Revised: October 30, 2026

AutoVPN Overlay Load Balancing Across Multiple Uplinks for Cisco Secure Routers using Cisco Meraki Dashboard

Feature history of AutoVPN overlay load balancing

Table 1: Feature history: AutoVPN overlay load balancing

Feature	Release	Feature history
AutoVPN overlay load balancing	Cisco IOS XE 26.1.1 Release	From this release, Cisco 8000 Series Secure Routers can be configured for AutoVPN overlay load balancing feature using Cisco Meraki Dashboard.
Headless mode support	Cisco IOS XE 26.2.1 Release	Headless-mode resiliency maintains AutoVPN data-plane continuity when a device loses connectivity to the Cisco Meraki Dashboard or VPN Registry. In this release, headless-mode resiliency is supported through a CLI profile.

Active-active overlay load balancing

Active/Active Overlay allows multiple AutoVPN or IPsec tunnel paths to be active at the same time. VPN-destined traffic is distributed across eligible tunnel paths on a per-flow basis, so packets from one flow remain on the selected path while the eligible path set is stable. This avoids packet reordering and keeps backup paths established and usable.

Benefits of active-active overlay load balancing

- Keeps backup tunnel paths established and ready instead of waiting for a tunnel to be built after a failure.
- Supports equal-cost or weighted flow distribution when explicit uplink bandwidth is configured.

Prerequisites for active-active overlay load balancing

- The devices must have eligible local and remote uplinks for AutoVPN.
- Active-active overlay requires both multi-uplink-tunnels=true and load-balance=true. This is the default active-active configuration.
- For weighted distribution, configure explicit bandwidth under the uplink interface.
- Hub WAN uplinks must be reachable through their static public IP addresses and UDP ports.
- The spoke must have static public IP and port information for the resilient hub.
- Static hub information can be configured for one or more hubs and one or more uplinks per hub.
- Headless-mode resiliency is supported only through the CLI profile.

Restrictions for active-active overlay load balancing

- Traffic distribution is per flow, not per packet.
- The default source-IP or destination-IP hash can map multiple sessions between the same two hosts to the same tunnel.
- Weighted ratios are approximate and require many flows.

Headless-mode resiliency

Headless-mode resiliency maintains AutoVPN data-plane continuity when a device loses connectivity to the Cisco Meraki Dashboard or VPN Registry. Headless-mode resiliency is supported through a CLI profile.

Headless mode does not change Active/Active load-balancing behavior. Active/Active forwarding still requires multi-uplink-tunnels=true and load-balance=true.

Configure active-active overlay

Choose one of the options to configure active/active overlay:

- Active/active overlay configurations on Cisco Meraki Dashboard
- Active/active overlay configurations on Cisco Secure Routers

Configure active-active overlay on Cisco Meraki Dashboard

For information on enabling this feature on the Cisco Meraki Dashboard, refer to SD-WAN and Traffic Shaping.

Step 1 From the Cisco Meraki Dashboard, go to **Security & SD-WAN** > **Configure** > **SD-WAN & traffic shaping**.

Step 2 Enable Active-Active AutoVPN under global uplink preferences.

Configure active-active overlay on Cisco Secure Routers

Note

The router configurations are not manually set on the router, instead, all the configurations are automatically pushed from the Cisco Meraki cloud.

Active-active overlay requires both multi-uplink-tunnels and load-balance to be set to true.

When multi-uplink-tunnels=true and load-balance=true, multiple AutoVPN tunnels are created and eligible flows are distributed across the available tunnel paths.

If load-balance=false, the tunnels are still created, but only the primary tunnel carries traffic. The routes are installed with different metrics, so only the primary route is visible in the routing table.

AutoVPN tunnel configuration

```
crypto
 autovpn
  <autovpn-cfg>
   local <peer-num> <peer-id> <is-hub> true true <uplink0> <uplink1> ...
```

Equal-cost and weighted load balancing

Active/Active Overlay supports equal-cost and weighted load balancing.

By default, every eligible uplink has a weight of 1. Traffic is therefore distributed equally across the eligible uplinks.

To enable weighted load balancing, configure explicit bandwidth under an uplink interface:

```
hub(config)#interface <uplink-interface>
hub(config-if)#bandwidth <bandwidth-kbps>
```

When explicit bandwidth is configured, the bandwidth value is used to determine the relative load-balancing weight. Higher-bandwidth uplinks receive a larger share of eligible flows, while lower-bandwidth uplinks receive a smaller share.

Improve flow distribution with port-aware CEF hashing

When many sessions use the same source and destination IP addresses, enable port-aware hashing so CEF also includes TCP/UDP source and destination ports.

```
ip cef load-sharing algorithm include-ports source destination
```

Configure headless-mode resiliency

Configure resilient-hub on the spoke to specify the static IP address and UDP port for one or more hub uplinks.

For hub uplinks with configured static IP and port information, the spoke initiates IKEv2/IPsec tunnels using the configured values. For hub uplinks without configured static information, the spoke uses information learned from the VPN Registry.

When static information is configured, it takes priority over VPN Registry information for that hub uplink.

```
crypto autovpn <autovpn-cfg>
   remote <device-num> <device-id> <is-hub> <peer-psk> <multi-uplink-tunnels>
   resilient-hub <device-num> <uplink0-ip> <uplink0-port> [<uplink1-ip> <uplink1-port> <uplink2-ip>
<uplink2-port>]
```

Verify active-active overlay load balancing

Use these checks in order: map AutoVPN sessions to Virtual-Access interfaces, check packet counters on the VA adjacency, and optionally check interface and IPsec SA counters.

Map AutoVPN sessions to Virtual-Access interfaces

```
SPOKE#show platform software autovpn sessions
 SRC IP        DST IP      TUNNEL IF       UPLINK IF          WEIGHT
 10.1.11.11    10.1.9.9    Virtual-Access3 GigabitEthernet2   0
 10.1.22.22    10.1.9.9    Virtual-Access2 GigabitEthernet3   2000000
 10.1.22.22    10.1.10.10  Virtual-Access1 GigabitEthernet3   2000000
```

- TUNNEL IF identifies the Virtual-Access interface for each AutoVPN session.
- UPLINK IF identifies the physical WAN interface used by the tunnel.
- WEIGHT identifies the bandwidth or weight used for weighted distribution.
- Use the output to map Virtual-Access to uplink to peer path.

Check packets on the virtual access adjacency

```
hub#show adjacency Virtual-Access1 int | sec TAG
 TAG      Virtual-Access1                    point2point(4)
          0 packets, 0 bytes
          P2P-ADJ
          Next chain element:
            IP adj out of GigabitEthernet3, addr 10.1.22.5
            IPSec tunnel
            IP Tunnel stack to 10.1.10.10
            route-via enabled: GigabitEthernet3
```

- Packet and byte counters should increase during traffic.
- The IPSec tunnel entry confirms an encrypted tunnel adjacency.
- IP adj out of <uplink> confirms the physical uplink used.
- IP Tunnel stack to <peer> confirms the remote tunnel endpoint.

Optional counter checks

```
hub#show interface Virtual-Access1
```

Verify input/output packet and byte counters before and after traffic.

```
Device#show crypto ipsec sa
 interface: Virtual-Access1
  local addr 55.0.1.66
  current_peer 10.1.15.15 port 4500
  #pkts encaps: 2520
  #pkts encrypt: 2520
  #pkts decaps: 4705
  #pkts decrypt: 4705
```

Verify headless mode status

Verify the headless mode status:

```
spoke#show crypto autovpn session
Local Information:
  Device Number: 7001, ID: 700000000000000000000000000001, Role: Spoke
  Multi-Uplink : True, Load-Balance: False

  Local-WAN info learnt from VPN Registries:
   Local-WAN             Interface                    Private-IP:Port     NAT-Type
   ------------------------------------------------------------------------------------
   WAN-1                 GigabitEthernet1              1.1.1.2:4500       Friendly
   WAN-2                 GigabitEthernet2              2.2.2.2:4500       Friendly

  VPN Registry-1    10.126.71.240:9350
   Local-WAN        Status            Public-IP:Port
   ------------------------------------------------
   WAN-1             Connected      10.126.71.250:5065
   WAN-2             Connected      10.126.71.250:5064

  VPN Registry-2    10.126.71.241:9350
   Local-WAN        Status            Public-IP:Port
   ------------------------------------------------
   WAN-1          Disconnected      10.126.71.250:5065
   WAN-2          Disconnected      10.126.71.250:5064

Peers learnt from VPN Registries:

  Peer Number: 5001, ID: 500000000000000000000000000001, Role:Hub
  Peer-WAN  Priv-IP:port(From Reg1) Priv-IP:port(From Reg2) Pub-IP:port(From Reg1) Pub-IP:port(From Reg2)
  ------------------------------------------------------------------------------------------------------
  WAN-1           5.5.5.2:4500              5.5.5.2:4500     10.126.71.250:5062     10.126.71.250:5062
  WAN-2           6.6.6.2:4500              6.6.6.2:4500     10.126.71.250:5063     10.126.71.250:5063

Session Details:
  Peer-No        Local-IP:Port         Peer-IP:Port        Local-WAN# Peer-WAN#  Session-Type    State
Interface    Initiator
```

```
---------------------------------------------------------------------------------------------------------------------------------

 5001          1.1.1.2:4500          5.5.5.2:4500        1         1         Static        Connected
Vi1         Yes
 5001          1.1.1.2:4500          6.6.6.2:4500        1         2        Private        Connected
Vi2         Yes
 5001          2.2.2.2:4500          5.5.5.2:4500        2         1         Static        Connected
Vi3         Yes
 5001          2.2.2.2:4500          6.6.6.2:4500        2         2        Private        Connected
Vi4         Yes
```

Static session-type indicates that the session is in headless mode.