Understand Catalyst SD-WAN Tracker Usability and Use Cases

Contents

Introduction

This document describes Catalyst SD-WAN enterprise overlay networks, tracker usability and use cases.

Background Information

Catalyst SD-WAN enterprise overlay networks typically interact with a wide variety of external workloads, applications, and services. any of which can be located in the cloud, data-center/hubs, or remote branches. The SD-WAN control plane is responsible for advertising routes towards these services across the overlay in a scalable manner. In situations where critical applications and services become unreachable along a specific path, network operators typically must be able to detect these events and redirect user traffic to more appropriate paths to prevent indefinite traffic blackholing. To detect and rectify these types of network failures, the Catalyst SD-WAN control plane relies on trackers to monitor the health of external services and make routing changes as appropriate.

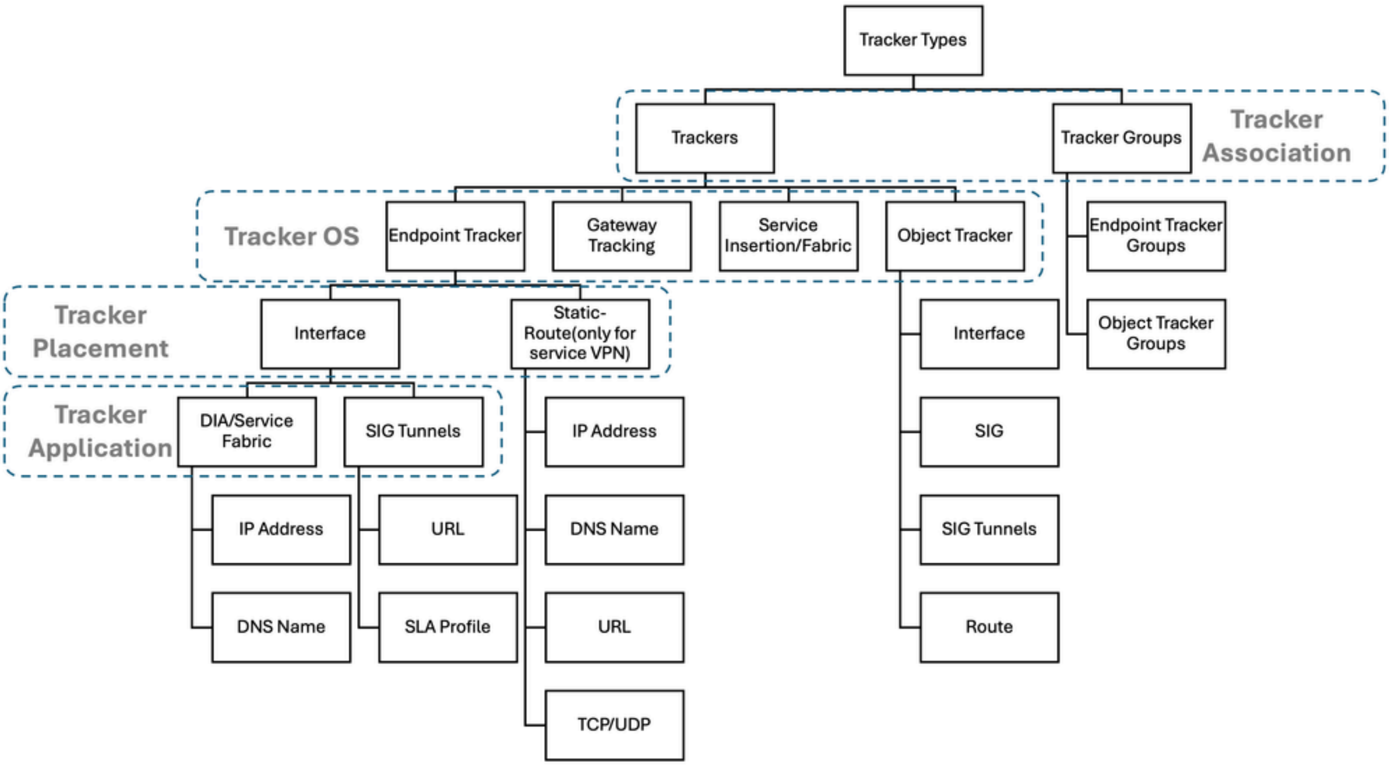
A tracker is a control plane reachability detection mechanism which sends probe packets toward a specific endpoint and notifies of reachability status changes (up or down) of the endpoint to interested modules. Trackers are designed as a scalable, high-level abstraction of the native Cisco IOS-XE® IP SLA feature, which can form a variety of probes (including HTTP, ICMP, and DNS). When a tracker notifies a client module of a status change, that module can take appropriate action to prevent traffic blackholing, such as installing or uninstalling a route or set of routes. The current applications of trackers within both SD-WAN and SD-Routing solutions include, but are not limited to: DIA (Direct Internet Access) trackers, SIG (Secure Internet Gateway) trackers, service trackers, static route trackers, tracker groups, and so on.

To build highly-available networks that are resilient to service failures, it is crucial to understand when to use each type of tracker configuration/model. The objective of this article is to explain where and how each type of tracker is used. The various trackers are addressed here, as well as the primary use case of each tracker, and the basic configuration workflows to implement each solution. Lastly, this article introduces a walk-through of general caveats involving trackers in Cisco IOS-XE®.

This article draws a distinction between the **endpoint-tracker** (SD-WAN and SD-Routing specific) and **object tracker** solutions (native IOS-XE), which address different use cases.

Types of Trackers

This chart provides a brief overview of all types of trackers available in the Cisco Catalyst SD-WAN solution:



From the preceding chart, there are four areas where trackers can be classified: **Tracker Association**, **Tracker OS**, **Tracker Placement** and **Tracker Application**. The next section describes each classification:

1. **Tracker Association**: This classification describes whether a tracker is a **single tracker** or a **tracker group**. Cisco Catalyst SD-WAN supports the usage of multiple trackers in a group (up to 2 at this time of writing) and the overall status of the tracker group is determined by a boolean AND or OR operator. Examples include an **endpoint-tracker** group or an **object tracker** group.
2. **Tracker OS**: This classification describes the Cisco IOS-XE® operating system or mode in which the tracker is supported. Cisco Catalyst IOS-XE routers support two operating modes:

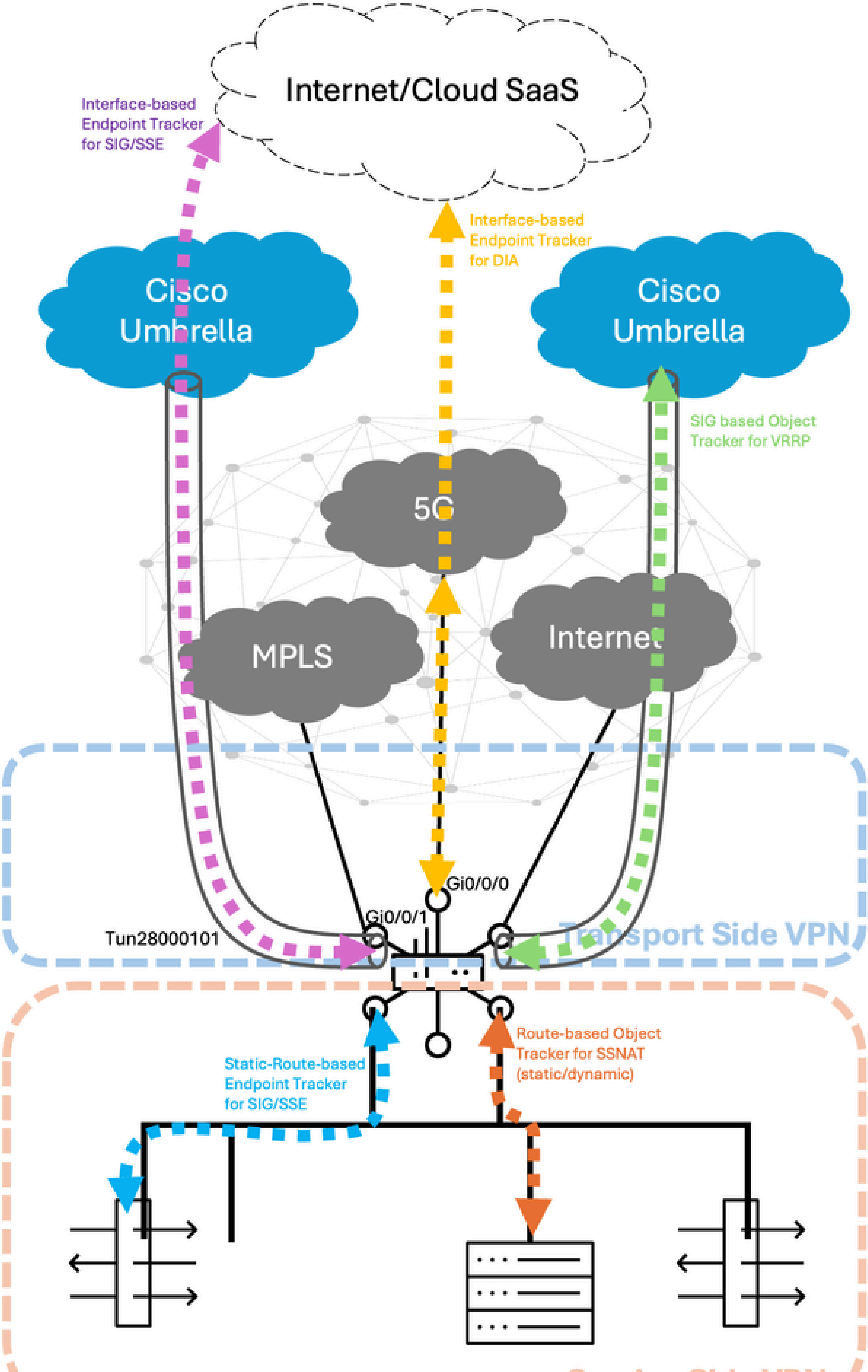
- **Autonomous mode** and
- **Controller mode**.

All **endpoint-tracker** and **gateway tracking** features are both intended for controller-mode (SD-WAN) use cases, whereas the **object tracker** is intended for autonomous-mode (SD-Routing) use cases.

3. **Tracker Placement**: This classification describes the location in which the tracker is configured. Currently, Cisco Catalyst SD-WAN supports applying trackers on either **interfaces**, **static routes**, or **services**.

4. **Tracker Application**: This classification describes the high-level **use cases** and **features** supported by Cisco Catalyst SD-WAN. While there are numerous areas of application of trackers, a few of them include: Direct Internet Access (DIA), Secure Internet Gateway (SIG), Secure Service Edge (SSE), Service-Side VPN tracking, and so on.

Here is a visual depiction of the tracker probe traffic across service/transport VPNs for several use cases on a Cisco Catalyst SD-WAN Edge (which can also be referred to as cEdge or vEdge):


Internet/Cloud SaaS
Interface-based Endpoint Tracker for SIG/SSE
Interface-based Endpoint Tracker for DIA
Cisco Umbrella
Cisco Umbrella
SIG based Object Tracker for VRRP
5G
MPLS
Internet
Gi0/0/0
Gi0/0/1
Tun28000101
Transport Side VPN
Static-Route-based Endpoint Tracker for SIG/SSE
Route-based Object Tracker for SSNAT (static/dynamic)

configured on SD-WAN Edge platforms on the transport-side VPN. By default, this is enabled under the basic system profile configurations (**Track Default Gateway**) under the Catalyst SD-WAN Manager. This helps to continuously monitor the next-hop address specified under each default static route in transport VPN, to ensure link/route failover, in the event of the reachability failure to the next-hop (which is also called the gateway, hence the name gateway-tracking). To learn more about gateway tracking, please visit the configuration guide.

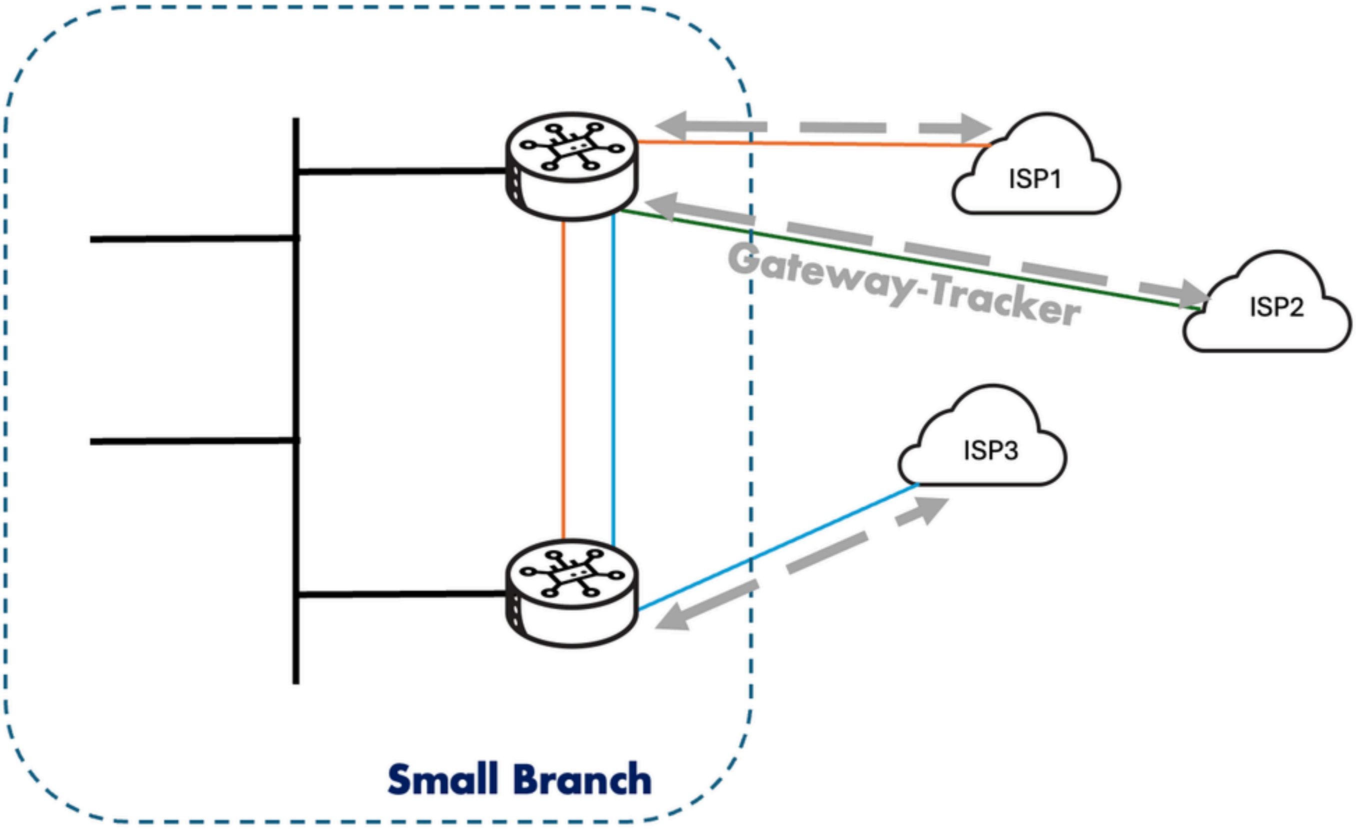
The type of probes used here are ARP-request unknown-unicast flooded packets. The intervals used are:

- Hello: 10 seconds
- Holdtime: 100 seconds
- Packet/Probe Type: ARP

Along with gateway tracking, also used is **tranport tracking** on SD-WAN Edges for checking the routed path between the local device and a Cisco Catalyst SD-WAN Validator. This is done by using ICMP probes at regular interval of 3s. This is configured using the "track-transport" keyword under the SD-WAN system configuration mode. This helps in regular monitoring of the DTLS connection to the Cisco Catalyst SD-WAN Validator from the respective WAN Edge. To learn more about transport tracking, please visit the configuration guide.

Use Cases

Gateway Tracking is a feature which is **implicitly configured by default** on SD-WAN for all static default routes that belong to the transport VPN or Global Routing Table (GRT). The usage of the feature does not always originate from the Manager template configuration standpoint, but can also evolve from received/acquired default static routes in the scenarios of using a DHCP server with options #3, #81 and so on.



Configuration

Applied by default in Cisco Catalyst SD-WAN:

```
!
system
 <snip>
 track-transport
 track-default-gateway
 <snip>
!
```

Verification

Here are ways to verify this as per Legacy configuration and Configuration group:

- **Configuration group: Configuration** > **Configuration Groups** > **System profile** > **Basic sub-profile** > **Track Settings section** > **Track Default Gateway (default:ON)**
- **Legacy configuration: Configuration** > **Templates** > **Feature Templates** > **System template** > **Advanced section** > **Gateway Tracking (default:ON)**

Service Insertion 1.0 and Service Fabric 2.0 Tracking

Service Insertion 1.0 Tracking was introduced in 20.3/17.3 release, and is a feature targeted toward ensuring the service address (or forwarding address) is reachable or available. This information helps the Edge to dynamically add or withdraw next-hop information from Control/Data policy. With the configuration of Service Insertion 1.0, the tracker (or tracking address) is enabled by default toward the service address. Based on this, the forwarding address and the service address is the same in 1.0. Even though service trackers are automatically configured with services, these trackers can be disabled using the **no track-enable** command, or by disabling the tracker knob in Configuration group/Legacy configuration. Since these are the only two possible operations (enable/disable) with trackers associated with services under Service Insertion 1.0, there are no further parameters that can be tweaked (such as threshold, multipler, interval). The type of probes used here are an ICMP Echo-request packets.

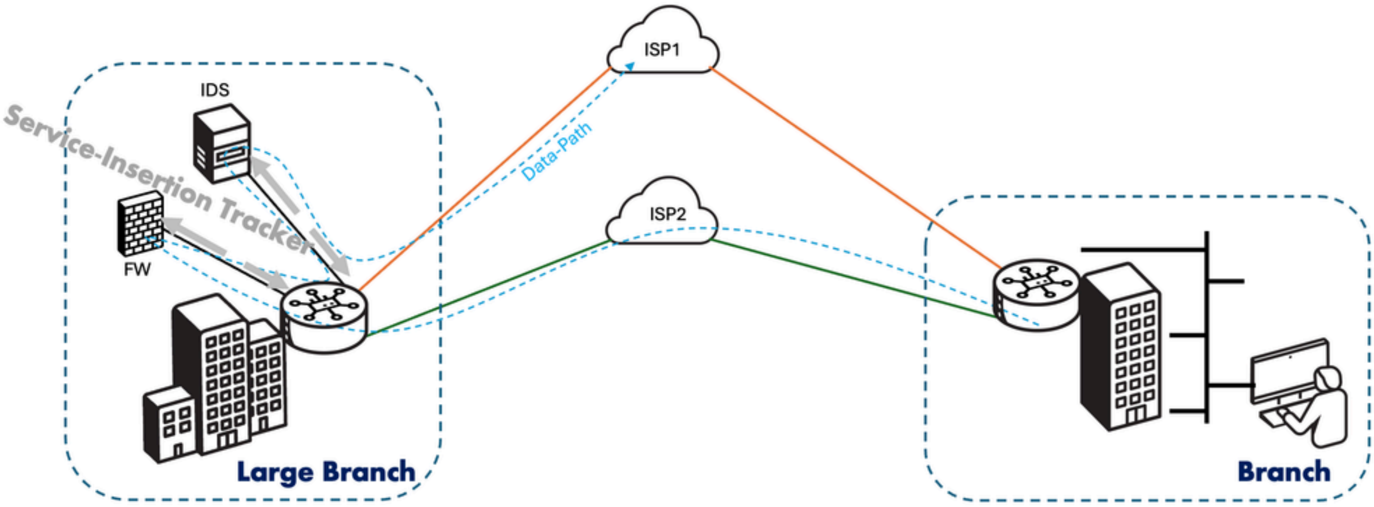
To learn more about service insertion 1.0 tracking, please visit the configuration guide. The default intervals used in service insertion 1.0 tracking are:

- Probe Interval: 5 probes every 60 seconds
- Multiplier: 5 times
- Packet/Probe Type: ICMP Echo/Echo-reply

Service Fabric 2.0 Tracking comes as part of **Service Insertion 2.0 feature** offering in Cisco Catalyst SD-WAN introduced from 20.13/17.13 release onwards. In this new variant of service insertion, the default method used by the configuration profiles and templates is still to have an implicit tracker pointing to each defined service address (or forwarding address) in **a service-HA-pair per rx/tx interface**. However, with Service Fabric 2.0, you can now split the forwarding address from the tracking address. This can be done simply by defining separate endpoint trackers for tracking a different endpoint address than the service address itself. This topic is expanded upon further in the next sections.

Use Cases

The primary use case for service trackers is for scalable monitoring of service reachability, particularly for service chaining. Service chaining can be deployed in a network that consists of multiple VPNs, where each VPN represents a different function or organization, to ensure that traffic between VPNs flows through a firewall. For example, in a large campus network, interdepartmental traffic can go through a firewall, while intradepartmental traffic can be routed directly. Service chaining can be seen in scenarios where an operator must satisfy regulatory compliance, such as Payment Card Industry Data Security Standard (PCI DSS), where PCI traffic must flow through firewalls in a centralized data center or regional hub:



Configuration

The configurations are the same as the normal workflow for setting up Service Insertion 1.0 on SD-WAN. The Service Insertion 1.0 Trackers would be enabled by default on all service addresses.

- **Configuration group: Configuration** > **Configuration Groups** > **Service Profile** > **Service VPN** > **Service** section:

1. Click on the **Add Service** button.

2. Choose a **Service Type**.

3. Enlist the **Service Address** (max of 4 possible, separated by a comma).

4. Verify that the Tracking knob is enabled (by default). This can be disabled, if required.

- **Legacy configuration: Configuration** > **Templates** > **Feature Templates** > **Cisco VPN (service)** > **Service** section:

1. Click on the **New Service** button

2. Choose a **Service Type**.

3. Enlist the **Service Address** (max of 4 possible, separated by a comma).

4. Verify that the Tracking knob is enabled (by default). This can be disabled, if required.

Note: The moment Step 3 is configured (from either Configuration group or Legacy configuration), the tracker automatically gets initiated to the various defined service addresses

From the CLI standpoint, the configuration for Service Insertion 1.0 appears like this:

```
!
sdwan
 service firewall vrf 1
  ipv4 address 10.10.1.4
!
```

Verification

The steps for verification extends to the similar steps followed as part of interface-based endpoint trackers used in the preceding sections.

There are two verification options of explicitly configured endpoint tracker.

- **On SD-WAN Manager: Monitor > Devices > {select Device-Name} > Applications > Tracker**:

Check under Individual Tracker and view the statistics of the tracker (Tracker Types, Status, Endpoint, Endpoint Type, VPN Index, Host Name, Round Trip Time) based on your configured Tracker Name.

- **On SD-WAN Manager: Monitor > Devices > {select Device-Name} > Events**:

In the case of flaps detected on the tracker, the respective logs would populate in this section with details such as **host name, attach point-name, tracker name, new state, address family**, and **vpn id**.

On CLI of the Edge:

```
Router#show endpoint-tracker
Interface                        Record Name            Status          Address Family   RTT in msecs
1:1:9:10.10.1.4                  1:10.10.1.4            Up              IPv4             1

Router#show endpoint-tracker records
Record Name                      Endpoint                              EndPoint Type Threshold(ms) Mult
1:10.10.1.4                      10.10.1.4                             IP            300           3

Router#show ip sla summary
IPSLAs Latest Operation Summary
Codes: * active, ^ inactive, ~ pending
All Stats are in milliseconds. Stats with u are in microseconds

ID           Type        Destination       Stats       Return      Last
                                                       Code        Run
-----------------------------------------------------------------------
*5           icmp-echo   10.10.1.4         RTT=1       OK          51 seconds ago
```

Interface Endpoint Trackers Used for DIA

NAT DIA Endpoint Trackers trackers are primarily designated for monitoring reachability of applications via a NAT DIA interface on SD-WAN Edge platforms.

For **Direct Internet Access (DIA)** use cases, NAT DIA trackers are primarily used to track the transport side interface and trigger a failover to either another available transport side interface or via SD-WAN overlay tunnels (using data-policy). This feature was introduced from 20.3/17.3 release onwards, and NAT fallback feature-option is available from 20.4/17.4 release. If the tracker determines that the local internet is unavailable via the NAT DIA interface, the router withdraws the NAT route from Service VPN, and reroutes the traffic based on the local routing configuration. The tracker continues to periodically check the status of the path to the interface. When it detects that the path is functioning again, the router reinstalls the NAT route to the internet. To learn more about DIA trackers, please visit the configuration guide.

In the tracker definition, you can choose to either give an IP address of an endpoint reachable via the NAT DIA interface (configured as "endpoint-ip") OR provide a Fully Qualified Domain Name (FQDN) to the endpoint (configured as "endpoint-dns-name").

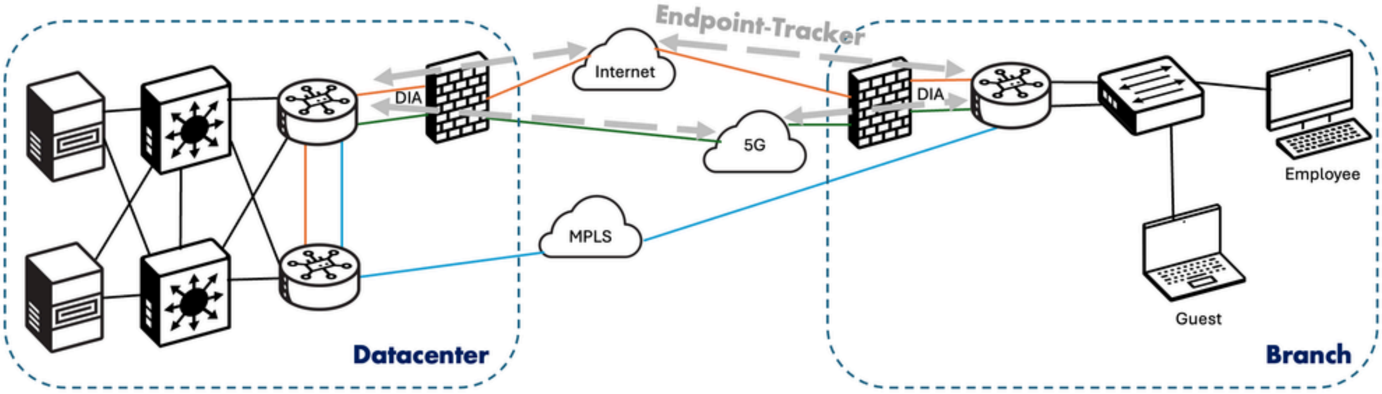
The type of probes used here are an HTTP request packet, very similar to an HTTP API request PDU stack. The intervals used are:

- Probe Interval: 60 seconds

- Multiplier: 180 seconds (since #retries is 3 = 3 x 60 seconds)
- Packet/Probe Type: HTTP

Use Cases

DIA is often deployed as an optimization at branch sites to avoid backhauling to a data center any branch traffic destined toward the Internet. Nonetheless, when DIA in branch sites is used, any lack of reachability along NAT DIA routes has to still fall back to alternate paths to avoid blackholding and loss of service. For sites that wish to use fallback to the DC (via SD-WAN overlay using NAT fallback) in the event of local DIA breakout failure. Leverage these interface based endpoint trackers on the DIA-capable interfaces on the branch-side edges to detect failures in order to initiate a failover to the backup/DC path. This way, high availability of Internet service is achieved with minimum disruption in the enterprise while still optimizing Internet traffic with DIA:



Configuration

These interface-based endpoint trackers have to be configured manually to enable this feature set. Here are the ways to configure it, depending on the type of configuration method preferred by the user.

- **Configuration group: Configuration > Configuration Groups > Transport & Management Profile > Ethernet Interface > Add Feature > Tracker**:

1. Define an endpoint tracker name.

2. Choose an endpoint tracker type (between HTTP-default and ICMP).

Note : ICMP endpoint tracker type was introduced since 20.13/17.13 release onwards.

3. Select the Endpoint (between Endpoint IP-default & Endpoint DNS Name).

Note: If Endpoint DNS Name is chosen, please ensure a valid DNS-server or nameserver is defined under the transport VPN/VRF using the Trasnport VPN configuration profile.

4. Enter the Address or DNS Name (FQDN) where the tracker probes have to be sent towards (the format depends on the previous step).

5. (Optional) You can choose to change the Probe Interval (default = 60 seconds) and Number of Retries (default = 3 times) to fasten the failure detection time.

- **Legacy configuration**:

Step 1. Definition of the Interface-Based Endpoint Tracker: **Configuration > Templates > Feature Templates > System template > Tracker** section:

1. Under Trackers sub-section, select the **New Endpoint Tracker** button.

2. Define an **endpoint tracker name**.

3. Choose the **Tracker Type** (between interface-default & static-route) as interface, since DIA use cases are of concern here.

4. Choose the **Endpoint Type** (between IP Address-default and DNS Name).

5. Enter the Endpoint IP Address or Endpoint DNS Name where the tracker probes must be sent towards (the format depends on the previous step).

6. (Optional) You can choose to change the Probe Threshold (default = 300 ms), Interval (default = 60 seconds) and Multipler (default = 3 times).

Step 2. Apply the Interface-Based Endpoint Tracker to an interface on the transport VPN: **Templates > Feature Templates > Cisco VPN Interface Ethernet > Advanced** section:

1. Enter the name of the **Endpoint Tracker** defined in the preceding **Step 1** into the **Tracker** field.

From the CLI standpoint, the configurations look like this:

```
(i) IP Address Endpoint :

!
endpoint-tracker t22
  tracker-type interface
  endpoint-ip 8.8.8.8
!
interface GigabitEthernet1
 <snip>
  endpoint-tracker t22
end
!

(ii) DNS Name Endpoint :

!
endpoint-tracker t44
tracker-type interface
endpoint-dns-name www.cisco.com
!
interface GigabitEthernet1
  <snip>
  endpoint-tracker t44
end
!
```

Verification

There are two verification options of explicitly configured endpoint trackers.

- On **SD-WAN Manager: Monitor > Devices > {select Device-Name} > Applications > Tracke**r:

Check under Individual Tracker and view the statistics of the tracker (**Tracker Types, Status, Endpoint, Endpoint Type, VPN Index, Host Name, Round Trip Time**) based on your configured **Tracker Name**.

- On **SD-WAN Manager: Monitor > Devices > {select Device-Name} > Events**:

In the case of flaps detected on the tracker, the respective logs would populate in this section with details such as **host name, attach point-name, tracker name, new state, address family**, and **vpn id**.

On CLI of the Edge:

```
Router#show endpoint-tracker interface GigabitEthernet1
Interface                        Record Name            Status          Address Family   RTT in msecs
GigabitEthernet1             t22                    Up              IPv4             2               2

Router#sh ip sla sum
IPSLAs Latest Operation Summary
Codes: * active, ^ inactive, ~ pending
All Stats are in milliseconds. Stats with u are in microseconds

ID           Type        Destination       Stats       Return      Last
                                                       Code        Run
-----------------------------------------------------------------------
*2           http        8.8.8.8           RTT=4       OK          56 seconds ag
                                                                   o

Router#show endpoint-tracker records
Record Name                      Endpoint                              EndPoint Type Threshold(ms) Mult
t22                              8.8.8.8                               IP            300           3
t44                        www.cisco.com                            DNS_NAME         300           3
```

Interface Endpoint Trackers Used for SIG Tunnels/SSE

When endpoint trackers are being used for SIG Tunnel/SSE use cases, it primarily indicates that the enterprise is looking for a cloud-based security stack offering which is easily made available nowadays using the help of Secure Internet Gateway (SIG) or Secure Service Edge (SSE) providers, such as Cisco, Cloudflare, Netskope, ZScalar and so on. Both SIG Tunnels and SSE come as part of the cloud security deployment model, wherein the branch uses the cloud to deliver the necessary security solutions it needs. The SIG Tunnels use case was the very initial offering of integrating Cisco Catalyst SD-WAN with such SIG providers (from 20.4/17.4 release), however with the evolution of cloud-delivered security offerings - the SSE use case was introduced (from 20.13/17.13 release) for covering use cases with providers such as Cisco (via Cisco Secure Access) and ZScalar.

IT requires a reliable and explicit approach to protect and connect with agility. It is now common to provide remote employees direct access to cloud applications, such as Microsoft 365 and Salesforce with additional security. The demand for cloud-delivered networking and security expands daily as contractors, partners, Internet of Things (IoT) devices, and so on, require network access. The convergence of network and security functions closer to end devices, at the cloud edge, is known as a service model called Cisco SASE. Cisco SASE combines cloud delivered networking and security functions to provide secure access to applications for all users or devices, from anywhere and at any time. Secure Service Edge (SSE) is a network security approach that helps organizations improve the security posture of their work environment while reducing complexity for end users and IT departments. To learn more about SIG Tunnel/SSE trackers, please visit the configuration guide.

Use Cases

Such interface-based endpoint trackers are used in such SIG Tunnel/SSE use cases, wherein you want to keep track of a well-known SaaS application URL endpoint or a specific URL endpoint of concern. Nowadays, SSE is the more commonly used scenario eversince the SASE architecture was split into SSE core functionalities and SD-WAN functionalities. You then want to choose between active & standby roles within the IPSec Tunnels created from a site (in this case, the DC). The user gets the choice to attach the tracker under the respective tunnel interface.

In the case of SSE providers, such as Cisco Secure Access (by Cisco) - an implicit endpoint tracker get used which is configured by default. However, the user does have a choice of creating a custom endpoint tracker and getting that attached to the IPSec Tunnel interface. The parameters of the default/implicit endpoint tracker used in SSE are:

For Cisco SSE:

Tracker Name: DefaultTracker

Endpoint being tracked: http://service.sig.umbrella.com

Endpoint Type: API_URL

Threshold: 300 ms

Multiplie: 3

Interval: 60 sec

For ZScaler SSE:

Tracker Name: DefaultTracker

Endpoint being tracked: http://gateway.zscalerthree.net/vpnte

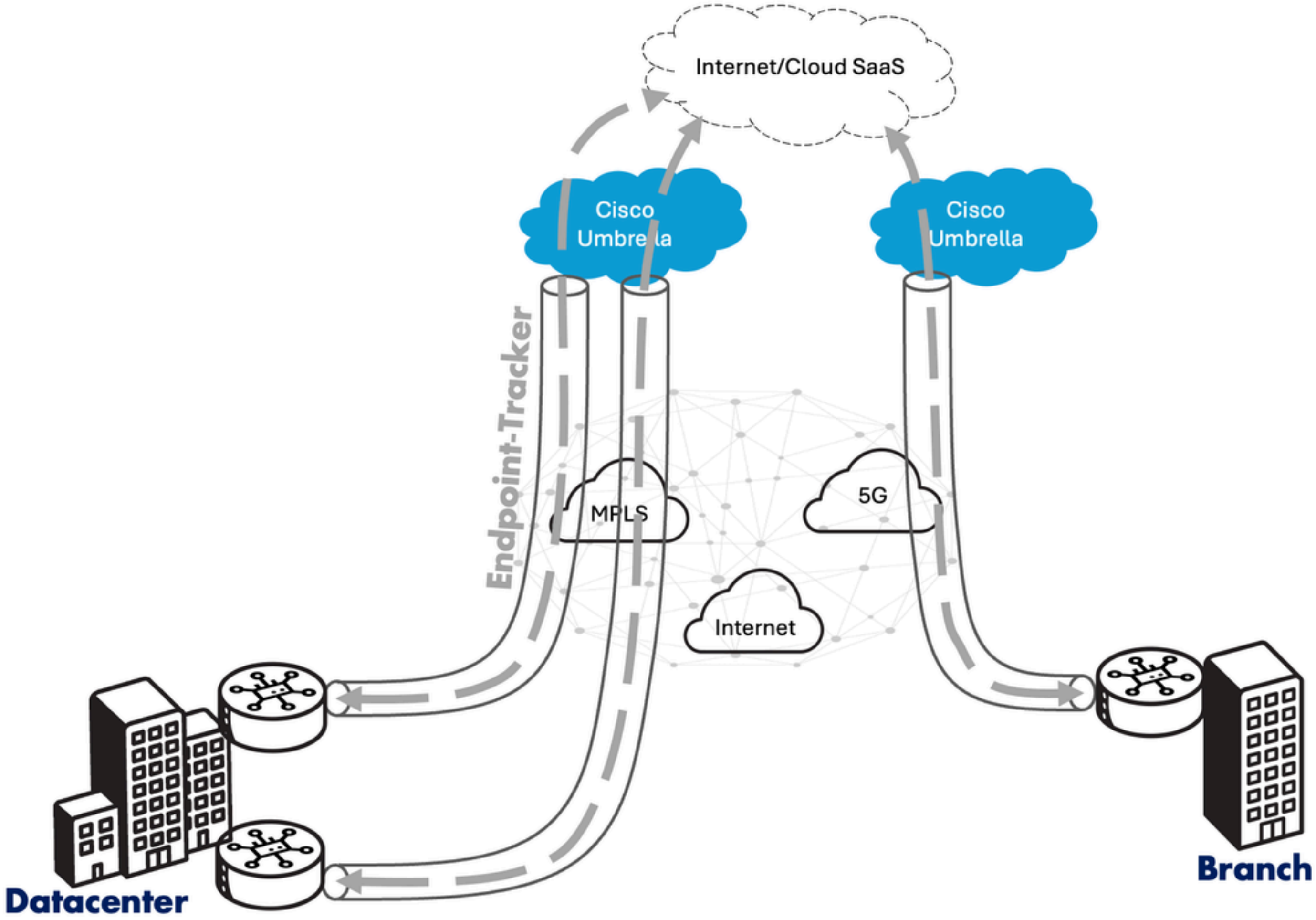
Endpoint Typ: API_URL

Threshold: 300 ms

Multiplier: 3

Interval: 60 sec

In the case of SIG Tunnels, there is no default/implicit endpoint tracker defined. Hence, the user has to manually configure an interface-based endpoint tracker in the event they want to track the IPSec Tunnel interface towards the SIG provider-cloud:



Configuration

In the case of SSE providers, the user does not have to define any endpoint tracker explicitly (unless desired). However, the workflows are different based on the type of configuration.

As a pre-requisite, you have to define the SIG/SSE credentials **Administration** > **Settings** > **External Services** > **Cloud Credentials**:

1. Under **Cloud Provider Credentials**, toggle the option of **Umbrella** or **Cisco SSE** (or both).

2. Define the parameters, such as **Organization ID, API Key, Secret**).

Set configuration group **Configuration** > **Policy Groups** > **Secure Internet Gateway/Secure Service Edge**:

1. Click on **Add Secure Internet Gateway** or **Add Secure Service Edge**.

2. Define a name and description.

3. Select one of the radio buttons under SIG/SSE Provider (either **Umbrella** or **Cisco SSE**).

4. Under the **Tracker** section, define the source IP address that is used to source the tracker probes.

5. If you choose to define an explicit/custom endpoint tracker, then click **Add Tracker**, then fill in the parameters for the endpoint tracker (**Name, API URL of Endpoint, Threshold, Probe Interval**, and **Multiplier**).

6. Under the **Configuration** section, create the tunnel interfaces wherein you can define the parameters (such as **Interface Name, Description, Tracker, Tunnel Source Interface, Datacenter Primary/Secondary**).

> **Note**: In Step 6 is where the user is given the option of attaching the defined endpoint tracker to the respective IPSec Tunnel. Please note this is an optional field.

7. Under the **High Availability** section, create an Interface Pair and define your Active Interface and Backup Interface along with their respective weights. Then apply the preceding configured policy group to the relevant edges.

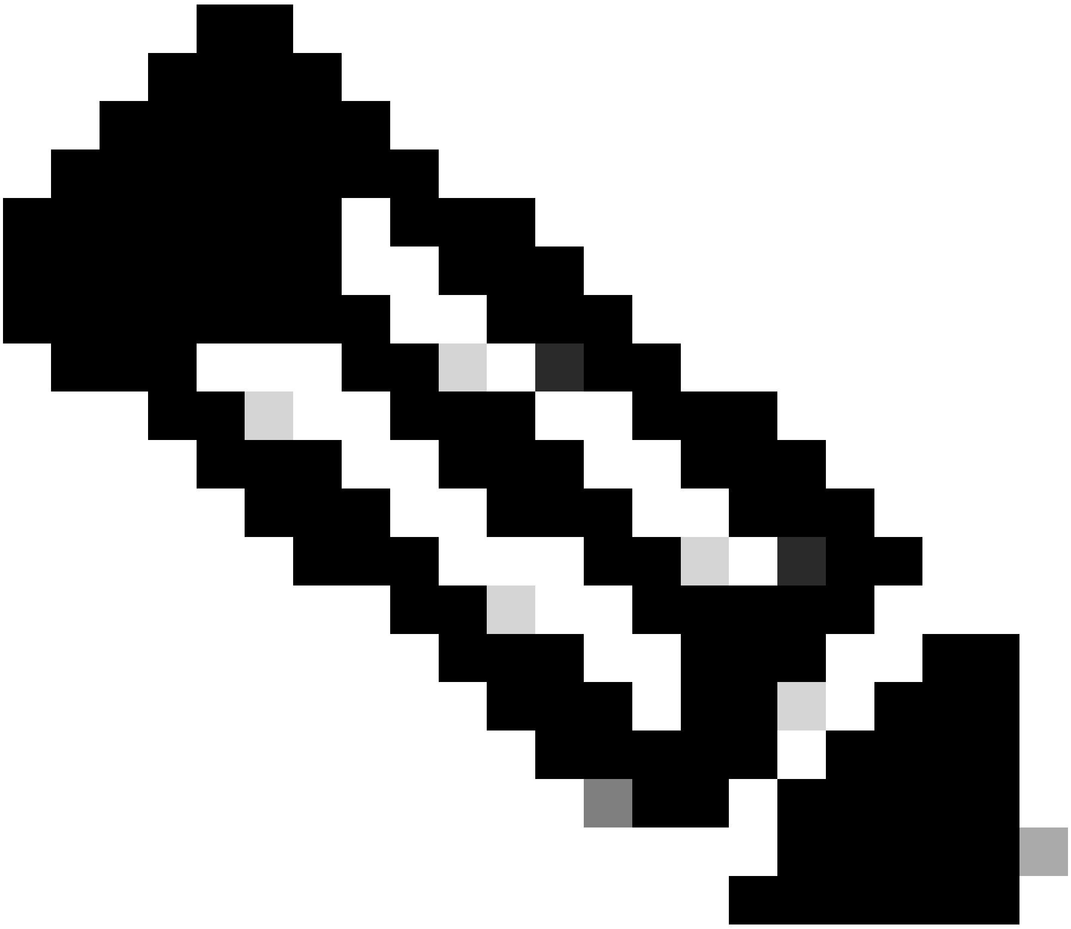
Set legacy configuration **Configuration > Templates > Feature Templates > Cisco Secure Internet Gateway** feature template:

1. Select one of the radio buttons under SIG Provider (either Umbrella, ZScalar or Generic).

2. Under the Tracker (BETA) section, define the source IP address that is used to source the tracker probes.

5. If you choose to define an explicit/custom endpoint tracker, then click on New Tracker, and fill in the parameters for the endpoint tracker (**Name, API url of endpoint, Threshold, Interval**, and **Multiplier**).

6. Under the **Configuration** section, create the tunnel interfaces (by clicking **Add Tunnel**) wherein you can define the parameters (such as **Interface Name, Description, Tracker, Tunnel Source Interface, Datacenter Primary/Secondary**).



> **Note**: In step 6 is where the user is given the option of attaching the defined endpoint tracker to the respective IPSec Tunnel. Please note this is an optional field.

7. Under the **High Availability** section, define your Active Interface and Backup Interface along with their respective weights.

From the CLI standpoint, the configurations look like this:

```
(i) For the default interface-based endpoint tracker applied with SSE
!
endpoint-tracker DefaultTracker
 tracker-type     interface
 endpoint-api-url http://service.sig.umbrella.com
!
interface Tunnel16000101
```

```
 description auto primary-dc
 ip unnumbered GigabitEthernet1
 ip mtu 1400
 endpoint-tracker DefaultTracker
 <snip>
end
!

(ii) For the custom interface-based endpoint tracker (can be applied in SIG & SSE use-cases)
!
endpoint-tracker cisco-tracker
 tracker-type     interface
 endpoint-api-url http://www.cisco.com
!
interface Tunnel16000612
 ip unnumbered GigabitEthernet1
 ip mtu 1400
 endpoint-tracker cisco-tracker
 <snip>
end
!
```

Verification

There are verification options of explicitly configured endpoint trackers.

- On **SD-WAN Manager: Monitor** > **Devices** > **{select Device-Name}** > **Applications** > **Tracker**:

Check under Individual Tracker and view the statistics of the tracker (Tracker Types, Status, Endpoint, Endpoint Type, VPN Index, Host Name, Round Trip Time) based on your configured Tracker Name.

- On **SD-WAN Manager: Monitor** > **Devices** > **{select Device-Name}** > **Events**:

In the case of flaps detected on the tracker, the respective logs would populate in this section with details such as **host name, attach point-name, tracker name, new state, address family**, and **vpn id**.

On CLI of the Edge:

```
Router#show endpoint-tracker interface Tunnel16000612
Interface                        Record Name            Status          Address Family   RTT in msecs
t Hop
Tunnel16000612               cisco-tracker          Up              IPv4             26            31

Router#show endpoint-tracker interface Tunnel16000101
Interface                        Record Name            Status          Address Family   RTT in msecs
t Hop
Tunnel16000101               DefaultTracker         Up              IPv4             1             10

Router#show endpoint-tracker records
Record Name                      Endpoint                              EndPoint Type Threshold(ms) Mult
s) Tracker-Type
DefaultTracker                   http://gateway.zscalerthree.net/vpnte API_URL       300           3
   interface
cisco-tracker                    http://www.cisco.com                  API_URL       300           3
   interface
```

Interface Endpoint Trackers Used for Service Fabric 2.0

Service Fabric 2.0 Tracking which was introduced in 20.13/17.13 release, is an enhanced variant of the service insertion 1.0 tracking - wherein users get the ability to customize the trackers to a larger extent. The default behavior is retained from the previous version of Service Insertion (1.0), a tracker would get initiated by default with the definition of each service address (or forwarding address) in a service-HA-pair per rx/tx. But with Service Insertion 2.0, the tracking address (IP/endpoint to the tracked) can be separated from the forwarding address (usually the service address). This is done using custom endpoint trackers defined on the VPN level. To learn more about Service Fabric 2.0 trackers, please visit the configuration guide.

If the user chooses to use the default tracker, the specifications of the tracker probes are:

- Hello: 1 probe every 30 seconds
- Multiplier: 3 times
- Packet/Probe Type: ICMP Echo/Echo-reply

If the user chooses to use a custom tracker, then the specifications of the tracker probes are:

- Hello: 1 probe every 60 seconds
- Multiplier: 3 times
- Packet/Probe Type: ICMP Echo-request/reply

Use Cases

Use cases of Service Insertion 1.0 mentioned in the preceding sections apply here as well.

Configuration

There is support for workflow based configuration for Service Insertion 2.0, which is a wizard-guided approach, helping to simplify the user-experience, while adhering to the standard Configuration group workflow steps.

1. Define the Service Chain - Configuration group under the **Configuration** > **Service Insertion** > **Service Chain Definitions** section:

a. Click the **Add Service Chain Definition** button.

b. Fill in the details of the **Name** and **Description of the Service**.

c. Fill in a list format (by selecting from the drop-down), the **Service Type**.

2. Define the Service Chain Instance - Configuration group under the **Configuration** > **Service Insertion** > **Service Chain Configurations** section:

a. Click **Add Service Chain Configuration**.

b. In the Service Chain Definition step, select the radio button titled **Select Existing**, and choose the previously defined Service.

c. Provide a Name and Description for the **Start Service Chain Configuration** step.

d. In the Service Chain Configuration for Manually Connected Services step, select the **Service Chain VPN-ID**.

e. Then, for each defined service in the service chain instance (represented in sub-tabs), under service details, provide the Type of attachment (IPv4, IPv6 or Tunnel Connected).

f. Select the **Advanced** checkbox. If you need to have active-backup/HA use cases (also enable the Add parameters for Backup knob) or even if you need to define a custom endpoint tracker (also enable the Custom Tracker knob).

g. If you have scenarios wherein the egress (tx) traffic goes to the service via one interface and the return traffic from the service is ingressed (rx) via another interface, turn on the **Traffic from service is received on a different interface** knob.

h. With the **Advanced** and **Custom Tracker** knobs enabled, define the Service IPv4 Address (forwarding address), the SD-WAN Router Interface (to which the service is connected) and the Tracker Endpoint (tracking address). You can also modify the custom tracker parameters such as interval and multiplier (by clicking the edit button).

i. Repeat steps (e), (f), (g) and (h) for each subsequently defined service.

3. Attach the **Service Chain Instance** to the configuration Profile of the Edge - Configuration group under the **Configuration > Configuration Groups > Service Profile > Service VPN > Add Feature > Service Chain Attachment Gateway**:

a. Provide a Name and Description to this Service Chain Attachment Gateway parcel.

b. Select the previously defined Service Chain Definition (in step 1).

c. Re-add/verify the details as performed in step 2. For tracker definition, the only difference from the preceding step 2 is that you get a chance to give a Tracker Name and also select the Tracker Type (from service-icmp to ipv6-service-icmp).

From the CLI standpoint, the configurations look like this:

```
!
endpoint-tracker tracker-service
 tracker-type service-icmp
 endpoint-ip  10.10.1.4
!
service-chain SC1
 service-chain-description FW-Insertion-Service-1
 service-chain-vrf 1
 service firewall
  sequence 1
  service-transport-ha-pair 1
   active
    tx ipv4 10.10.1.4 GigabitEthernet3 endpoint-tracker tracker-service
!
```

Verification

- On SD-WAN Manager **Monitor > Devices > {select Device-Name} > Applications > Tracker**:

Check under Individual Tracker and view the statistics of the tracker (Tracker Types, Status, Endpoint, Endpoint Type, VPN Index, Host Name, Round Trip Time) based on your configured Tracker Name.

- On SD-WAN Manager **Monitor** > **Devices** > **{select Device-Name}** > **Events**:

In the case of flaps detected on the tracker, the respective logs would populate in this section with details such as **host name, attach point-name, tracker name, new state, address family**, and **vpn id**.

On CLI of the Edge:

```
Router#show endpoint-tracker
Interface                        Record Name            Status          Address Family   RTT in msecs
1:101:9:tracker-service          tracker-service        Up              IPv4             10

Router#show endpoint-tracker records
Record Name                      Endpoint                              EndPoint Type Threshold(ms) Mult
tracker-service                  10.10.1.4                             IP            300           3

Router#show ip sla summary
IPSLAs Latest Operation Summary
Codes: * active, ^ inactive, ~ pending
All Stats are in milliseconds. Stats with u are in microseconds

ID           Type        Destination       Stats       Return      Last
                                                       Code        Run
-----------------------------------------------------------------------
*6           icmp-echo   10.10.1.4         RTT=1       OK          53 seconds ago

Router#show platform software sdwan service-chain database

Service Chain: SC1
   vrf: 1
   label: 1005
   state: up
   description: FW-Insertion-Service-1

   service: FW
      sequence: 1
      track-enable: true
      state: up
      ha_pair: 1
         type: ipv4
         posture: trusted
         active: [current]
            tx: GigabitEthernet3, 10.10.1.4
                endpoint-tracker: tracker-service
                state: up
            rx: GigabitEthernet3, 10.10.1.4
                endpoint-tracker: tracker-service
                state: up
```

Static-Route Endpoint Trackers Used for Static Route Tracking (Service-Side)

The second type of endpoint trackers are termed as static-route-based endpoint trackers. As the name itself indicates, these types of trackers are used primarily to track the next-hop address of any static route defined under the Service-Side VPN. By default, all "connected" and "static" route types are advertised into OMP protocol - post which all remote sites that contain the respective service VPN becomes aware of that destination prefix (wherein the next-hop point to the TLOC of the originating site). The originating site is the site where the specific static route was initiated from.

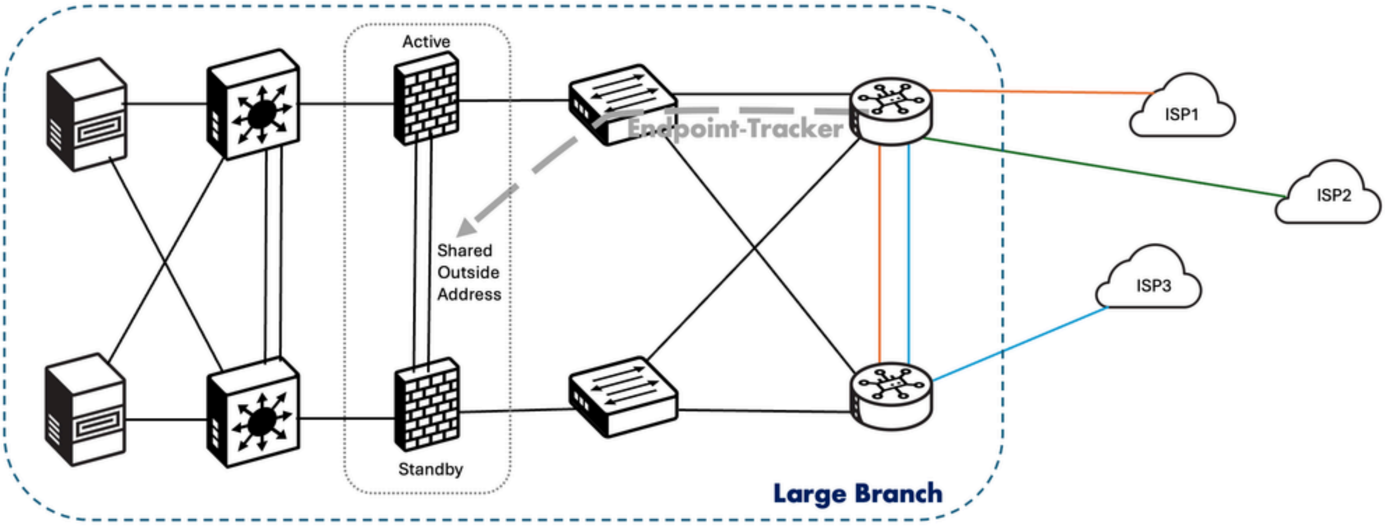
However, in the event the next-hop address in the static route becomes unreachable, the route does not stop getting advertised into OMP. This would cause issues of traffic getting blackholed for flows destined towards the originating site. This brings about the need for the need to attaching a tracker to the static route, to ensure advertisement of the static route into OMP ONLY when the next-hop address is reachable. This feature was introduced in 20.3/17.3 release for basic IP address type static-route-based endpoint trackers. From 20.7/17.7 release onwards, support was added for sending tracker probes to only particular TCP or UDP ports of the next-hop IP address (in cases while using firewalls to only open certain ports for tracking purposes). To learn more about Static Route trackers, please visit the configuration guide.

The type of probes used here are a simple ICMP echo-request packet. The intervals used are:

- Hello: 60 seconds
- Holdtime: 180 seconds (since #retries is 3 = 3 x 60 seconds)
- Packet/Probe Type: ICMP Echo/Echo-reply

Use Cases

This type of static-route-based endpoint trackers are used for service side tracking of next-hop addresses in static routes. One such common scenario would be of tracking the LAN side next-hop address corresponding to a pair of active/standby firewalls, which share the Outside IP address-based on which Outside interface is playing the role of "active" firewall. In cases where firewall rules seem to be highly constrictive, wherein only certain ports are opened for use case based purposes, the static route tracker can be used to track the specific TCP/UDP port to the next-hop IP address that belongs on the LAN-side firewall Outside interface.



Configuration

These static-route-based endpoint trackers have to be configured manually to enable this feature-set. Here are the ways to configure it, depending on the type of configuration method preferred by the user.

- Configuration group **Configuration > Configuration Groups > Service Profile > Service VPN > Add Feature > Tracker**:

1. Provide a Name, Description & Tracker Name for the new (endpoint) tracker being defined.

2. Choose the endpoint type, depending on whether you need to only track the next-hop IP address (choose **Address** radio button), or even specific TCP/UDP ports (choose **Protocol** radio button).

3. Enter the Address, in an IP address format. Also enter the protocol (TCP or UDP) and the port number, in case you chose **Protocol** as your endpoint type in the previous step.

4. You can change the default values given for **Probe Interval, Number of Retries**, and **Latency Limit**, if required.

- **Configuration > Configuration Groups > Service Profile > Service VPN > Route** section:

1. Select **Add IPv4/IPv6 Static Route** button.

2. Fill in the details, such as Network Address, Subnet Mask, Next-Hop, Address, AD.

3. Click on **Add Next Hop With Tracker** button.

4. Re-enter the Next-Hop address, AD and choose from the drop-down the previous created (endpoint) tracker name.

- Legacy configuration **Configuration > Templates > Feature Templates > System Template > Tracker** section:

1. Select New Endpoint Tracker button.

2. Provide a Name for the new (endpoint) tracker being defined.

3. Change the **Tracker Type** radio button to **static-route**.

4. Choose the endpoint type, as next-hop IP address (choose IP Address radio button).

5. Enter the Endpoint IP, in an IP address format.

6. You can change the default values given for **Probe Interval, Number of Retries**, and **Latency Limit**, if required.

- **Configuration > Templates > Feature Templates > Cisco VPN (service-side ONLY) > IPv4/IPv6 Route** section:

1. Selec**t New IPv4/IPv6 Route** button.

2. Fill in the details, such as Prefix, Gateway.

3. Click the **Add Next Hop With Tracker** button.

4. Re-enter the Next-Hop Address, AD (Distance) and manually enter the previous created (endpoint) tracker name.

From the CLI standpoint, the configurations look like this:

```
(i) For the static-route-based endpoint tracker being used with IP address :
!
endpoint-tracker nh10.10.1.4-s10.20.1.0
 tracker-type static-route
 endpoint-ip  10.10.1.4
!
```

```
track nh10.10.1.4-s10.20.1.0 endpoint-tracker
!
ip route vrf 1 10.20.1.0 255.255.255.0 10.10.1.4 track name nh10.10.1.4-s10.20.1.0
!

(ii) For the static-route-based endpoint tracker being used with IP address along with TCP/UDP port :
!
endpoint-tracker nh10.10.1.4-s10.20.1.0-tcp-8484
 tracker-type static-route
 endpoint-ip  10.10.1.4 tcp 8484
!
track nh10.10.1.4-s10.20.1.0-tcp-8484 endpoint-tracker
!
ip route vrf 1 10.20.1.0 255.255.255.0 10.10.1.4 track name nh10.10.1.4-s10.20.1.0-tcp-8484
!
```

Verification

There are two areas of verification of explicitly configured endpoint trackers.

- On **SD-WAN Manager Monitor** > **Devices** > **{select Device-Name}** > **Real Time**:

1. Under **Device Options**, type in "Endpoint Tracker Info."

2. Check under **Individual Tracker (Attach Point Name)** and view the statistics of the tracker (**Tracker State, Associated Tracker Record Name, Latency in mx from device to the endpoint, Last Updated timestamp**) based on your configured **Tracker Name.**

- On **SD-WAN Manager Monitor** > **Devices** > **{select Device-Name}** > **Events**:

In the case of flaps detected on the tracker, the respective logs would populate in this section with details such as **host name, attach point-name, tracker name, new state, address family**, and **vpn id**.

On CLI of the Edge:

```
Router#sh endpoint-tracker static-route
Tracker Name                   Status          RTT in msec     Probe ID
nh10.10.1.4-s10.20.1.0         UP              1               3

Router#show track endpoint-tracker
Track nh10.10.1.4-s10.20.1.0
  Ep_tracker-object
  State is Up
    2 changes, last change 00:01:54, by Undefined
  Tracked by:
    Static IP Routing 0

Router#sh endpoint-tracker records
Record Name                      Endpoint                              EndPoint Type Threshold(ms) Mult
nh10.10.1.4-s10.20.1.0           10.10.1.4                             IP            300           3

Router#sh ip sla summ
IPSLAs Latest Operation Summary
Codes: * active, ^ inactive, ~ pending
All Stats are in milliseconds. Stats with u are in microseconds
```

```
ID           Type        Destination       Stats       Return      Last
                                                       Code        Run
-----------------------------------------------------------------------
*3           icmp-echo   10.10.1.4         RTT=1       OK          58 seconds ago

EFT-BR-11#sh ip static route vrf 1
Codes: M - Manual static, A - AAA download, N - IP NAT, D - DHCP,
       G - GPRS, V - Crypto VPN, C - CASA, P - Channel interface processor,
       B - BootP, S - Service selection gateway
       DN - Default Network, T - Tracking object
       L - TL1, E - OER, I - iEdge
       D1 - Dot1x Vlan Network, K - MWAM Route
       PP - PPP default route, MR - MRIPv6, SS - SSLVPN
       H - IPe Host, ID - IPe Domain Broadcast
       U - User GPRS, TE - MPLS Traffic-eng, LI - LIIN
       IR - ICMP Redirect, Vx - VXLAN static route
       LT - Cellular LTE, Ev - L2EVPN static route
Codes in []: A - active, N - non-active, B - BFD-tracked, D - Not Tracked, P - permanent, -T Default Tr


Codes in (): UP - up, DN - Down, AD-DN - Admin-Down, DL - Deleted
Static local RIB for 1

M  10.20.1.0/24 [1/0] via 10.10.1.4 [A]
T               [1/0] via 10.10.1.4 [A]
```

Interface Object Trackers Used for VRRP Tracking

Object Trackers are trackers designed for autonomous mode consumption (use cases). These trackers having use cases varying from VRRP-based interface/tunnel tracking to service-VPN NAT tracking.

For VRRP tracking use cases, the VRRP state is determined based on the tunnel link status. If the tunnel or interface is down on the primary VRRP, the traffic is directed to the secondary VRRP. The secondary VRRP router in the LAN segment becomes primary VRRP to provide gateway for the service-side traffic. This use case is only applicable for service-VPN, and helps for failing over the VRRP role on the LAN side in the event of failure on the SD-WAN overlay (Interface or Tunnels in the case of SSE). For attaching trackers to VRRP groups, ONLY object trackers can be used (not endpoint trackers). This feature was introduced from 20.7/17.7 release for Cisco Catalyst SD-WAN Edges.

There are no probes used here by the tracker. Instead, it uses the line-protocol state to decide on the tracker state (up/down). There are no reaction intervals in interface line-protocol based trackers - the moment the interface/tunnel line-protocol goes DOWN, the track state is also brought to the DOWN state. Then depending on the action of either shutdown or decrement, the VRRP group would accordingly re-converge. To learn more about VRRP Interface trackers, please visit the configuration guide.

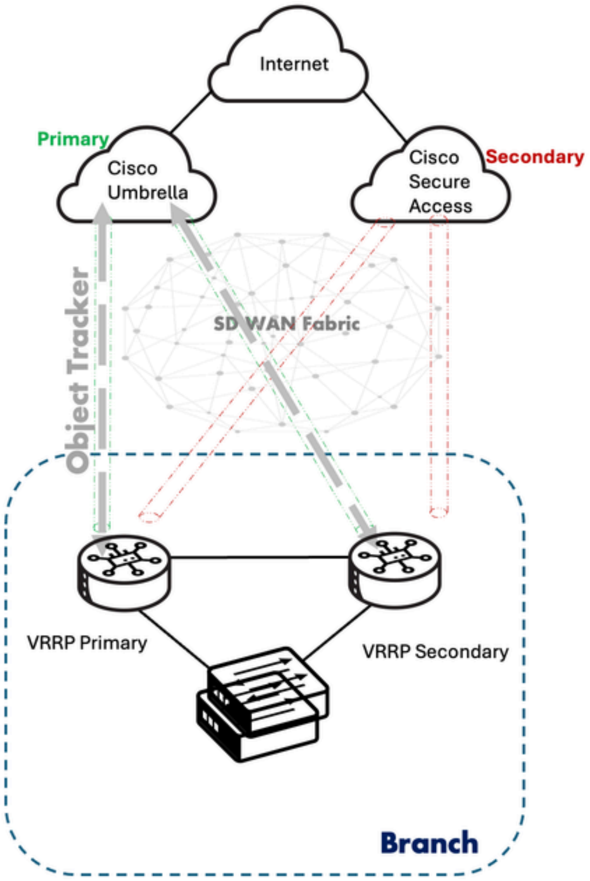
Use Cases

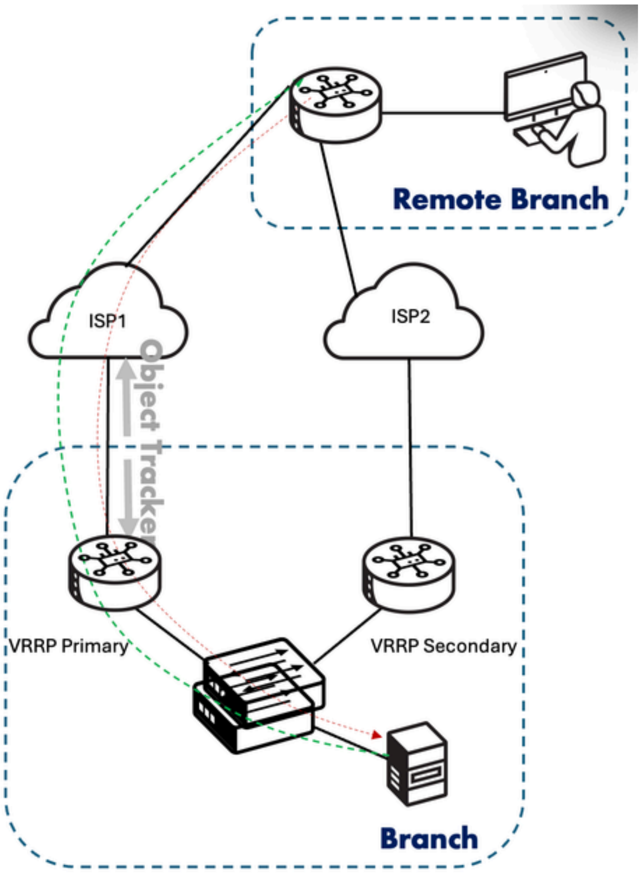
There are multiple use cases based on the criteria required for implementing VRRP-based interface tracking. Currently, the two modes supported are (i) interface (meaning any Tunnel interface which is bound with a local TLOC) or (ii) SIG interface (relating to SIG tunnel interfaces). In each case, the part being tracked is interface line-protocol.

Dual Router with Internet: The track object is bound to the VRRP group. In case the object of the tracker (which in this case is the SIG tunnel interface) goes down, this notifies the VRRP Primary router to trigger the state transition from Primary to backup and the backup router to become Primary. This state change can be influenced or triggered through two types of operations :

1. Decrement : Wherein the VRRP priority for the interface on which VRRP VIP is configured is reduced or decremented by a certain value, in the event the track object state transitions from UP to DOWN.
2. Shutdown : This is a method where the VRRP process is shutdown on the applied interface, in the event the track object state transitions from UP to DOWN. This method is not recommended in use cases where there are instances of asymmetric forwarding.

TLOC Change Preference: To avoid asymmetric traffic coming from other SDWAN sites into the site where VRRP runs on service VPN, the TLOC preference of the VRRP Primary router gets boosted by 1 if configured. You can even modify this value under configuration groups. This ensures that traffic from WAN to LAN is attracted by the VRRP Primary router itself. The traffic from LAN to WAN gets attracted by VRRP mechanism of VRRP Primary. This feature is independent to VRRP interface tracker. This is an optional command (tloc-change-pref) from the CLI standpoint.





Configuration

The configuration of object trackers are done via system templates in Legacy configuration, and then subsequently attaching the object tracker to the respective VRRP group under the service-VPN Ethernet Interface feature template. In Configuration group, this mechanism has been simplified by directly getting an option to add the object tracker to the respective service profile Ethernet Interface profile. Here are the ways to configure it, depending on the type of configuration method preferred by the user.

- Configuration group **Configuration** > **Configuration Groups** > **Service Profile** > **Ethernet Interface** > **Add Feature** > **Object Tracker**:

1. Provide a Name and Description for the new object tracker being defined.
2. Select the **Tracker Type** (among **Interface** and **SIG**).
3. Allocate an **Object Tracker ID**.
4. Provide the **Interface Name** (depending on the option chosen in step 2).

- **Configuration** > **Configuration Groups** > **Service Profile** > **Ethernet Interface** > **VRRP** section:

1. Under **IPv4 Settings**, click **Add VRRP IPv4**.
2. Define a **VRRP Group ID** and provide a local Priority for this service side ethernet interface.
3. Provide the **VRRP Virtual IP (VIP) Address**.
4. Enable the **TLOC Preference Change** knob and also provide the **TLOC Preference Change Value** (to handle asymmetric routing).
5. Click Add **VRRP Tracking Object**.
6. Under **Associate Object Tracker**, select from the dropdown on the **Object Tracker** (based on Name) you created before
7. Choose a **Tracker Action** (either **Shutdown** or **Decrement**).
8. Enter the **Decrement Value** (depending on the option chosen in step 7).

- Legacy configuration **Configuration** > **Templates** > **Feature Templates** > **System** > **Tracker** section:

1. Click the **New Object Tracker** button.
2. Select the **Tracker Type** (among **Interface** and **SIG**).
3. Allocate an **Object ID**.
4. Provide the **Interface name** (depending on the option chosen in step 2).

- **Configuration** > **Templates** > **Ethernet Interface (belonging to service side)** > **VRRP** section:

1. Click the **New VRRP** button.
2. Define a **VRRP Group ID** and provide a local Priority (optional, default value of 100 is chosen) for this service side ethernet interface.
3. Provide the **VRRP Virtual IP (VIP) Address**.
4. Enable the **TLOC Preference Change** knob and also provide the TLOC Preference Change Value (to handle asymmetric routing).
5. Under Object Tracker, click on **Add Tracking Object**.
6. Enter the **Object Tracker ID** (defined under system template).
7. Choose a **Tracker Action** (either **Shutdown** or **Decrement**).
8. Enter the **Decrement Value** (depending on the option chosen in step 7).

From the CLI standpoint, the configurations look like this:

```
(i) Using interface (Tunnel) Object Tracking :
!
track 10 interface Tunnel1 line-protocol
!
interface GigabitEthernet3
 description   SERVICE VPN 1
 no shutdown
 <snip>
 vrrp 10 address-family ipv4
  vrrpv2
  address 10.10.1.1
  priority 120
  timers advertise 1000
  track 10 decrement 40
  tloc-change increase-preference 120
 exit
exit
!
(ii) Using SIG interface Object Tracking :
!
track 20 service global
!
interface GigabitEthernet4
 description   SERVICE VPN 1
 no shutdown
 <snip>
 vrrp 10 address-family ipv4
  vrrpv2
  address 10.10.2.1
  priority 120
  timers advertise 1000
  track 20 decrement 40
  tloc-change increase-preference 120
 exit
exit
!
```

Verification

There are two options to verify explicitly configured object trackers for VRRP use cases.

- On **SD-WAN Manager Monitor** > **Devices** > **{select Device-Name}** > **Real Time**:

1. Under **Device Options**, type in "VRRP Information".

2. Check under **Individual VRRP Group (Group ID)** and view the statistics of the tracker (**Track Prefix Name, Track State, Discontinuity Time**, and **Last State Change Time**) based on your configured Object Tracker IDs.

- On **SD-WAN Manager Monitor** > **Devices** > **{select Device-Name}** > **Events**:

In the case of state change detected on the object tracker, the corresponding **VRRP Group** to which it is attached changes its state. The respective logs would populate in this section (with the Name as Vrrp Group State Change) with details such as **host name, if number, grp id, addr type, if name, vrrp group-state, state change-reason**, and **vpn id.**

On CLI of the Edge:

```
Router#show vrrp 10 GigabitEthernet 3
GigabitEthernet3 - Group 10 - Address-Family IPv4
  State is MASTER
  State duration 59 mins 56.703 secs
  Virtual IP address is 10.10.1.1
  Virtual MAC address is 0000.5E00.010A
  Advertisement interval is 1000 msec
  Preemption enabled
  Priority is 120
  State change reason is VRRP_TRACK_UP
  Tloc preference configured, value 120
    Track object 10 state UP decrement 40
  Master Router is 10.10.1.3 (local), priority is 120
  Master Advertisement interval is 1000 msec (expires in 393 msec)
  Master Down interval is unknown
  FLAGS: 1/1

Router#show track 10
Track 10
  Interface Tunnel1 line-protocol
  Line protocol is Up
    7 changes, last change 01:00:47
  Tracked by:
    VRRPv3 GigabitEthernet3 IPv4 group 10

Router#show track 10 brief
Track Type        Instance                   Parameter        State Last Change
10    interface   Tunnel1                    line-protocol    Up    01:01:02

Router#show interface Tunnel1
Tunnel1 is up, line protocol is up
  Hardware is Tunnel
  Interface is unnumbered. Using address of GigabitEthernet1 (172.25.12.1)
  MTU 9980 bytes, BW 100 Kbit/sec, DLY 50000 usec,
     reliability 255/255, txload 1/255, rxload 2/255
  Encapsulation TUNNEL, loopback not set
  Keepalive not set
  Tunnel linestate evaluation up
  Tunnel source 172.25.12.1 (GigabitEthernet1)
  <snip>
```

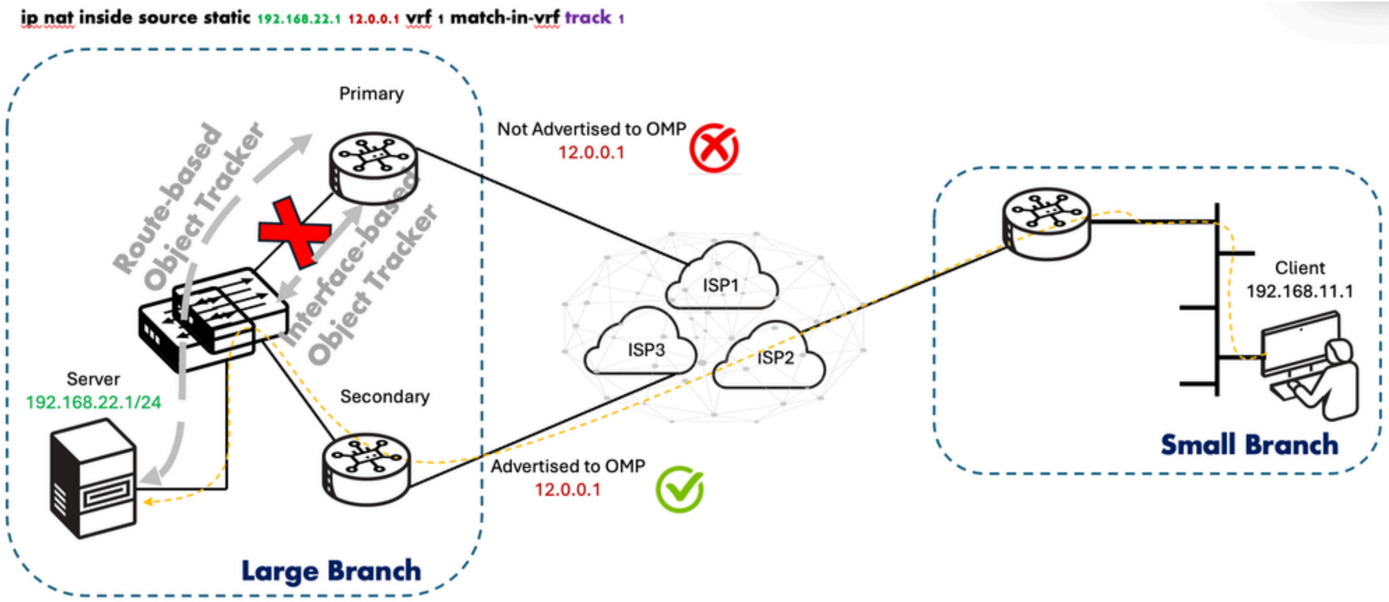
Interface/Route Object Trackers Used for Service-VPN NAT Tracking

The service-side NAT Object Tracker was a feature introduced in 20.8/17.8 release, wherein the inside global address used in service-VPN NAT (inside static NAT and inside dynamic NAT) is only advertised into OMP if (i) the inside local address is found to be reachable OR (ii) the line protocol of the LAN/service-side interface is UP as per the attached object tracker. Hence, the object tracker types that can be used are (i) route or (ii) interface. Depending on the state of the LAN prefix or LAN interface, NAT route advertisements through OMP are either added or removed. You can view event logs in Cisco SD-WAN Manager for monitoring which NAT route advertisements are added or removed.

There are no probes used here by the tracker. Instead, it uses (i) the presence of a routing entry in the routing table OR (ii) the line-protocol state to decide on the tracker state (up/down). There are no reaction intervals in the presence of a routing entry or interface line-protocol based trackers - the moment the routing entry or interface line-protocol goes DOWN, the track state is also brought to the DOWN state. Immediately the inside global address used in the NAT statement associated with the object tracker is stopped from being advertised into OMP. To learn more about Service VPN NAT trackers, please visit the configuration guide.

Use Cases

If a LAN interface or a LAN prefix is down, the service-side NAT object tracker goes down automatically. You can view event logs in Cisco SD-WAN Manager for monitoring which NAT route advertisements are added or removed. In the next use case, the client is required to access the server in the large branch. However, the problem arises in situations when either the route pointing to the server on the large branch edges (in HA) get removed OR when the LAN-side (service-side) interface goes down on any one edge in the large branch. In such situations, when you apply service-side NAT with object tracker, ensure - ensure that the traffic coming inbound from the client is always directed to the correct edge located in the large branch by controlling the inside global address advertisement into OMP. In case such control is not enforced on the route advertisement into OMP, the traffic ends up getting blackholed due to non-reachability from that respective edge to the server in the large branch.



```
ip nat inside source static 192.168.22.1 12.0.0.1 vrf 1 match-in-vrf track 1
```

Configuration

The configuration of object trackers are done via system templates in Legacy configuration, and then subsequently attaching the object tracker to the respective NAT statement (inside static or inside dynamic) in the service-VPN feature template. In Configuration group, this mechanism has been simplified by directly getting an option to add the object tracker to the respective service profile Ethernet Interface profile. Here are the ways to configure it, depending on the type of configuration method preferred by the user.

- Configuration group **Configuration** > **Configuration Groups** > **Service Profile** > **Add Feature** > **Object Tracker**:

1. Provide a Name and Description for the new object tracker being defined.
2. Select the **Tracker Type** (among **Interface** and **route**).
3. Allocate an **Object Tracker ID**.
4. Provide the **Interface Name OR provide the Route IP, Route IP Mask**, and **VPN** (depending on the option chosen in step 2).

- **Configuration** > **Configuration Groups** > **Service Profile** > **NAT** section:

1. Create a NAT Pool (mandatory for triggering SSNAT) by clicking the **Add NAT Pool** button.

2. Provide the details of the NAT Pool, such as **NatPool Name, Prefix Length, Range Start, Range End**, and **Direction.**

3. Move to **Static NAT** in the same section and click the **Add New Static NAT** button. (You can also choose to attach the object tracker to inside dynamic pool NAT).

4. Provide the details such as **Source IP, Translated Source IP**, and **Static NAT Direction**.

5. Under the **Associate Object Tracker** field, choose from the dropdown list the previously created object tracker.

- Legacy configuration **Configuration > Templates > Feature Templates > System > Tracker** section:

1. Click the **New Object Tracker** button.
2. Select the **Tracker Type** (among **Interface** and **route**).
3. Allocate an **Object ID**.
4. Provide the **Interface name OR Route IP, Route IP Mask**, and **VPN (**depending on the option chosen in step 2).

- **Configuration** > **Templates** > **Cisco VPN (belonging to service side)** > **N**AT section:

1. Create a NAT Pool (mandatory for triggering SSNAT) by clicking the **New NAT Pool** button.

2. Provide the details of the NAT Pool, such as **Nat Pool Name, NAT Pool Prefix Length, NAT Pool Range Start, NAT Pool Range End**, and **NAT Direction**.

3. Move to Static NAT in the same section and click the **New Static NAT** button. (You can also choose to attach the object tracker to inside dynamic pool NAT).

4. Provide the details such as **Source IP Address, Translated Source IP Address, Static NAT Direction.**

5. Under the **Add Object Tracker** field, type in the name of the previously created object tracker.

From the CLI standpoint, the configurations look like this:

```
(i) Using route-based object tracking on SSNAT (inside static or inside dynamic) :
!
track 20 ip route 192.168.10.4 255.255.255.255 reachability
 ip vrf 1
!
ip nat pool natpool10 14.14.14.1 14.14.14.5 prefix-length 24
ip nat inside source list global-list pool natpool10 vrf 1 match-in-vrf overload
ip nat inside source static 10.10.1.4 15.15.15.1 vrf 1 match-in-vrf track 20
!
(ii) Using interface-based object tracking on SSNAT (inside static or inside dynamic) :
!
track 20 interface GigabitEthernet3 line-protocol
!
ip nat pool natpool10 14.14.14.1 14.14.14.5 prefix-length 24
ip nat inside source list global-list pool natpool10 vrf 1 match-in-vrf overload
ip nat inside source static 10.10.1.4 15.15.15.1 vrf 1 match-in-vrf track 20
!
```

Assumption with SSNAT use case is that users apply data-policy to match traffic for both in -> out and out -> in NAT flows.

Verification

There are two areas of verification of explicitly configured object trackers for NAT use cases.

- On **SD-WAN Manager : Monitor > Devices > {select Device-Name} > Real Time**:

1. Under Device Options, type in "IP NAT Translation".

2. Check under Individual NAT Translation and view the statistics of the entry (Inside Local address/port,

Inside Global address/port, Outside Local address/port, Outside Global address/port, VRF ID, VRF Name and Protocol) based on your configured Object Tracker IDs.

- On **SD-WAN Manager : Monitor > Devices > {select Device-Name} > Events:**

In the case of state change detected on the object tracker corresponding to the NAT route being pruned into OMP, events named "NAT Route Change" appear, which contain details such as host name, object tracker, address, mask, route type and update. Here, the address and mask maps to the inside global address as configured under the static NAT statement.

On CLI of the Edge:

```
Router#show ip nat translations vrf 1
Pro  Inside global         Inside local          Outside local         Outside global
---  15.15.15.1            10.10.1.4             ---                   ---
icmp 15.15.15.1:4          10.10.1.4:4           20.20.1.1:4           20.20.1.1:4
Total number of translations: 2

Router#show track 20
Track 20
  IP route 192.168.10.4 255.255.255.255 reachability
  Reachability is Up (OSPF)
    4 changes, last change 00:02:56
  VPN Routing/Forwarding table "1"
  First-hop interface is GigabitEthernet3
  Tracked by:
    NAT 0

Router#show track 20 brief
Track Type        Instance                   Parameter        State Last Change
20    ip route    192.168.10.4/32            reachability     Up    00:03:04

Remote-Router#show ip route vrf 1 15.15.15.1

Routing Table: 1
Routing entry for 15.15.15.1/32
  Known via "omp", distance 251, metric 0, type omp
  Redistributing via ospf 1
  Advertised by ospf 1 subnets
  Last update from 10.10.10.12 on Sdwan-system-intf, 00:03:52 ago
  Routing Descriptor Blocks:
  * 10.10.10.12 (default), from 10.10.10.12, 00:03:52 ago, via Sdwan-system-intf
      Route metric is 0, traffic share count is 1

Remote-Router#show sdwan omp routes 15.15.15.1/32
<snip>
                                                      PATH                      ATTRIBUTE
TENANT    VPN    PREFIX              FROM PEER        ID     LABEL    STATUS    TYPE       TLOC IP
-------------------------------------------------------------------------------------------------------
0         1      15.15.15.1/32       1.1.1.3          1      1003     C,I,R     installed  10.10.10.12
                                     1.1.1.3          2      1003     Inv,U     installed  10.10.10.12
                                     1.1.1.3          3      1003     C,I,R     installed  10.10.10.12
```