes 10.1.1.176 10.1.1.254
```

This prevents overlap creation, even if the addresses are not routed outside of the FlexVPN cloud, which might impair troubleshooting.

Hub Overlay Address

Both hubs can retain the same IP address on a virtual−template interface; however, this can impact troubleshooting in some cases. This design choice makes it easier to deploy and plan, since the spoke must have only one peer address for Border Gateway Protocol (BGP).

In some cases, it might not be desired or needed.

Routing

It is necessary for hubs to exchange information about the spokes that are connected.

Hubs must be able to exchange the specific routes of devices they have connected, and still provide a summary to the spokes.

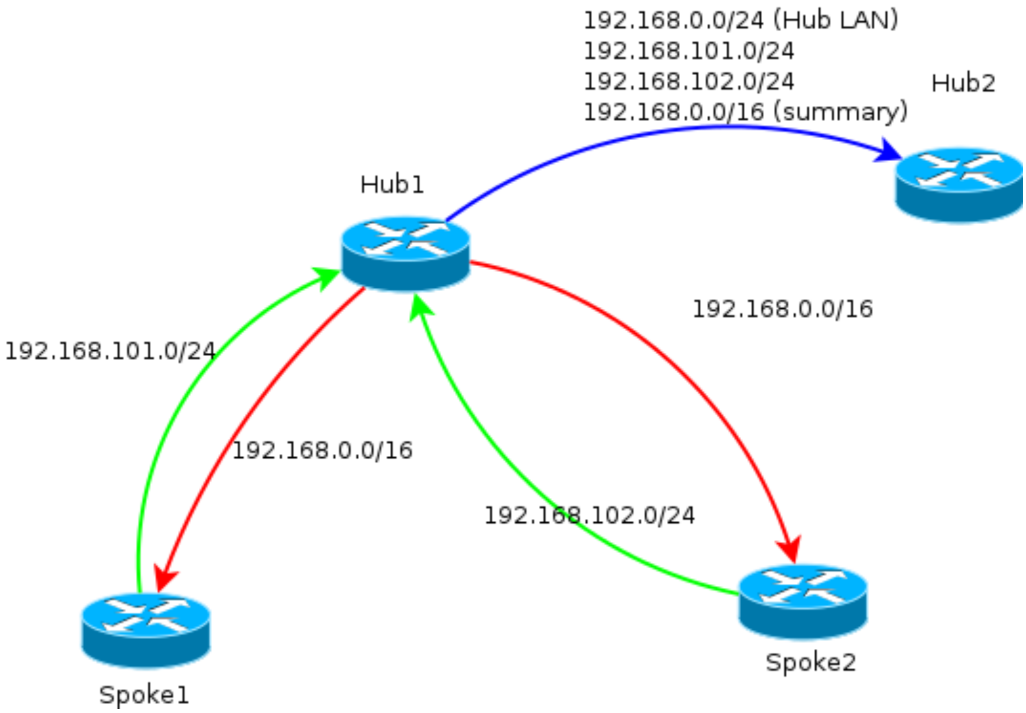
Since Cisco recommends that you use iBGP with FlexVPN and DMVPN, only that routing protocol is shown.

```
bgp log−neighbor−changes
 bgp listen range 10.1.1.0/24 peer−group Spokes
 network 192.168.0.0
 neighbor Spokes peer−group
 neighbor Spokes remote−as 65001
 neighbor 192.168.0.2 remote−as 65001
 neighbor 192.168.0.2 route−reflector−client
 neighbor 192.168.0.2 next−hop−self all
 neighbor 192.168.0.2 unsuppress−map ALL

access−list 1 permit any

route−map ALL permit 10
 match ip address 1
```

This configuration allows:

- Dynamic listener from addresses assigned to spokes
- Advertising network of ***192.168.0.0/24***
- Advertising summary route of ***192.168.0.0/16*** to all spokes. The aggregate−address configuration creates a static route for that prefix via null0 interface, which is a discard route that is used in order to prevent routing loops.
- Forwarding of specific prefixes to the other hub
- Route−reflector client to make sure that hubs exchange information learned from spokes between each other

This diagram represents the prefix exchange in BGP in this setup, from the perspective of one of the hubs.



Note: In this diagram, the green line represents information provided by spokes to the hub, the red line represents information provided by each hub to the spokes (a summary only), and the blue line represents the prefixes exchanged between hubs.

Network Summaries Use

Summaries might not be applicable or desired in some scenarios. Use caution when you designate the destination IP in prefixes, because iBGP does not override the next hop by default.

Summaries are recommended in networks that change state frequently. For example, unstable Internet connections might require summaries in order to: avoid the removal and addition of prefixes, limit the number of updates, and allow most setups to scale properly.

Spoke−to−Spoke Tunnels

In the scenario and configuration mentioned in the previous section, spokes on different hubs are not able to establish direct spoke−to−spoke tunnels. Traffic between spokes connected to different hubs flows over the central devices.

There is an easy workaround for this. However, it requires that Next Hop Resolution Protocol (NHRP) with same network−ID is enabled between hubs. This can be achieved, for example, if you create a point−to−point Generic Routing Encapsulation (GRE) tunnel between hubs. Then, IPsec is not required.

Verify

The Output Interpreter Tool (registered customers only) supports certain ***show*** commands. Use the Output Interpreter Tool in order to view an analysis of ***show*** command output.

The ***show crypto ikev2 sa*** command informs you about where the spoke is currently connected.

The ***show crypto ikev2 client flexvpn*** command allows an administrator to understand the current state of the FlexVPN client operation.

```
Spoke2# show crypto ikev2 client flexvpn

Profile : Flex_Client
 Current state:ACTIVE
 Peer : 172.25.1.1
 Source : Ethernet0/0
 ivrf : IP DEFAULT
 fvrf : IP DEFAULT
 Backup group: Default
 Tunnel interface : Tunnel1
 Assigned IP address: 10.1.1.111
```

A successful failover with the ***show logging*** configuration logs this output on the spoke device:

```
%CRYPTO−5−IKEV2_SESSION_STATUS: Crypto tunnel v2 is DOWN.  Peer 172.25.1.1:500
Id: 172.25.1.1
%FLEXVPN−6−FLEXVPN_CONNECTION_DOWN: FlexVPN(Flex_Client) Client_public_addr =
172.16.2.2 Server_public_addr = 172.25.1.1
%LINEPROTO−5−UPDOWN: Line protocol on Interface Tunnel1, changed state to down
%LINEPROTO−5−UPDOWN: Line protocol on Interface Tunnel1, changed state to up
%CRYPTO−5−IKEV2_SESSION_STATUS: Crypto tunnel v2 is UP.  Peer 172.25.2.1:500
Id: 172.25.2.1
%FLEXVPN−6−FLEXVPN_CONNECTION_UP: FlexVPN(Flex_Client) Client_public_addr =
172.16.2.2 Server_public_addr = 172.25.2.1 Assigned_Tunnel_v4_addr = 10.1.1.177
```

In this output, the spoke disconnects from ***hub 172.25.1.1***, the Flex_Client client configuration block detects failure and forces a connection to ***172.25.2.1*** where a tunnel comes up, and a spoke is assigned an IP of ***10.1.1.177***.

Troubleshoot

The Output Interpreter Tool (registered customers only) supports certain ***show*** commands. Use the Output Interpreter Tool in order to view an analysis of ***show*** command output.

Note: Refer to Important Information on Debug Commands before you use ***debug*** commands.

Here are the relevant debug commands:

- ***debug crypto ikev2***
- ***debug radius***

Related Information

- ***FlexVPN and Internet Key Exchange Version 2 Configuration Guide, Cisco IOS Release 15 M&T***
- ***Technical Support & Documentation − Cisco Systems***

Updated: Sep 16, 2013 Document ID: 116413