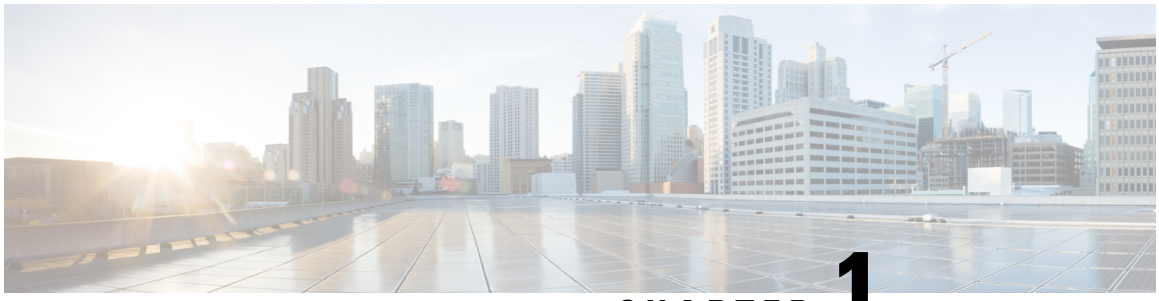
Policies Configuration Guide for vEdge Routers, Cisco SD-WAN Release 20

First Published: 2020-03-17

Last Modified: 2022-08-26

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

CONTENTS



CHAPTER 1

Read Me First

Note To achieve simplification and consistency, the Cisco SD-WAN solution has been rebranded as Cisco Catalyst SD-WAN. In addition, from Cisco IOS XE SD-WAN Release 17.12.1a and Cisco Catalyst SD-WAN Release 20.12.1, the following component changes are applicable: **Cisco vManage** to **Cisco Catalyst SD-WAN Manager**, **Cisco vAnalytics** to **Cisco Catalyst SD-WAN Analytics**, **Cisco vBond** to **Cisco Catalyst SD-WAN Validator**, **Cisco vSmart** to **Cisco Catalyst SD-WAN Controller**, and **Cisco Controllers** to **Cisco Catalyst SD-WAN Control Components**. See the latest Release Notes for a comprehensive list of all the component brand name changes. While we transition to the new names, some inconsistencies might be present in the documentation set because of a phased approach to the user interface updates of the software product.

Related References

- Cisco Catalyst SD-WAN Control Components Compatibility Matrix and Server Recommendations
- Cisco Catalyst SD-WAN Device Compatibility

User Documentation

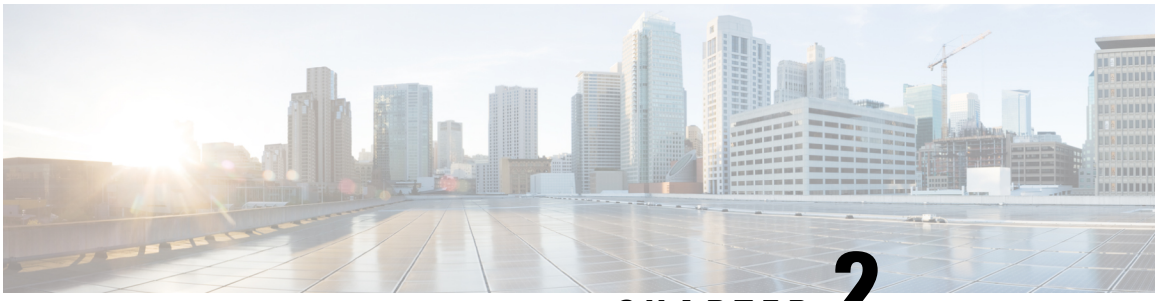
- User Documentation for Cisco SD-WAN Release 20

Communications, Services, and Additional Information

- Sign up for Cisco email newsletters and other communications at: Cisco Profile Manager.
- For information on the latest technical, advanced, and remote services to increase the operational reliability of your network visit Cisco Services.
- To browse and discover secure, validated enterprise-class apps, products, solutions, and services, visit Cisco Devnet.
- To obtain general networking, training, and certification titles from Cisco Press Publishers, visit Cisco Press.
- To find warranty information for a specific product or product family, visit Cisco Warranty Finder.
- To view open and resolved bugs for a release, access the Cisco Bug Search Tool.
- To submit a service request, visit Cisco Support.

Documentation Feedback

To provide feedback about Cisco technical documentation use the feedback form available in the right pane of every online document.



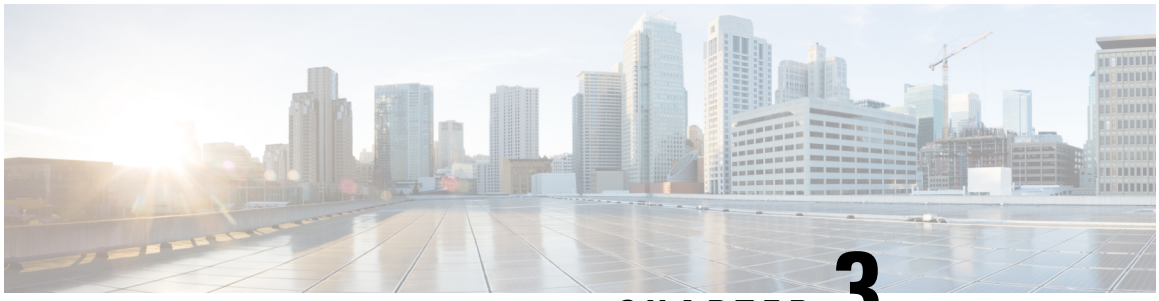
CHAPTER 2

What's New in Cisco Catalyst SD-WAN

Note To achieve simplification and consistency, the Cisco SD-WAN solution has been rebranded as Cisco Catalyst SD-WAN. In addition, from Cisco IOS XE SD-WAN Release 17.12.1a and Cisco Catalyst SD-WAN Release 20.12.1, the following component changes are applicable: **Cisco vManage** to **Cisco Catalyst SD-WAN Manager**, **Cisco vAnalytics** to **Cisco Catalyst SD-WAN Analytics**, **Cisco vBond** to **Cisco Catalyst SD-WAN Validator**, **Cisco vSmart** to **Cisco Catalyst SD-WAN Controller**, and **Cisco Controllers** to **Cisco Catalyst SD-WAN Control Components**. See the latest Release Notes for a comprehensive list of all the component brand name changes. While we transition to the new names, some inconsistencies might be present in the documentation set because of a phased approach to the user interface updates of the software product.

Note Cisco is constantly enhancing the Cisco Catalyst SD-WAN solution with every release and we try and keep the content in line with the latest enhancements. The following table lists new and modified features we documented in the Configuration, Command Reference, and Hardware Installation guides. For information on additional features and fixes that were committed to the Cisco Catalyst SD-WAN solution, see the *Resolved and Open Bugs* section in the Release Notes.

What's New in Cisco SD-WAN (vEdge) Release 20.x



CHAPTER 3

Policy Overview

Note To achieve simplification and consistency, the Cisco SD-WAN solution has been rebranded as Cisco Catalyst SD-WAN. In addition, from Cisco IOS XE SD-WAN Release 17.12.1a and Cisco Catalyst SD-WAN Release 20.12.1, the following component changes are applicable: **Cisco vManage** to **Cisco Catalyst SD-WAN Manager**, **Cisco vAnalytics** to **Cisco Catalyst SD-WAN Analytics**, **Cisco vBond** to **Cisco Catalyst SD-WAN Validator**, **Cisco vSmart** to **Cisco Catalyst SD-WAN Controller**, and **Cisco Controllers** to **Cisco Catalyst SD-WAN Control Components**. See the latest Release Notes for a comprehensive list of all the component brand name changes. While we transition to the new names, some inconsistencies might be present in the documentation set because of a phased approach to the user interface updates of the software product.

Policy influences the flow of data traffic and routing information among Cisco vEdge deviceCisco IOS XE Catalyst SD-WAN devices in the overlay network.

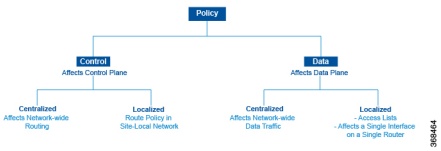
Policy comprises:

- Routing policy—which affects the flow of routing information in the network's control plane.
- Data policy—which affects the flow of data traffic in the network's data plane.

To implement enterprise-specific traffic control requirements, you create basic policies, and deploy advanced features that are activated by means of the policy configuration infrastructure.

Just as the Cisco Catalyst SD-WAN overlay network architecture clearly separates the control plane from the data plane and control between centralized and localized functions, the Cisco Catalyst SD-WAN policy is cleanly separated. Policies apply either to control plane or data plane traffic, and they are configured either centrally on Cisco SD-WAN Controllers or locally on Cisco vEdge deviceCisco IOS XE Catalyst SD-WAN devices. The following figure illustrates the division between control and data policy, and between centralized and local policy.

Figure 1: Policy Architecture



Control and Data Policy

Control policy is the equivalent of routing protocol policy, and data policy is equivalent to what are commonly called access control lists (ACLs) and firewall filters.

Centralized and Localized Policy

The Cisco Catalyst SD-WAN policy design provides a clear separation between centralized and localized policy. In short, centralized policy is provisioned on the centralized Cisco SD-WAN Controllers in the overlay network, and the localized policy is provisioned on Cisco vEdge devices, which sit at the network edge between a branch or enterprise site and a transport network, such as the Internet, MPLS, or metro Ethernet.

Centralized Policy

Centralized policy refers to policy provisioned on Cisco SD-WAN Controllers, which are the centralized controllers in the Cisco Catalyst SD-WAN overlay network. Centralized policy comprises two components:

- Control policy, which affects the overlay network–wide routing of traffic
- Data policy, which affects the data traffic flow throughout the VPN segments in the network

Centralized control policy applies to the network-wide routing of traffic by affecting the information that is stored in the Cisco SD-WAN Controller's route table and that is advertised to the Cisco vEdge devices. The effects of centralized control policy are seen in how Cisco vEdge devices direct the overlay network's data traffic to its destination.

Note The centralized control policy configuration itself remains on the Cisco SD-WAN Controller and is never pushed to local devices.

Centralized data policy applies to the flow of data traffic throughout the VPNs in the overlay network. These policies can permit and restrict access based either on a 6-tuple match (source and destination IP addresses and ports, DSCP fields, and protocol) or on VPN membership. These policies are pushed to the selected Cisco vEdge devices.

Localized Policy

Localized policy refers to a policy that is provisioned locally through the CLI on the Cisco vEdge devices, or through a Cisco SD-WAN Manager device template.

Localized control policy is also called as route policy, which affects (BGP and OSPF) routing behavior on the site-local network.

Localized data policy allows you to provision access lists and apply them to a specific interface or interfaces on the device. Simple access lists permit and restrict access based on a 6-tuple match (source and destination IP addresses and ports, DSCP fields, and protocol), in the same way as with centralized data policy. Access lists also allow provisioning of class of service (CoS), policing, and mirroring, which control how data traffic flows out of and in to the device's interfaces and interface queues.

The design of the Cisco Catalyst SD-WAN policy distinguishes basic and advanced policies. Basic policy allows you to influence or determine basic traffic flow through the overlay network. Here, you perform standard policy tasks, such as managing the paths along which traffic is routed through the network, and permitting or blocking traffic based on the address, port, and DSCP fields in the packet's IP header. You can

also control the flow of data traffic into and out of a Cisco vEdge device 's interfaces, enabling features such as class of service and queuing, mirroring, and policing.

Advanced features of Cisco Catalyst SD-WAN policy offer specialized policy-based network applications. Examples of these applications include the following:

- Service chaining, which redirects data traffic to shared devices in the network, such as firewall, intrusion detection and prevention (IDS), load balancer, and other devices, before the traffic is delivered to its destination. Service chaining obviates the need to have a separate device at each branch site.
- Application-aware routing, which selects the best path for traffic based on real-time network and path performance characteristics.
- Cflowd, for monitoring traffic flow.
- Converting a Cisco vEdge device into a NAT device, to allow traffic destined for the Internet or other public network can exit directly from the Cisco vEdge device.

By default, no policy of any kind is configured on Cisco vEdge devices, either on the centralized Cisco SD-WAN Controllers or the local Cisco vEdge devices. When control plane traffic, which distributes route information, is unpolicied:

- All route information that OMP propagates among the Cisco vEdge devices is shared, unmodified, among all Cisco SD-WAN Controllers and all Cisco vEdge devices in the overlay network domain.
- No BGP or OSPF route policies are in place to affect the route information that Cisco vEdge device s propagate within their local site network.

When data plane traffic is unpolicied, all data traffic is directed towards its destination based solely on the entries in the local Cisco vEdge device's route table, and all VPNs in the overlay network can exchange data traffic.

- Policy Architecture, on page 7
- Cisco Catalyst SD-WAN Controller Policy Components, on page 15
- Design Cisco Catalyst SD-WAN Controller Policy Processing and Application, on page 20
- Cisco Cisco Catalyst SD-WAN Controller Policy Operation, on page 21
- Configure and Execute Cisco SD-WAN Controller Policies, on page 27

Policy Architecture

This topic offers an orientation about the architecture of the Cisco Catalyst SD-WAN policy used to implement overlay network-wide policies. These policies are called Cisco SD-WAN Validator **policy** or **centralized policy**, because you configure them centrally on a Cisco SD-WAN Controller. Cisco SD-WAN Controller policy affects the flow of both control plane traffic (routing updates carried by Overlay Management Protocol (OMP) and used by the Cisco SD-WAN Controllers to determine the topology and status of the overlay network) and data plane traffic (data traffic that travels between the Cisco vEdge devices across the overlay network).

With Cisco Catalyst SD-WAN, you can also create routing policies on the Cisco vEdge devices. These policies are simply traditional routing policies that are associated with routing protocol (BGP or OSPF) locally on the devices. You use them in the traditional sense for controlling BGP and OSPF, for example, to affect the exchange of route information, to set route attributes, and to influence path selection.

Centralized Control Policy Architecture

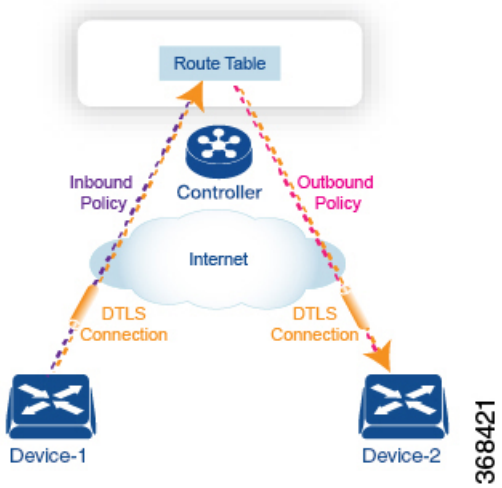
In the Cisco Catalyst SD-WAN network architecture, centralized control policy is handled by the Cisco SD-WAN Controller, which effectively is the routing engine of the network. The Cisco SD-WAN Controller is the centralized manager of network-wide routes, maintaining a primary route table for these routes. The Cisco SD-WAN Controller builds its route table based on the route information advertised by the Cisco vEdge devices in its domain, using these routes to discover the network topology and to determine the best paths to network destinations. The Cisco SD-WAN Controller distributes route information from its route table to the devices in its domain which in turn use these routes to forward data traffic through the network. The result of this architecture is that networking-wide routing decisions and routing policy are orchestrated by a central authority instead of being implemented hop by hop, by the devices in the network.

Centralized control policy allows you to influence the network routes advertised by the Cisco SD-WAN Controllers. This type of policy, which is provisioned centrally on the Cisco SD-WAN Controller, affects both the route information that the Cisco SD-WAN Controller stores in its primary route table and the route information that it distributes to the devices.

Centralized control policy is provisioned and applied only on the Cisco SD-WAN Controller. The control policy configuration itself is never pushed to devices in the overlay network. What is pushed to the devices, using the Overlay Management Protocol (OMP), are the results of the control policy, which the devices then install in their local route tables and use for forwarding data traffic. This design means that the distribution of network-wide routes is always administered centrally, using policies designed by network administrators. These policies are always implemented by centralized Cisco SD-WAN Controllers, which are responsible for orchestrating the routing decisions in the Cisco Catalyst SD-WAN overlay network.

Within a network domain, the network topology map on all Cisco SD-WAN Controllers must be synchronized. To support this, you must configure identical policies on all the Cisco SD-WAN Controllers in the domain.

Figure 2: Centralized Control Policy



All centralized control plane traffic, including route information, is carried by OMP peering sessions that run within the secure, permanent DTLS connections between devices and the Cisco SD-WAN Controllers in their domain. The end points of an OMP peering session are identified by the system IDs of the devices, and the peering sessions carry the site ID, which identifies the site in which the device is located. A DTLS connection and the OMP session running over it remain active as long as the two peers are operational.

Control policy can be applied both inbound, to the route advertisements that the Cisco SD-WAN Controller receives from the devices, and outbound, to advertisements that it sends to them. Inbound policy controls which routes and route information are installed in the local routing database on the Cisco SD-WAN Controller, and whether this information is installed as-is or is modified. Outbound control policy is applied after a route is retrieved from the routing database, but before a Cisco SD-WAN Controller advertises it, and affects whether the route information is advertised as-is or is modified.

Route Types

The Cisco SD-WAN Controller learns the network topology from OMP routes, which are Cisco Catalyst SD-WAN-specific routes carried by OMP. There are three types of OMP routes:

- Cisco Catalyst SD-WAN OMP routes—These routes carry prefix information that the devices learn from the routing protocols running on its local network, including routes learned from BGP and OSPF, as well as direct, connected, and static routes. OMP advertises OMP routes to the Cisco SD-WAN Controller by means of an OMP route SAFI (Subsequent Address Family Identifier). These routes are commonly simply called OMP routes.
- TLOC routes—These routes carry properties associated with transport locations, which are the physical points at which the devices connect to the WAN or the transport network. Properties that identify a TLOC include the IP address of the WAN interface and a color that identifies a particular traffic flow. OMP advertises TLOC routes using a TLOC SAFI.
- Service routes—These routes identify network services, such as firewalls and IDPs, that are available on the local-site network to which the devices are connected. OMP advertises these routes using a service SAFI.

Default Behavior Without Centralized Control Policy

By default, no centralized control policy is provisioned on the Cisco SD-WAN Controller. This results in the following route advertisement and redistribution behavior within a domain:

- All Cisco vEdge devices redistribute all the route-related prefixes that they learn from their site-local network to the Cisco SD-WAN Controller. This route information is carried by OMP route advertisements that are sent over the DTLS connection between the devices and the Cisco SD-WAN Controller. If a domain contains multiple Cisco SD-WAN Controllers, the devices send all OMP route advertisements to all the controllers.
- All the devices send all TLOC routes to the Cisco SD-WAN Controller or controllers in their domain, using OMP.
- All the devices send all service routes to advertise any network services, such as firewalls and IDPs, that are available at the local site where the device is located. Again, these are carried by OMP.
- The Cisco SD-WAN Controller accepts all the OMP, TLOC, and service routes that it receives from all the devices in its domain, storing the information in its route table. The Cisco SD-WAN Controller tracks which OMP routes, TLOCs, and services belong to which VPNs. The Cisco SD-WAN Controller uses all the routes to develop a topology map of the network and to determine routing paths for data traffic through the overlay network.
- The Cisco SD-WAN Controller redistributes all information learned from the OMP, TLOC, and service routes in a particular VPN to all the devices in the same VPN.
- The devices regularly send route updates to the Cisco SD-WAN Controller.

- The Cisco SD-WAN Controller recalculates routing paths, updates its route table, and advertises new and changed routing information to all the devices.

Behavior Changes with Centralized Control Policy

When you do not want to redistribute all route information to all Cisco vEdge devices in a domain, or when you want to modify the route information that is stored in the Cisco Catalyst SD-WAN Controller's route table or that is advertised by the Cisco Catalyst SD-WAN Controller, you design and provision a centralized control policy. To activate the control policy, you apply it to specific sites in the overlay network in either the inbound or the outbound direction. The direction is with respect to the Cisco Catalyst SD-WAN Controller. All provisioning of centralized control policy is done on the Cisco Catalyst SD-WAN Controller.

Applying a centralized control policy in the inbound direction filters or modifies the routes being advertised by the Cisco vEdge device before they are placed in the route table on the Cisco Catalyst SD-WAN Controller. As the first step in the process, routes are either accepted or rejected. Accepted routes are installed in the route table on the Cisco Catalyst SD-WAN Controller either as received or as modified by the control policy. Routes that are rejected by a control policy are silently discarded.

Applying a control policy in outbound direction filters or modifies the routes that the Cisco Catalyst SD-WAN Controller redistributes to the Cisco vEdge devices. As the first step of an outbound policy, routes are either accepted or rejected. For accepted routes, centralized control policy can modify the routes before they are distributed by the Cisco Catalyst SD-WAN Controller. Routes that are rejected by an outbound policy are not advertised.

VPN Membership Policy

A second type of centralized data policy is VPN membership policy. It controls whether a Cisco vEdge device can participate in a particular VPN. VPN membership policy defines which VPNs of a device is allowed and which is not allowed to receive routes from.

VPN membership policy can be centralized, because it affects only the packet headers and has no impact on the choice of interface that a Cisco vEdge device uses to transmit traffic. What happens instead is that if, because of a VPN membership policy, a device is not allowed to receive routes for a particular VPN, the Cisco Catalyst SD-WAN Controller never forwards those routes to that driver.

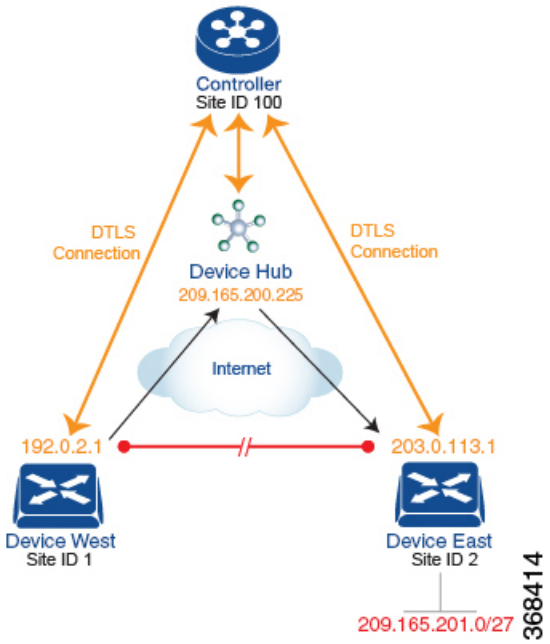
Examples of Modifying Traffic Flow with Centralized Control Policy

This section provides some basic examples of how you can use centralized control policies to modify the flow of data traffic through the overlay network.

Create an Arbitrary Topology

When data traffic is exchanged between two Cisco vEdge devices, if you have provisioned no control policy, the two devices establish an IPsec tunnel between them and the data traffic flows directly from one device to the next. For a network with only two devices or with just a small number of devices, establishing connections between each pair of devices is generally not been an issue. However, such a solution does not scale. In a network with hundreds or even thousands of branches, establishing a full mesh of IPsec tunnels tax the CPU resources of each device.

Figure 3: Arbitrary Topology



One way to minimize this overhead is to create a hub-and-spoke type of topology in which one of the devices acts as a hub site that receives the data traffic from all the spoke, or branch, devices and then redirects the traffic to the proper destination. This example shows one of the ways to create such a hub-and-spoke topology, which is to create a control policy that changes the address of the TLOC associated with the destination.

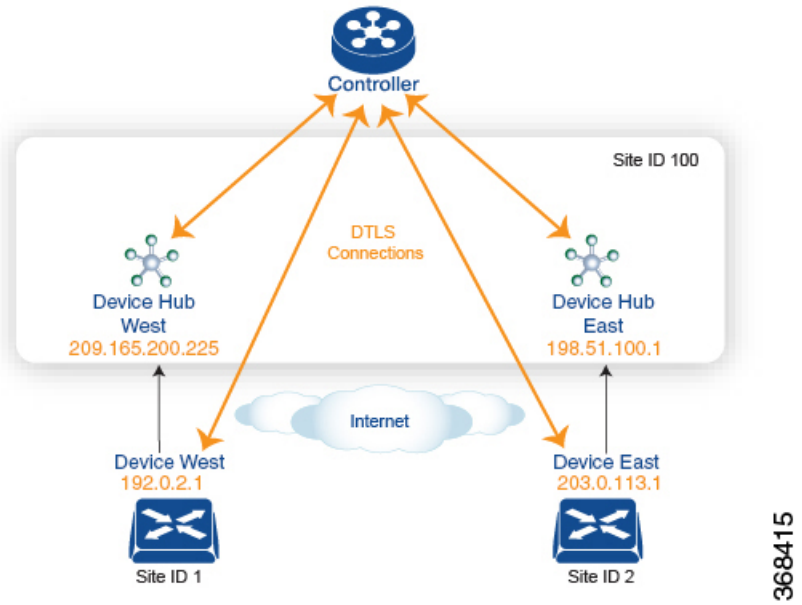
The figure illustrates how such a policy might work. The topology has two branch locations, West and East. When no control policy is provisioned, these two devices exchange data traffic with each other directly by creating an IPsec tunnel between them (shown by the red line). Here, the route table on the Device West contains a route to Device East with a destination TLOC of 203.0.113.1, color gold (which we write as the tuple {192.0.2.1, gold}), and Device East route table has a route to the West branch with a destination TLOC of {203.0.113.1, gold}.

To set up a hub-and-spoke–type topology here, we provision a control policy that causes the West and East devices to send all data packets destined for the other device to the hub device. (Remember that because control policy is always centralized, you provision it on the Cisco Catalyst SD-WAN Controller.) On the Device West, the policy simply changes the destination TLOC from {203.0.113.1, gold} to {209.165.200.225, gold}, which is the TLOC of the hub device, and on the Device East, the policy changes the destination TLOC from {192.0.2.1, gold} to the hub's TLOC, {209.165.200.225, gold}. If there were other branch sites on the west and east sides of the network that exchange data traffic, you could apply these same two control policies to have them redirect all their data traffic through the hub.

Set Up Traffic Engineering

Control policy allows you to design and provision traffic engineering. In a simple case, suppose that you have two devices acting as hub devices. If you want data traffic destined to a branch Cisco vEdge device to always transit through one of the hub devices, set the TLOC preference value to favor the desired hub device.

Figure 4: Traffic Engineering Topology

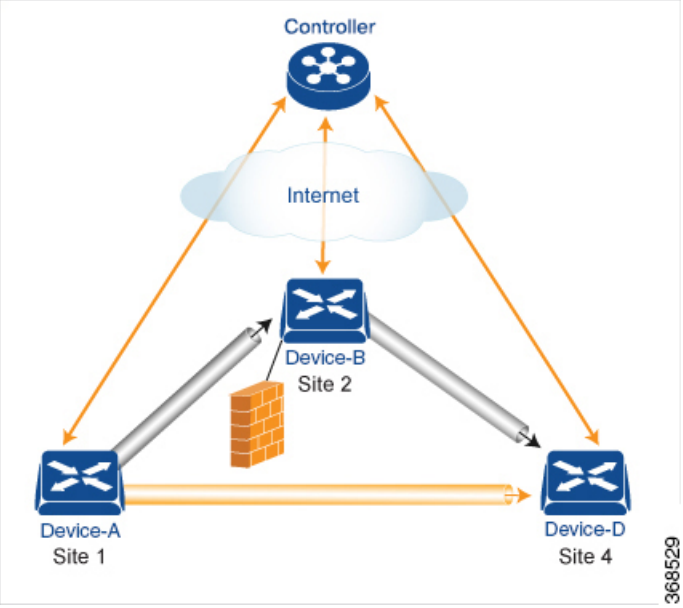


The figure shows that Site ID 100 has two hub devices, one that serves the West side of the network and a second that serves the East side. Data traffic from the Device West must be handled by the Device West hub, and similarly, data traffic from the Device East branch must go through the Device East hub.

To engineer this traffic flow, you provision two control policies, one for Site ID 1, where the Device West device is located, and a second one for Site ID 2. The control policy for Site ID 1 changes the TLOC for traffic destined to the Device East to {209.165.200.225, gold}, and the control policy for Site ID 2 changes the TLOC for traffic destined for Site ID 1 to {198.51.100.1, gold}. One additional effect of this traffic engineering policy is that it load-balances the traffic traveling through the two hub devices.

With such a traffic engineering policy, a route from the source device to the destination device is installed in the local route table, and traffic is sent to the destination regardless of whether the path between the source and destination devices is available. Enabling end-to-end tracking of the path to the ultimate destination allows the Cisco Catalyst SD-WAN Controller to monitor the path from the source to the destination, and to inform the source device when that path is not available. The source device can then modify or remove the path from its route table.

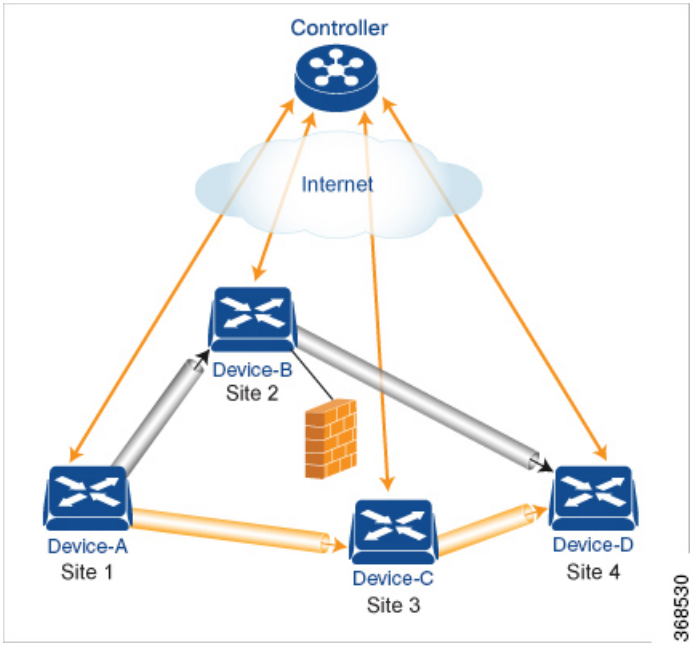
Figure 5: Traffic Engineering 2



The figure Traffic Engineering 2 illustrates end-to-end path tracking. It shows that traffic from Device-A that is destined for Device-D first goes to an intermediate device, Device-B, perhaps because this intermediate device provides a service, such as a firewall. (You configure this traffic engineering with a centralized control policy that is applied to Device-A, at Site 1.) Then Device-B, which has a direct path to the ultimate destination, forwards the traffic to Device-D. So, in this example, the end-to-end path between Device-A and Device-D comprises two tunnels, one between Device-A and Device-B, and the second between Device-B and Device-D. The Cisco Catalyst SD-WAN Controller tracks this end-to-end path, and it notifies Device-A if the portion of the path between Device-B and Device-D becomes unavailable.

As part of end-to-end path tracking, you can specify how to forwarded traffic from the source to the ultimate destination using an intermediate device. The default method is strict forwarding, where traffic is always sent from Device-A to Device-B, regardless of whether Device-B has a direct path to Device-D or whether the tunnel between Device-B and Device-D is up. More flexible methods forward some or all traffic directly from Device-A to Device-D. You can also set up a second intermediate device to provide a redundant path with the first intermediate device is unreachable and use an ECMP method to forward traffic between the two. The figure Traffic Engineering3 adds Device-C as a redundant intermediate device.

Figure 6: Traffic Engineering 3



Centralized control policy, which you configure on Cisco Catalyst SD-WAN Controllers, affects routing policy based on information in OMP routes and OMP TLOCs.

This type of policy allows you to set actions for matching routes and TLOCs, including redirecting packets through network services, such as firewalls, a feature that is called service chaining.

In domains with multiple Cisco Catalyst SD-WAN Controllers, all the controllers must have the same centralized control policy configuration to ensure that routing within the overlay network remains stable and predictable.

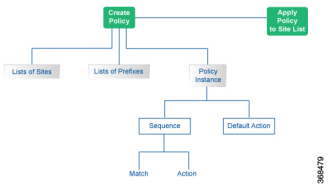
Configure Centralized Policy Based on Prefixes and IP Headers

A centralized data policy based on source and destination prefixes and on headers in IP packets consists of a series of numbered (ordered) sequences of match-action pair that are evaluated in order, from lowest sequence number to highest sequence number. When a packet matches one of the match conditions, the associated action is taken and policy evaluation on that packets stops. Keep this in mind as you design your policies to ensure that the desired actions are taken on the items subject to policy.

If a packet matches no parameters in any of the sequences in the policy configuration, it is dropped and discarded by default.

Configuration Components

The following figure illustrates the configuration components for a centralized data policy:



Cisco Catalyst SD-WAN Controller Policy Components

The Cisco SD-WAN Controller policies that implement overlay network-wide policies are implemented on a Cisco Catalyst SD-WAN Control Components. Because Cisco SD-WAN Controllers are centralized devices, you can manage and maintain Cisco SD-WAN Controller policies centrally, and you can ensure consistency in the enforcement of policies across the overlay network.

The implementation of Cisco SD-WAN Controller policy is done by configuring the entire policy on the Cisco Catalyst SD-WAN Control Components. Cisco SD-WAN Controller policy configuration is accomplished with three building blocks:

- Lists define the targets of policy application or matching.
- Policy definition, or policies, controls aspects of control and forwarding. There are different types of policy, including:
  - app-route-policy (for application-aware routing)
  - cflowd-template (for cflowd flow monitoring)
  - control-policy (for routing and control plane information)
  - data-policy (for data traffic)
  - vpn-membership-policy (for limiting the scope of traffic to specific VPNs)
- Policy application controls what a policy is applied towards. Policy application is site-oriented, and is defined by a specific list called a site-list.

You assemble these three building blocks to Cisco SD-WAN Controller policy. More specifically, policy is the sum of one or more lists, one policy definition, and at least one policy applications, as shown in the table below.

Table 1: The Three Building Blocks of Cisco SD-WAN Controller Policies

Lists		Policy Definition		Policy Application
`data-prefix-list`: List of prefixes for use with a data-policy `prefix-list`: List of prefixes for use with any other policy `site-list`: List of site-id:s for use in policy and apply-policy `tloc-list` : List of tloc:s for use in policy `vpn-list` : List of vpn:s for use in policy	+	`app-route-policy`: Used with `sla-classes` for application-aware routing `cflowd-template`: Configures the `cflowd` agents on the Cisco vEdge devices `control-policy`: Controls OMP routing control `data-policy`: Provides vpn-wide policy-based routing `vpn-membership-policy`: Controls vpn membership across nodes	+	`apply-policy`: Used with a `site-list` to determine where policies are applied
=				

Lists		Policy Definition		Policy Application
Complete policy definition configured on Cisco SD-WAN Controller and enforced either on Cisco SD-WAN Controller or on Cisco vEdge devices.				

Lists

Lists are how you group related items so that you can reference them all together. Examples of items you put in lists are prefixes, TLOCs, VPNs, and overlay network sites. In the Cisco SD-WAN Controller policy, you invoke lists in two places: when you create a policy definition and when you apply a policy. Separating the definition of the related items from the definition of policy means that when you can add or remove items from a lists, you make the changes only in a single place: You do not have to make the changes through the policy definition. So if you add ten sites to your network and you want to apply an existing policy to them, you simply add the site identifiers to the site list. You can also change policy rules without having to manually modify the prefixes, VPNs, or other things that the rules apply to.

Table 2: List Types

List type	Usage
data-prefix-list	Used in data-policy to define prefix and upper layer ports, either individually or jointly, for traffic matching.
prefix-list	Used in control-policy to define prefixes for matching RIB entries.
site-list	Used in control-policy to match source sites, and in apply-policy to define sites for policy application.
tloc-list	Used in control-policy to define TLOCs for matching RIB entries and to apply redefined TLOCs to vRoutes.
vpn-list	Used in control-policy to define prefixes for matching RIB entries, and in data-policy and app-route-policy to define VPNs for policy application.

The following configuration shows the types of Cisco SD-WAN Controller policy lists:

```
policy
    lists
     data-prefix-list app1
      ip-prefix 209.165.200.225/27 port 100
    !
    prefix-list pfx1
     ip-prefix 209.165.200.225/27
    !
    site-list site1
     site-id 100
    !
    tloc-list site1-tloc
     tloc 209.165.200.225 color mpls
    vpn-list vpn1
     vpn1
    !
   !
```

Policy Definition

The policy definition is where you create the policy rules. You specify match conditions (route-related properties for control policy and data-related fields for data policy) and actions to perform when a match occurs. A policy contains match–action pairings that are numbered and that are examined in sequential order. When a match occurs, the action is performed, and the policy analysis on that route or packet terminates. Some types of policy definitions apply only to specific VPNs.

Table 3: Policy Types

Policy type	Usage
policy-type	Can be **control-policy**, **data-policy**, or **vpn-menbership**—dictates the type of policy. Each type has a particular syntax and a particular set of match conditions and settable actions.
vpn-list	Used by data-policy and app-route-policy to list the VPNs for which the policy is applicable.
sequence	Defines each sequential step of the policy by sequence number.
match	Decides what entity to match on in the specific policy sequence.
action	Determines the action that corresponds to the preceding match statement.
default-action	Action to take for any entity that is not matched in any sequence of the policy. By default, the action is set to reject.

The following configuration shows the components of the Cisco SD-WAN Controller policy definition. These items are listed in the logical order you should use when designing policy, and this order is also how the items are displayed in the configuration, regardless of the order in which you add them to the configuration.

```
policy
    policy-type name
     vpn-list vpn-list
     sequence number
       match
        <route | tloc vpn | other>
    !
    action <accept reject drop>
     set attribute value
    !
    default-action <reject accept>
    !
   !
  !
```

Policy Application

The following are the configuration components:

Component	Usage
site-list	Determines the sites to which a given policy is applies. The direction (**in** | **out**) applies only to control-policy.
policy-type	The policy type can be **control-policy**, **data-policy**, or **vpn-membership**—and name refer to an already configured policy to be applied to the sites specified in the site-list for the section.

For a policy definition to take effect, you associate it with sites in the overlay network.

```
apply-policy
 site-list name
  control-policy name <inout>
 !
 site-list name
  data-policy name
  vpn-membership name
  !
 !
```

Policy Example

For a complete policy, which consists of lists, policy definition, and policy application. The example illustrated below creates two lists (a site-list and a tloc-list), defines one policy (a control policy), and applies the policy to the site-list. In the figure, the items are listed as they are presented in the node configuration. In a normal configuration process, you create lists first (group together all the things you want to use), then define the policy itself (define what things you want to do), and finally apply the policy (specify the sites that the configured policy affects).

```
apply-policy
 site-list site1  --------→ Apply the defined policy towards the sites in site-list
  control-policy prefer_local out
  !
policy
 lists
 site-list site1
  site-id 100
 tloc-list prefer_site1 ---→ Define the lists required for apply-policy and for use within
 the policy
  tloc 192.0.2.1 color mols encap ipsec preference 400
 control-policy prefer_local
  sequence 10
   match route
    site-list sitele ------->Lists previously defined used within policy
  !
   action accept
    set
     tloc-list prefer_site
    !
   !
  !
```

TLOC Attributes Used in Policies

A transport location, or TLOC, defines a specific interface in the overlay network. Each TLOC consists of a set of attributes that are exchanged in OMP updates among the Cisco IOS XE Catalyst SD-WAN devices.

Each TLOC is uniquely identified by a 3-tuple of IP address, color, and encapsulation. Other attributes can be associated with a TLOC.

The TLOC attributes listed below can be matched or set in Cisco SD-WAN Controller policies.

Table 4:

TLOC Attribute	Function	Application Point Set By	Application Point Modify By
Address (IP address)	system-ip address of the source device on which the interface is located.	Configuration on source device	control-policy data-policy
Carrier	Identifier of the carrier type. It primarily indicates whether the transport is public or private.	Configuration on source device	control-policy
Color	Identifier of the TLOC type.	Configuration on source device	control-policy data-policy
Domain ID	Identifier of the overlay network domain.	Configuration on source device	control-policy
Encapsulation	Tunnel encapsulation, either IPsec or GRE.	Configuration on source device	control-policy data-policy
Originator	system-ip address of originating node.	Configuration on any originator	control-policy
Preference	OMP path-selection preference. A higher value is a more preferred path.	Configuration on source device	control-policy
Site ID	Identification for a give site. A site can have multiple nodes or TLOCs.	Configuration on source device	control-policy
Tag	Identifier of TLOC on any arbitrary basis.	Configuration on source device	control-policy

Cisco Catalyst SD-WAN Route Attributes Used in Policies

A Cisco Catalyst SD-WAN route, defines a route in the overlay network and is similar to a standard IP route, has a TLOC and VPN attributes. The Cisco vEdge devices exchange routes in OMP updates.

The routes attributes listed below can be matched or set in Cisco SD-WAN Controller policies.

Table 5:

Route Attribute	Function	Application Point Set By	Application Point Modify By
Origin	Source of the route, either BGP, OSPF, connected, static.	Source device	control-policy

Route Attribute	Function	Application Point Set By	Application Point Modify By
Originator	Source of the update carrying the route.	Any originator	control-policy
Preference	OMP path-selection preference. A higher value is a more preferred path.	Configuration on source device or policy	control-policy
Service	Advertised service associated with the route.	Configuration on source device	control-policy
Site ID	Identifier for a give site. A site can have multiple nodes or TLOCs.	Configuration on source device	control-policy
Tag	Identification on any arbitrary basis.	Configuration on source device	control-policy
TLOC	TLOC used as next hop for the route.	Configuration on source device or policy	control-policy data-policy
VPN	VPN to which the route belongs.	Configuration on source device or policy	control-policy data-policy

Design Cisco Catalyst SD-WAN Controller Policy Processing and Application

Understanding how a Cisco SD-WAN Controller policy is processed and applied allows for proper design of policy and evaluation of how policy is implemented across the overlay network.

Policy is processed as follows:

- A policy definition consists of a numbered, ordered sequence of match–action pairings. Within each policy, the pairings are processed in sequential order, starting with the lowest number and incrementing.
- As soon as a match occurs, the matched entity is subject to the configured action of the sequence and is then no longer subject to continued processing.
- Any entity not matched in a sequence is subject to the default action for the policy. By default, this action is reject.

Cisco SD-WAN Controller policy is applied on a per-site-list basis, so:

- When applying policy to a site-list, you can apply only one of each type of policy. For example, you can have one control-policy and one data-policy, or one control-policy in and one control-policy out. You cannot have two data policies or two outbound control policies.
- Because a site-list is a grouping of many sites, you should be careful about including a site in more than one site-list. When the site-list includes a range of site identifiers, ensure that there is no overlap. If the same site is part of two site-lists and the same type of policy is applied to both site-lists, the policy behavior is unpredictable and possibly catastrophic.

- Control-policy is unidirectional, being applied either inbound to the Cisco SD-WAN Controller or outbound from it. When control-policy is needed in both directions, configure two control policies.
- Data-policy is bidirectional and can be applied either to traffic received from the service side of the Cisco vEdge device, traffic received from the tunnel side, or all of these combinations.
- VPN membership policy is always applied to traffic outbound from the Cisco SD-WAN Controller.
- Control-policy remains on the Cisco SD-WAN Controller and affects routes that the controller sends and receives.
- Data-policy is sent to either the Cisco vEdge devices in the site-list. The policy is sent in OMP updates, and it affects the data traffic that the devices send and receive.
- When any node in the overlay network makes a routing decision, it uses any and all available routing information. In the overlay network, it is the Cisco Catalyst SD-WAN Controller that distributes routing information to the Cisco vEdge device nodes.
- In a network deployment that has two or more Cisco Catalyst SD-WAN Controllers, each controller acts independently to disseminate routing information to other Cisco SD-WAN Controllers and to Cisco vEdge devices in the overlay network. So, to ensure that the Cisco SD-WAN Controller policy has the desired effect in the overlay network, each Cisco SD-WAN Controller must be configured with the same policy, and the policy must be applied identically. For any given policy, you must configure the identical policy and apply it identically across all the Cisco SD-WAN Controllers.

Note When you deploy a policy, the deployment status is updated only for 30 minutes, which is the timeout limit for policies. After the timeout period, the deployment task status is not monitored. If you are deploying a bigger policy with more number of lines, and if it takes more than 30 minutes, the task status will not be monitored.

Cisco Cisco Catalyst SD-WAN Controller Policy Operation

At a high level, control policy operates on routing information, which in the Cisco Catalyst SD-WAN network is carried in OMP updates. Data policy affects data traffic, and VPN membership controls the distribution of VPN routing tables.

The basic Cisco SD-WAN Controller policies are:

- Control Policy
- Data Policy
- VPN Membership

Control Policy

Control policy, which is similar to standard routing policy, operates on routes and routing information in the control plane of the overlay network. Centralized control policy, which is provisioned on the Cisco SD-WAN Controller, is the Cisco Catalyst SD-WAN technique for customizing network-wide routing decisions that determine or influence routing paths through the overlay network. Local control policy, which is provisioned

on a Cisco vEdge device, allows customization of routing decisions made by BGP and OSPF on site-local branch or enterprise networks.

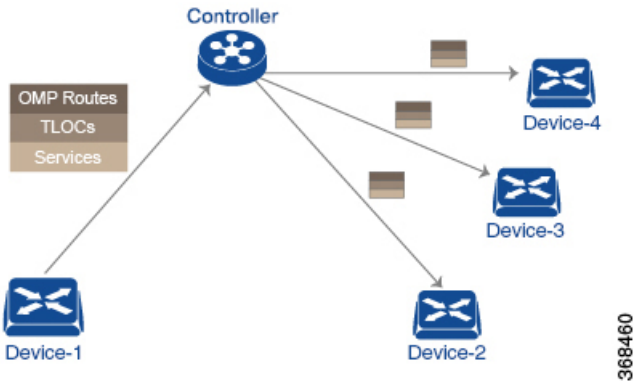
The routing information that forms the basis of centralized control policy is carried in Cisco Catalyst SD-WAN route advertisements, which are transmitted on the DTLS or TLS control connections between Cisco SD-WAN Controllers and Cisco vEdge devices. Centralized control policy determines which routes and route information are placed into the centralized route table on the Cisco SD-WAN Controller and which routes and route information are advertised to the Cisco vEdge devices in the overlay network. Basic centralized control policy establish traffic engineering, to set the path that traffic takes through the network. Advanced control policy supports a number of features, including service chaining, which allows Cisco vEdge devices in the overlay network to share network services, such as firewalls and load balancers.

Centralized control policy affects the OMP routes that are distributed by the Cisco SD-WAN Controller throughout the overlay network. The Cisco SD-WAN Controller learns the overlay network topology from OMP routes that are advertised by the Cisco vEdge devices over the OMP sessions inside the DTLS or TLS connections between the Cisco SD-WAN Controller and the devices.

Three types of OMP routes carry the information that the Cisco SD-WAN Controller uses to determine the network topology:

- Cisco Catalyst SD-WAN OMP routes, which are similar to IP route advertisements, advertise routing information that the devices have learned from their local site and the local routing protocols (BGP and OSPF) to the Cisco SD-WAN Controller. These routes are also referred to as OMP routes or Routes.
- TLOC routes carry overlay network–specific locator properties, including the IP address of the interface that connects to the transport network, a link color, which identifies a traffic flow, and the encapsulation type. (A TLOC, or transport location, is the physical location where a Cisco vEdge device connects to a transport network. It is identified primarily by IP address, link color, and encapsulation, but a number of other properties are associated with a TLOC.)
- Service routes advertise the network services, such as firewalls, available to VPN members at the local site.

Figure 7: Control Policy Topology



By default, no centralized control policy is provisioned. In this bare, unpolicied network, all OMP routes are placed in the Cisco SD-WAN Controller's route table as is, and the Cisco SD-WAN Controller advertises all OMP routes, as is, to all the devices in the same VPN in the network domain.

By provisioning centralized control policy, you can affect which OMP routes are placed in the Cisco SD-WAN Controller's route table, what route information is advertised to the devices, and whether the OMP routes are modified before being put into the route table or before being advertised.

Cisco vEdge devices place all the route information learned from the Cisco SD-WAN Controllers, as is, into their local route tables, for use when forwarding data traffic. Because the Cisco SD-WAN Controller's role is to be the centralized routing system in the network, Cisco vEdge devices can never modify the OMP route information that they learn from the Cisco SD-WAN Controllers.

The Cisco SD-WAN Controller regularly receives OMP route advertisements from the devices and, after recalculating and updating the routing paths through the overlay network, it advertises new routing information to the devices.

The centralized control policy that you provision on the Cisco SD-WAN Controller remains on the Cisco SD-WAN Controller and is never downloaded to the devices. However, the routing decisions that result from centralized control policy are passed to the devices in the form of route advertisements, and so the affect of the control policy is reflected in how the devices direct data traffic to its destination.

A type of centralized control policy called service chaining allows data traffic to be routed through one or more network services, such as firewall, load balancer, and intrusion detection and prevention (IDP) devices, en route to its destination.

Localized control policy, which is provisioned locally on the devices, is called route policy. This policy is similar to the routing policies that you configure on a regular driver, allowing you to modify the BGP and OSPF routing behavior on the site-local network. Whereas centralized control policy affects the routing behavior across the entire overlay network, route policy applies only to routing at the local branch.

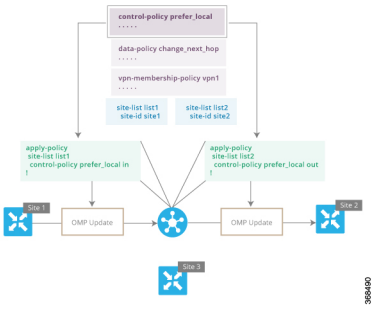
The Cisco Catalyst SD-WAN devices periodically exchange OMP updates, which carry routing information pertaining to the overlay network. Two of the things that these updates contain are Route attributes and Transport Locations (TLOC) attributes.

The Cisco SD-WAN Controller uses these attributes from the OMP updates to determine the topology and status of the overlay network, and installs routing information about the overlay network into its route table. The controller then advertises the overlay topology to the Cisco vEdge devices in the network by sending OMP updates to them.

Control policy examines the Route and TLOC attributes carried in OMP updates and can modify attributes that match the policy. Any changes that results from control policy are applied directionally, either inbound or outbound.

The figure shows a control-policy named **prefer_local** that is configured on a Cisco SD-WAN Controller and that is applied to Site 1 (via site-list list1) and to Site 2 (via site-list list2).

Figure 8: Control Policy Topology




```
Device# apply-policy
site-list list1
control-policy prefer_local in
!
```

The upper left arrow shows that the policy is applied to Site 1—more specifically, to **site-list list1**, which contains an entry for Site 1. The command **control-policy prefer_local in** is used to apply the policy to OMP updates that are coming in to the Cisco SD-WAN Controller from the Cisco vEdge device, which is inbound from the perspective of the controller. The **in** keyword indicates an **inbound** policy. So, for all OMP updates that the Site 1 devices send to the Cisco SD-WAN Controller, the "prefer_local" control policy is applied before the updates reach the route table on the Cisco SD-WAN Controller. If any Route or TLOC attributes in an OMP update match the policy, any changes that result from the policy actions occur before the Cisco SD-WAN Controller installs the OMP update information into its route table.

The route table on the Cisco SD-WAN Controller is used to determine the topology of the overlay network. The Cisco SD-WAN Controller then distributes this topology information, again via OMP updates, to all the devices in the network. Because applying policy in the inbound direction influences the information available to the Cisco SD-WAN Controller. It determines the network topology and network reachablity, modifying Route and TLOC attributes before they are placed in the controller's route table.

```
apply-policy
site-list list2
control-policy prefer_local out
!
```

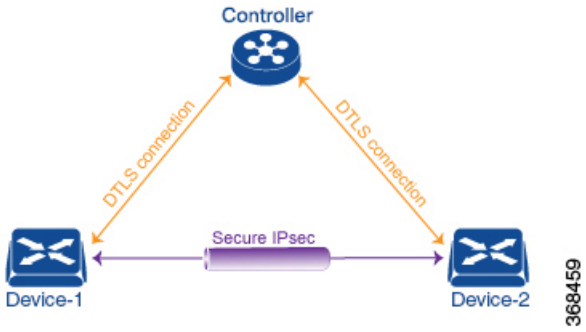
On the right side of the figure above, the "prefer_local" policy is applied to Site 2 via the **control-policy prefer_local out** command. The **out** keyword in the command indicates an **outbound policy**, which means that the policy is applied to OMP updates that the Cisco SD-WAN Controller is sending to the devices at Site 2. Any changes that result from the policy occur, after the information from the Cisco SD-WAN Controller's route table is placed in to an OMP update and before the devices receive the update. Again, note that the direction is outbound from the perpspective of the Cisco SD-WAN Controller.

In contrast to an inbound policy, which affects the centralized route table on the Cisco SD-WAN Controller and has a broad effect on the route attributes advertised to all the devices in the overlay network. A control policy applied in the outbound direction influences only the route tables on the individual devices included in the site-list.

The same control policy (the **prefer_local** policy) is applied to both the inbound and outbound OMP updates. However, the effects of applying the same policy to inbound and outbound are different. The usage shown in the figure illustrates the flexibility of the Cisco Catalyst SD-WAN control policy design architecture and configuration.

Data Policy

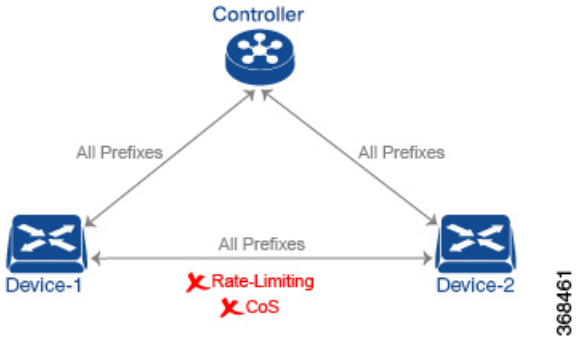
Data policy influences the flow of data traffic traversing the network based either on fields in the IP header of packets or the router interface on which the traffic is being transmitted or received. Data traffic travels over the IPsec connections between Cisco vEdge devices, shown in purple in the adjacent figure.



The Cisco Catalyst SD-WAN architecture implements two types of data policy:

- Centralized data policy controls the flow of data traffic based on the source and destination addresses and ports and DSCP fields in the packet's IP header (referred to as a 5-tuple), and based on network segmentation and VPN membership. These types of data policy are provisioned centrally, on the Cisco SD-WAN Controller, and they affect traffic flow across the entire network.
- Localized data policy controls the flow of data traffic into and out of interfaces and interface queues on a Cisco vEdge device. This type of data policy is provisioned locally using access lists. It allows you to classify traffic and map different classes to different queues. It also allows you to mirror traffic and to police the rate at which data traffic is transmitted and received.

By default, no centralized data policy is provisioned. The result is that all prefixes within a VPN are reachable from anywhere in the VPN. Provisioning centralized data policy allows you to apply a 6-tuple filter that controls access between sources and destinations.



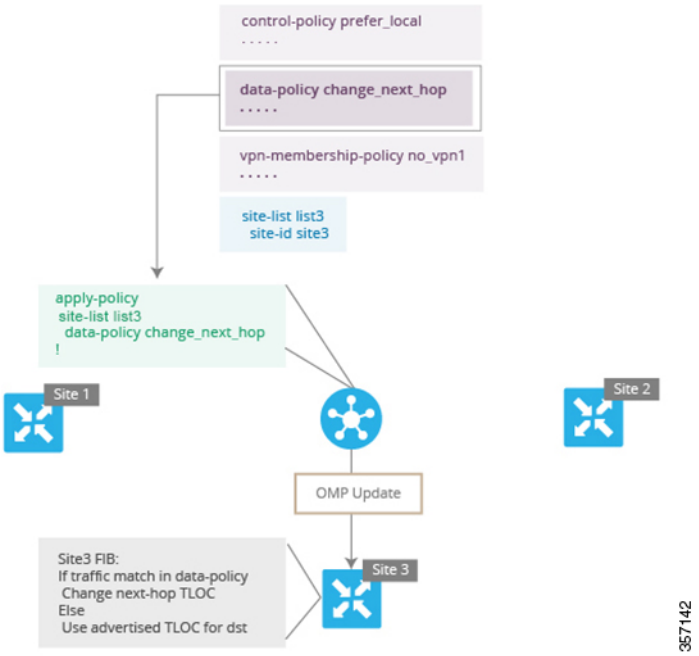
As with centralized control policy, you provision a centralized data policy on the Cisco SD-WAN Controller, and that configuration remains on the Cisco SD-WAN Controller. The effects of data policy are reflected in how the Cisco vEdge devices direct data traffic to its destination. Unlike control policy, however, centralized data polices are pushed to the devices in a read-only fashion. They are not added to the router's configuration file, but you can view them from the CLI on the router.

With no access lists provisioned on a Cisco vEdge device, all data traffic is transmitted at line rate and with equal importance, using one of the interface's queues. Using access lists, you can provision class of service, which allows you to classify data traffic by importance, spread it across different interface queues, and control the rate at which different classes of traffic are transmitted. You can provision policing. You can also provision packet mirroring.

Data policy examines fields in the headers of data packets, looking at the source and destination addresses and ports, and the protocol and DSCP values, and for matching packets, it can modify the next hop in a variety of ways or apply a policer to the packets. Data policy is configured and applied on the Cisco SD-WAN Controller, and then it is carried in OMP updates to the Cisco vEdge devices in the site-list that the policy is applied to. The match operation and any resultant actions are performed on the devices as it transmits or receives data traffic.

In the Data Policy Topology figure, a data policy named “change_next_hop” is applied to a list of sites that includes Site 3. The OMP update that the Cisco SD-WAN Controller sends to the devices at Site 3 includes this policy definition. When the device sends or receives data traffic that matches the policy, it changes the next hop to the specified TLOC. Non-matching traffic is forwarded to the original next-hop TLOC.

Figure 9: Data Policy Topology



In the **apply-policy** command for a data policy, specify a direction from the perspective of the device. The "all" direction in the figure applies the policy to incoming and outgoing data traffic transiting the tunnel interface. You can limit the span of the policy to only incoming traffic with a **data-policy change_next_hop from-tunnel** command or to only outgoing traffic with a **data-policy change_next_hop from-service** command.

VPN Membership Policy Operation

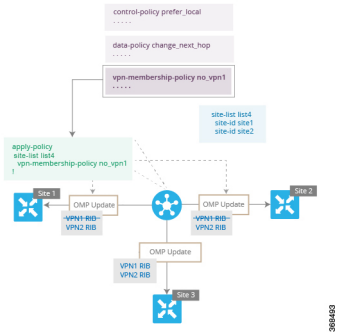
VPN membership policy, as the name implies, affects the VPN route tables that are distributed to particular Cisco vEdge devices. In an overlay network with no VPN membership policy, the Cisco Catalyst SD-WAN Controller pushes the routes for all VPNs to all the devices. If your business usage model restricts participation of specific devices in particular VPNs, a VPN membership policy is used to enforce this restriction.

The figure VPN Membership Topology illustrates how VPN membership policy works. This topology has three Cisco vEdge devices:

- The Cisco vEdge devices at Sites 1 and 2 service only VPN 2.
- The Cisco vEdge devices at Site 3 services both VPN 1 and VPN 2.

In the figure, the device at Site 3 receives all route updates from the Cisco SD-WAN Controller, because these updates are for both VPN 1 and VPN 2. However, because the other Cisco vEdge devices service only VPN 2, it can filter the route updates sent to them, remove the routes associated with VPN 1 and sends only the ones that apply to VPN 2.

Figure 10: VPN Membership Topology

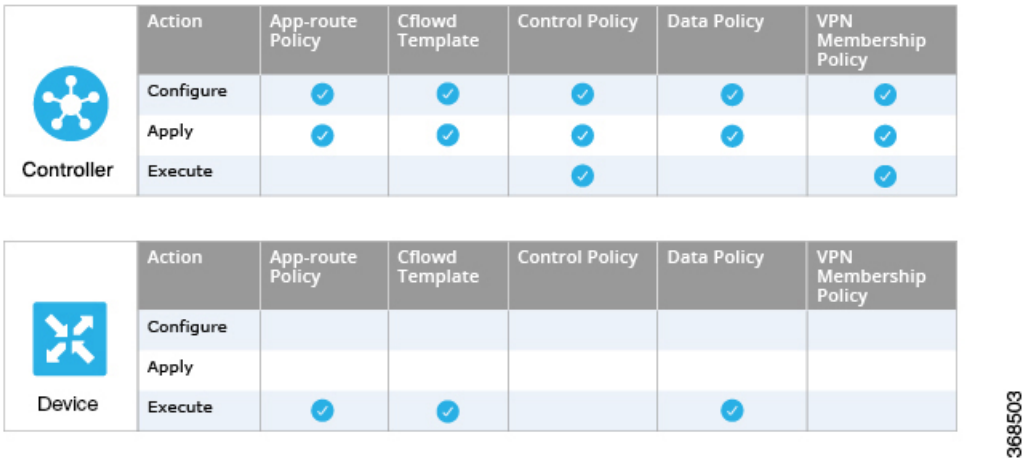


Notice that here, direction is not set when applying VPN membership policy. The Cisco SD-WAN Controller always applies this type of policy to the OMP updates that it sends outside to the Cisco vEdge devices.

Configure and Execute Cisco SD-WAN Controller Policies

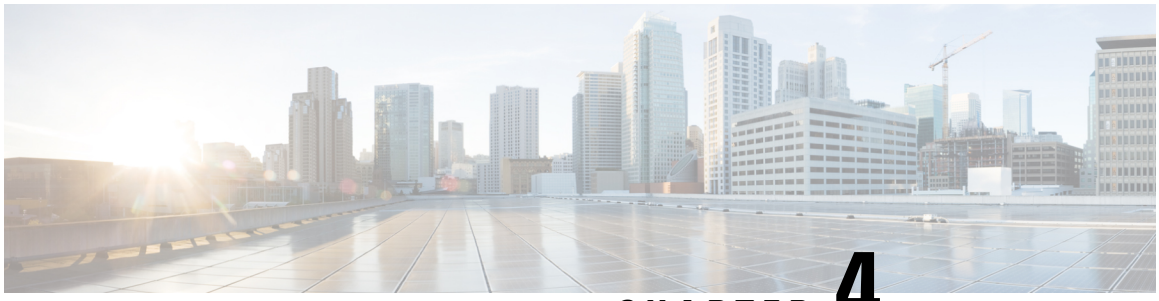
All Cisco SD-WAN Controller policies are configured on the Cisco vEdge devices, using a combination of policy definition and lists. All Cisco SD-WAN Controller policies are also applied on the Cisco vEdge devices, with a combination of apply-policy and lists. However, where the actual Cisco SD-WAN Controller policy executes depends on the type of policy, as shown in this figure:

Figure 11: Cisco SD-WAN Controller Policy



For control policy and VPN membership policy, the entire policy configuration remains on the Cisco SD-WAN Controller, and the actions taken as a result of routes or VPNs that match a policy are performed on the Cisco SD-WAN Controller.

For the other three policy types—application-aware routing, cflowd templates, and data policy—the policies are transmitted in OMP updates to the Cisco vEdge devices, and any actions taken as a result of the policies are performed on the devices.



CHAPTER 4

Centralized Policy

Note To achieve simplification and consistency, the Cisco SD-WAN solution has been rebranded as Cisco Catalyst SD-WAN. In addition, from Cisco IOS XE SD-WAN Release 17.12.1a and Cisco Catalyst SD-WAN Release 20.12.1, the following component changes are applicable: **Cisco vManage** to **Cisco Catalyst SD-WAN Manager**, **Cisco vAnalytics** to **Cisco Catalyst SD-WAN Analytics**, **Cisco vBond** to **Cisco Catalyst SD-WAN Validator**, **Cisco vSmart** to **Cisco Catalyst SD-WAN Controller**, and **Cisco Controllers** to **Cisco Catalyst SD-WAN Control Components**. See the latest Release Notes for a comprehensive list of all the component brand name changes. While we transition to the new names, some inconsistencies might be present in the documentation set because of a phased approach to the user interface updates of the software product.

The topics in this section provide overview information about the different types of centralized policies, the components of centralized policies, and how to configure centralized policies using Cisco SD-WAN Manager or the CLI.

Overview of Centralized Policies

Centralized policies refer to policies that are provisioned on Cisco SD-WAN Controllers, which are the centralized controllers in the Cisco Catalyst SD-WAN overlay network.

Types of Centralized Policies

Centralized Control Policy

Centralized control policy applies to the network-wide routing of traffic by affecting the information that is stored in the Cisco Catalyst SD-WAN Controller's route table and that is advertised to the Cisco vEdge devices. The effects of centralized control policy are seen in how Cisco vEdge devices direct the overlay network's data traffic to its destination.

Note The centralized control policy configuration itself remains on the Cisco Catalyst SD-WAN Controller and is never pushed to local devices.

Centralized Data Policy

Centralized data policy applies to the flow of data traffic throughout the VPNs in the overlay network. These policies can permit and restrict access based either on a 6-tuple match (source and destination IP addresses and ports, DSCP fields, and protocol) or on VPN membership. These policies are pushed to the selected Cisco vEdge devices.

Centralized Data Policy Based on Packet Header Fields

Policy decisions affecting data traffic can be based on the packet header fields, specifically, on the source and destination IP prefixes, the source and destination IP ports, the protocol, and the DSCP.

This type of policy is often used to modify traffic flow in the network. Here are some examples of the types of control that can be effected with a centralized data policy:

- Which set of sources are allowed to send traffic to any destination outside the local site. For example, local sources that are rejected by such a data policy can communicate only with hosts on the local network.
- Which set of sources are allowed to send traffic to a specific set of destinations outside the local site. For example, local sources that match this type of data policy can send voice traffic over one path and data traffic over another.
- Which source addresses and source ports are allowed to send traffic to any destination outside the local site or to a specific port at a specific destination.

Configure Centralized Policies Using Cisco SD-WAN Manager

To configure a centralized policy, use the Cisco SD-WAN Manager policy configuration wizard. The wizard consists of the following operations that guide you through the process of creating and editing policy components:

- Create Groups of Interest: Create lists that group together related items and that you call in the match or action components of a policy.
- Configure Topology and VPN Membership: Create the network structure to which the policy applies.
- Configure Traffic Rules: Create the match and action conditions of a policy.
- Apply Policies to Sites and VPNs: Associate the policy with sites and VPNs in the overlay network.
- Activate the centralized policy.

  For a centralized policy to take effect, you must activate the policy.

To configure centralized policies using Cisco SD-WAN Manager, use the steps identified in the procedures that follow this section.

Start the Policy Configuration Wizard

To start the policy configuration wizard:

1. From the Cisco SD-WAN Manager menu, choose **Configuration** > **Policies**.
2. Click **Centralized Policy**.
3. Click **Add Policy**.

   The policy configuration wizard appears, and the **Create Groups of Interest** window is displayed.

Configure Groups of Interest for Centralized Policy

In **Create Groups of Interest**, create new groups of list types as described in the following sections to use in a centralized policy:

Configure Application

1. In the groups of interest list, click **Application** list type.
2. Click **New Application List**.
3. Enter a name for the list.
4. Choose either **Application** or **Application Family**.

   Application can be the names of one or more applications, such as **Third Party Control**, **ABC News**, **Mircosoft Teams**, and so on. The Cisco vEdge devices support about 2300 different applications. To list the supported applications, use the **?** in the CLI.

   Application Family can be one or more of the following: **antivirus**, **application-service**, **audio_video**, **authentication**, **behavioral**, **compression**, **database**, **encrypted**, **erp**, **file-server**, **file-transfer**, **forum**, **game**, **instant-messaging**, **mail**, **microsoft-office**, **middleware**, **network-management**, **network-service**, **peer-to-peer**, **printer**, **routing**, **security-service**, **standard**, **telephony**, **terminal**, **thin-client**, **tunneling**, **wap**, **web**, and **webmail**.
5. In the **Select** drop-down, in the 'Search' filter, select the required applications or application families.
6. Click **Add**.

A few application lists are preconfigured. You cannot edit or delete these lists.

Microsoft_Apps—Includes Microsoft applications, such as Excel, Skype, and Xbox. To display a full list of Microsoft applications, click the list in the Entries column.

Google_Apps—Includes Google applications, such as Gmail, Google maps, and YouTube. To display a full list of Google applications, click the list in the Entries column.

Configure Color

1. In the groups of interest list, click **Color**.
2. Click **New Color List**.
3. Enter a name for the list.
4. In the **Select Color** drop-down, in the 'Search' filter select the required colors.

Colors can be: 3g, biz-internet, blue, bronze, custom1 through custom3, default, gold, green, lte, metro-ethernet, mpls, private1 through private6, public-internet, red, and silver.

5. Click **Add**.

To configure multiple colors in a single list, you can select multiple colors from the drop-down.

Configure Community

Table 6: Feature History

Feature Name	Release Information	Description
Ability to Match and Set Communities	Cisco SD-WAN Release 20.5.1 Cisco IOS XE Catalyst SD-WAN Release 17.5.1a Cisco vManage Release 20.5.1	This feature lets you match and set communities using a control policy. Control policies are defined and applied on devices to manipulate communities. With this feature, you can match and assign single or multiple BGP community tags to your prefixes based on which routing policies can be manipulated.

A community list is used to create groups of communities to use in a match clause of a route map. A community list can be used to control which routes are accepted, preferred, distributed, or advertised. You can also use a community list to set, append, or modify the communities of a route.

1. In the group of interest list, click **Community**.
2. Click **New Community List**.
3. Enter a name for the community list.
4. Choose either **Standard** or **Expanded**.
   - Standard community lists are used to specify communities and community numbers.
   - Expanded community lists are used to filter communities using a regular expression. Regular expressions are used to specify patterns to match community attributes.
5. In the **Add Community** field, enter one or more data prefixes separated by commas in any of the following formats:
   - *aa*:*nn*: Autonomous System (AS) number and network number. Each number is a 2-byte value with a range from 1 to 65535.
   - **internet**: Routes in this community are advertised to the internet community. This community comprises all BGP-speaking networking devices.
   - **local-as**: Routes in this community are not advertised outside the local AS number.
   - **no-advertise**: Attaches the NO_ADVERTISE community to routes. Routes in this community are not advertised to other BGP peers.
   - **no-export**: Attaches the NO_EXPORT community to routes. Routes in this community are not advertised outside the local AS or outside a BGP confederation boundary. To configure multiple BGP communities in a single list, include multiple **community** options, specifying one community in each option.

6. Click **Add**.

Configure Data Prefix

1. In the **Groups of Interest** list, click **Data Prefix**.
2. Click **New Data Prefix List**.
3. Enter a name for the list.
4. Choose either **IPv4** or **IPv6**.
5. In the **Add Data Prefix** field, enter one or more data prefixes separated by commas.
6. Click **Add**.

Configure Policer

1. In the groups of interest list, click **Policer**.
2. Click **New Policer List**.
3. Enter a name for the list.
4. Define the policing parameters:
   a. In the **Burst** field, enter the maximum traffic burst size, a value from 15,000 to 10,000,000 bytes.
   b. In the **Exceed** field, select the action to take when the burst size or traffic rate is exceeded. It can be **drop**, which sets the packet loss priority (PLP) to **low**.

      You can use the **remark** action to set the packet loss priority (PLP) to **high**.
   c. In the **Rate** field, enter the maximum traffic rate, a value from 0 through $2^{64} - 1$ bits per second (bps).
5. Click **Add**.

Configure Prefix

1. In the groups of interest list, click **Prefix**.
2. Click **New Prefix List**.
3. Enter a name for the list.
4. In the **Add Prefix** field, enter one or more data prefixes separated by commas.
5. Click **Add**.

Configure Site

1. In the groups of interest list, click **Site**.
2. Click **New Site List**.
3. Enter a name for the list.
4. In the **Add Site** field, enter one or more site IDs separated by commas.

For example, 100 or 200 separated by commas or in the range, 1- 4294967295.

5. Click **Add**.

Configure App Probe Class

1. In the groups of interest list, click **App Probe Class**.
2. Click **New App Probe Class**.
3. Enter the probe class name in the **Probe Class Name** field.
4. Select the required forwarding class from the **Forwarding Class** drop-down list.
5. In the **Entries** pane, select the appropriate color from the **Color** drop-down list and enter the **DSCP** value.

   You can add more entries if needed by clicking on the + symbol.
6. Click **Save**.

Configure SLA Class

1. In the groups of interest list, click **SLA Class**.
2. Click **New SLA Class List**.
3. Enter a name for the list.
4. Define the SLA class parameters:
   a. In the **Loss** field, enter the maximum packet loss on the connection, a value from 0 through 100 percent.
   b. In the **Latency** field, enter the maximum packet latency on the connection, a value from 0 through 1,000 milliseconds.
   c. In the **Jitter** field, enter the maximum jitter on the connection, a value from 1 through 1,000 milliseconds.
   d. Select the required app probe class from the **App Probe Class** drop-down list.
5. (Optional) Select the **Fallback Best Tunnel** checkbox to enable the best tunnel criteria.

   This optional filed is available from Cisco SD-WAN Release 20.5.1 to pick the best path or color from the available colors when SLA is not met. When this option is selected, you can choose the required criteria from the drop-down. The criteria are a combination of one or more of losses, latency, and, jitter values.
6. Select the **Criteria** from the drop-down list. The available criteria are:
   - Latency
   - Loss
   - Jitter
   - Latency, Loss
   - Latency, Jitter

- Loss, Latency
- Loss, Jitter
- Jitter, Latency
- Jitter, Loss
- Latency, Loss, Jitter
- Latency, Jitter, Loss
- Loss, Latency, Jitter
- Loss, Jitter, Latency
- Jitter, Latency, Loss
- Jitter, Loss, Latency

7. Enter the **Loss Variance (%)**, **Latency Variance (ms)**, and the **Jitter Variance (ms)** for the selected criteria.
8. Click **Add**.

Configure TLOC

1. In the groups of interest list, click **TLOC**.
2. Click **New TLOC List**. The **TLOC List** popup displays.
3. Enter a name for the list.
4. In the **TLOC IP** field, enter the system IP address for the TLOC.
5. In the **Color** field, select the TLOC's color.
6. In the **Encap** field, select the encapsulation type.
7. In the **Preference** field, optionally select a preference to associate with the TLOC.

   The range is 0 to 4294967295.
8. Click **Add TLOC** to add another TLOC to the list.
9. Click **Save**.

Note To use the `set tloc` and `set tloc-list` commands, you must use the set-vpn command.

For each TLOC, specify its address, color, and encapsulation. Optionally, set a preference value (from 0 to 232 – 1) to associate with the TLOC address. When you apply a TLOC list in an action accept condition, when multiple TLOCs are available and satisfy the match conditions, the TLOC with the highest preference value is used. If two or more TLOCs have the highest preference value, traffic is sent among them in an ECMP fashion.

If IPsec preference is set on the local preferred color for an edge router, the local TLOC and the color does not overlap with the centralized policy configured with local color preference. The edge router with local TLOC preference takes the precedence. In this case, the preferred TLOC configured in centralized policy is not considered.

Configure VPN

1. In the groups of interest list, click **VPN**.
2. Click **New VPN List**.
3. Enter a name for the list.
4. In the **Add VPN** field, enter one or more VPN IDs separated by commas.

   For example, 100 or 200 separated by commas or in the range, 1- 65530.
5. Click **Add**.

Configure Region

Minimum release: Cisco vManage Release 20.7.1

To configure a list of regions for Multi-Region Fabric (formerly Hierarchical SD-WAN), ensure that Multi-Region Fabric is enabled in **Administration** > **Settings**.

1. In the groups of interest list, click **Region**.
2. Click **New Region List**.
3. In the **Region List Name** field, enter a name for the region list.
4. In the **Add Region** field, enter one or more regions, separated by commas, or enter a range.

   For example, specify regions 1, 3 with commas, or a range 1-4.
5. Click **Add**.

Click **Next** to move to **Configure Topology and VPN Membership** in the wizard.

Integrating WAN Insight (WANI) into Cisco SD-WAN Manager

Table 7: Feature History

Feature Name	Release Information	Description
WAN Insight Policy Automation	Cisco IOS XE Catalyst SD-WAN Release 17.12.1a Cisco Catalyst SD-WAN Manager Release 20.12.1	With this feature, you can apply the recommendations that are available on Cisco SD-WAN Analytics toCisco SD-WAN Manager AAR policy and view the applied recommendations on Cisco SD-WAN Manager.

Cisco SD-WAN Analytics is a cloud-based analytics service for Cisco Catalyst SD-WAN offering comprehensive insights into application and network performance. The analytics service is available with

Cisco DNA Advantage and Cisco DNA Premier software subscriptions. Cisco SD-WAN Analytics collects and stores metadata about traffic flows in its cloud storage and produces analytics based on this collected data. Predictive Path Analytics generates recommendations for path based on long term insights. These recommendations need to be converted into policy created manually on Cisco SD-WAN Manager and then applied to the network.

The Predictive Path Recommendations feature allows you to apply active recommendations to the actionable centralized AAR policy to influence the forwarding decisions in the Cisco Catalyst SD-WAN network. The recommendations are applied as a part of the AAR policy and then pushed to Cisco SD-WAN Controller. The Predictive Path Recommendations are applied to the SD-WAN network as TLOC preferences in AAR policies.

For more information about using Predictive Path Recommendations, see Predictive Path Recommendations.

Apply Predictive Path Recommendations

When there are predictive path recommendations in Cisco SD-WAN Analytics, perform the following steps to apply the recommendations to the Application-Aware routing policies:

1. In the Cisco SD-WAN Manager menu, click the bell icon at the top-right corner. The **Notifications** pane is displayed with active alarms.
2. If there are any **Active Recommendations** in the **Notifications** pane, click on the site to view the recommendations. Alternatively, you can view from the Cisco SD-WAN Manager menu, click **Analytics** > **Predictive Networks**.
3. Click **Active Recommendations**, and then click **Apply**.
4. In the **Apply Predictive Path Recommendations** window, click **Proceed to Apply** to apply new recommendations.

   You can review the applied recommendations in the Cisco SD-WAN Manager generated configs and push the recommendations to Cisco SD-WAN Controller.

Points to Consider

- Cisco SD-WAN Manager pulls recommendations when you log in. If you want to update the recommendations, refresh the page or log in again.
- Cisco SD-WAN Manager support recommendations for application lists which are associated with some AAR policy only. If AAR Policy does not exist for a given application list, the recommendations are not valid and policy processing is not done.
- WAN Insights generates recommendations for standard App Groups even when the AAR Policy is not defined. However, the policy automation is not done since AAR policy is not defined.
- When for the same site and application list, if WANI generates a terminate for a recommendation which is applied and also generates another recommendation, the recommendations are applied based on the preferences.
- Application of WANI recommendations for Cloud OnRamp for SaaS is not supported.

Predictive Path Recommendations

WAN Insights (WANI) allows you to track the performance of your current network setup and tune your policies and paths to achieve the best user experience. Predictive path recommendations influence AAR policy TLOC preferences.

WAN Insights is a predictive network optimization tool that uses a statistical model to examine historical data from Cisco Catalyst SD-WAN, in order to find the best paths for application traffic. WANI analyzes the telemetry data exported during application traffic flows, and then generates long-term recommendations for paths that would reduce the probability of experiencing an SLA violation (for example, low-quality performance).

Predictive network associates some SLA with each application list that is defined in the AAR policy in order to detect SLA violations for the applications. This is used to calculate a probability of SLA violation on a given site and TLOC and generates recommendations.

For more information about configuring group of interest for data policies, see Configure Groups of Interest for Centralized Policy.

Configure Topology and VPN Membership

When you first open the **Configure Topology and VPN Membership** window, the **Topology** window is displayed by default.

To configure topology and VPN membership:

Hub-and-Spoke

1. In the **Add Topology** drop-down, select **Hub-and-Spoke**.
2. Enter a name for the hub-and-spoke policy.
3. Enter a description for the policy.
4. In the **VPN List** field, select the VPN list for the policy.
5. In the left pane, click **Add Hub-and-Spoke**. A hub-and-spoke policy component containing the text string **My Hub-and-Spoke** is added in the left pane.
6. Double-click the **My Hub-and-Spoke** text string, and enter a name for the policy component
7. In the right pane, add hub sites to the network topology:
   a. Click **Add Hub Sites**.
   b. In the **Site List** field, select a site list for the policy component.
   c. Click **Add**.
   d. Repeat these steps to add more hub sites to the policy component.
8. In the right pane, add spoke sites to the network topology:
   a. Click **Add Spoke Sites**.
   b. In the **Site List Field**, select a site list for the policy component.
   c. Click **Add**.
   d. Repeat these steps to add more spoke sites to the policy component.
9. Repeat steps as needed to add more components to the hub-and-spoke policy.
10. Click **Save Hub-and-Spoke Policy**.

Mesh

1. In the **Add Topology** drop-down, select **Mesh**.
2. Enter a name for the mesh region policy component.
3. Enter a description for the mesh region policy component.
4. In the **VPN List** field, select the VPN list for the policy.
5. Click **New Mesh Region**.
6. In the **Mesh Region Name** field, enter a name for the individual mesh region.
7. In the **Site List** field, select one or more sites to include in the mesh region.
8. Click **Add**.
9. Repeat these steps to add more mesh regions to the policy.
10. Click **Save Mesh Topology**.

Custom Control (Route & TLOC): Centralized route control policy (for matching OMP routes)

1. In the **Add Topology** drop-down, select **Custom Control (Route & TLOC)**.
2. Enter a name for the control policy.
3. Enter a description for the policy.
4. In the left pane, click **Sequence Type**. The **Add Custom Control Policy** popup displays.
5. Select **Route**. A policy component containing the text string **Route** is added in the left pane.
6. Double-click the **Route** text string, and enter a name for the policy component.
7. In the right pane, click **Sequence Rule**. The **Match/Actions** box opens, and **Match** is selected by default.
8. From the boxes under the **Match** box, select the desired policy match type. Then select or enter the value for that match condition. Configure additional match conditions for the sequence rule, as desired.
9. Click **Actions**. The **Reject** option is selected by default. To configure actions to perform on accepted packets, click the **Accept** option. Then select the action or enter a value for the action.
10. Click **Save Match and Actions**.
11. Click **Sequence Rule** to configure more sequence rules, as desired. Drag and drop to re-order them.
12. Click **Sequence Type** to configure more sequences, as desired. Drag and drop to re-order them.
13. Click **Save Control Policy**.

Custom Control (Route & TLOC): Centralized TLOC control policy (for matching TLOC routes)

1. In the **Add Topology** drop-down, select **Custom Control (Route & TLOC)**.
2. Enter a name for the control policy.
3. Enter a description for the policy.
4. In the left pane, click **Sequence Type**. The **Add Custom Control Policy** popup displays.

5. Select **TLOC**. A policy component containing the text string **TLOC** is added in the left pane.
6. Double-click the **TLOC** text string, and enter a name for the policy component.
7. In the right pane, click **Sequence Rule**. The **Match/Actions** box opens, and **Match** is selected by default.
8. From the boxes under the **Match** box, select the desired policy match type. Then select or enter the value for that match condition. Configure additional match conditions for the sequence rule, as desired.
9. Click **Actions**. The **Reject** option is selected by default. To configure actions to perform on accepted packets, click the **Accept** option. Then select the action or enter a value for the action.
10. Click **Save Match and Actions**.
11. Click **Sequence Rule** to configure more sequence rules, as desired. Drag and drop to re-order them.
12. Click **Sequence Type** to configure more sequences, as desired. Drag and drop to re-order them.
13. Click **Save Control Policy**.

A centralized control policy contains sequences of match–action pairs. The sequences are numbered to set the order in which a route or TLOC is analyzed by the match–action pairs in the policy.

Note Sequence can have either **match app-list** or **dns-app-list** configured for a policy, but not both. Configuring both **match app-list** and **dns-app-list** for a policy is not supported.

NAT DIA fallback and DNS redirection are not supported at the same time in data policy.

Each sequence in a centralized control policy can contain one match condition (either for a route or for a TLOC) and one action condition.

Default Action

If a selected route or TLOC does not match any of the match conditions in a centralized control policy, a default action is applied to it. By default, the route or TLOC is rejected.

If a selected data packet does not match any of the match conditions in a data policy, a default action is applied to the packet. By default, the data packet is dropped.

Import Existing Topology

1. In the **Add Topology** drop-down, click **Import Existing Topology**. The **Import Existing Topology** popup appears.
2. Select the type of topology.
3. For **Policy Type**, choose the name of the topology you want to import.
4. In the **Policy** drop-down, select a policy to import.

Note The policy configuration wizard does not let you import an already configured policy as in other instances of centralized policies (data, control, or application-aware routing). The policy must be configured in its entirety.

5. Click **Import**.

Click **Next** to move to **Configure Traffic Rules** in the wizard.

Create a VPN Membership Policy

1. In the **Specify your network topology** area, click **VPN Membership**.
2. Click **Add VPN Membership Policy**.

Note You can add only one VPN membership at a time, therefore all site lists and VPN lists must be included in a single policy.

The **Add VPN Membership Policy** popup displays.

3. Enter a name and description for the VPN membership policy.
4. In the **Site List** field, select the site list.
5. In the **VPN Lists** field, select the VPN list.
6. Click **Add List** to add another VPN to the VPN membership.
7. Click **Save**.
8. Click **Next** to move to **Configure Traffic Rules** in the wizard.

Configure Traffic Rules

Table 8: Feature History

Feature Name	Release Information	Description
Policy Matching with ICMP Message	Cisco SD-WAN Release 20.4.1 Cisco vManage Release 20.4.1	This feature provides support for a new match condition that you can use to specify a list of ICMP messages for centralized data policies, localized data policies, and Application-Aware Routing policies.

When you first open the **Configure Traffic Rules** window, **Application-Aware Routing** is selected by default.

You can also view already created AAR routing policies listed in the page. It provides various information related to the policies such as the Name of the policy, Type, Mode, Description, Update By, and Last Updated details.

Note You can refer to the Mode column for the security status details of the policy. The status helps to differentiate whether the policy is used in unified security or not. The mode status is applicable only for security policies and not relevant to any centralized or localized policies.

For more information on configuring traffic rules for the Cisco Catalyst SD-WAN Application Intelligence Engine (SAIE) flow, see Cisco Catalyst SD-WAN Application Intelligence Engine Flow.

Note In Cisco vManage Release 20.7.1 and earlier releases, the SAIE flow is called the deep packet inspection (DPI) flow.

To configure traffic rules for a centralized data policy:

1. Click **Traffic Data**.
2. Click the **Add Policy** drop-down.
3. Click **Create New**. The **Add Data Policy** window displays.
4. Enter a name and a description for the data policy.
5. In the right pane, click **Sequence Type**. The **Add Data Policy** popup opens.
6. Select the type of data policy you want to create, **Application Firewall**, **QoS**, **Service Chaining**, **Traffic Engineering**, or **Custom**.

Note If you want to configure multiple types of data policies for the same match condition, you need to configure a custom policy.

7. A policy sequence containing the text string **Application**, **Firewall**, **QoS**, **Service Chaining**, **Traffic Engineering**, or **Custom** is added in the left pane.
8. Double-click the text string, and enter a name for the policy sequence. The name you type is displayed both in the Sequence Type list in the left pane and in the right pane.
9. In the right pane, click **Sequence Rule**. The **Match/Action** box opens, and **Match** is selected by default. The available policy match conditions are listed below the box.

Match Condition	Procedure
None (match all packets)	Do not specify any match conditions.
Applications /Application Family List	**a.** In the **Match** conditions, click **Applications/Application Family List**. **b.** In the drop-down, select the application family. **c.** To create an application list: **1.** Click **New Application List**. **2.** Enter a name for the list. **3.** Click **Application** to create a list of individual applications. Click **Application Family** to create a list of related applications. **4.** In the **Select Application** drop-down, select the desired applications or application families. **5.** Click **Save**.

Match Condition	Procedure
Destination Data Prefix	**a.** In the **Match** conditions, click **Destination Data Prefix**. **b.** To match a list of destination prefixes, select the list from the drop-down. **c.** To match an individual destination prefix, enter the prefix in the **Destination: IP Prefix** field.
Destination Port	**a.** In the **Match** conditions, click **Destination Port**. **b.** In the **Destination Port** field, enter the port number. Specify a single port number, a list of port numbers (with numbers separated by a space), or a range of port numbers (with the two numbers separated with a hyphen [-]).
DNS Application List	Add an application list to enable split DNS. **a.** In the **Match** conditions, click **DNS Application List**. **b.** In the drop-down, select the application family.
DNS	Add an application list to process split DNS. **a.** In the **Match** conditions, click **DNS**. **b.** In the drop-down, select **Request** to process DNS requests for the DNS applications, and select **Response** to process DNS responses for the applications.
DSCP	**a.** In the **Match** conditions, click **DSCP**. **b.** In the **DSCP** field, type the DSCP value, a number from 0 through 63.
Packet Length	**a.** In the **Match** conditions, click **Packet Length**. **b.** In the **Packet Length** field, type the length, a value from 0 through 65535.
PLP	**a.** In the **Match** conditions, click **PLP** to set the **Packet Loss Priority**. **b.** In the **PLP** drop-down, select **Low** or **High**. To set the PLP to **High**, apply a policer that includes the **exceed remark** option.
Protocol	**a.** In the **Match** conditions, click **Protocol**. **b.** In the **Protocol** field, type the Internet Protocol number, a number from 0 through 255.
ICMP Message	To match ICMP messages, in the Protocol field, set the Internet Protocol Number to 1, or 58, or both. **Note** This field is available from , Cisco SD-WAN Release 20.4.1 Cisco vManage Release 20.4.1.

Match Condition	Procedure
Source Data Prefix	**a.** In the **Match** conditions, click **Source Data Prefix**. **b.** To match a list of source prefixes, select the list from the drop-down. **c.** To match an individual source prefix, enter the prefix in the **Source** field.
Source Port	**a.** In the **Match** conditions, click **Source Port**. **b.** In the **Source** field, enter the port number. Specify a single port number, a list of port numbers (with numbers separated by a space), or a range of port numbers (with the two numbers separated with a hyphen [-]).
TCP	**a.** In the **Match** conditions, click **TCP**. **b.** In the **TCP** field, **syn** is the only option available.

10. For QoS and Traffic Engineering data policies: From the **Protocol** drop-down list, select **IPv4** to apply the policy only to IPv4 address families, **IPv6** to apply the policy only to IPv6 address families, or **Both** to apply the policy to IPv4 and IPv6 address families.
11. To select one or more **Match** conditions, click its box and set the values as described.

Note Not all match conditions are available for all policy sequence types.

12. To select actions to take on matching data traffic, click the **Actions** box.
13. To drop matching traffic, click **Drop**. The available policy actions are listed in the right side.
14. To accept matching traffic, click **Accept**. The available policy actions are listed in the right side.
15. Set the policy action as described.

Note Not all actions are available for all match conditions.

Note If IPv4 packet contains non-initial fragment of UDP or TCP datagram, it has no L4 ports information available because there is no UDP or TCP header. For such fragments destination-port or source-port match is ignored.

In the following example, all the UDP packets to destination port 161 and any other IPv4 packets having protocol ID field in IPv4 header set to 17 with IPv4 header having fragment-offset set will be dropped.

```
policy
 app-visibility
 access-list SDWAN_101
  sequence 100
   match
    destination-port 161
    protocol         17
   !
   action drop
   !
  !
```

Action Condition	Description	Procedure
Counter	Count matching data packets.	**a.** In the **Action** conditions, click **Counter**. **b.** In the **Counter Name** field, enter the name of the file in which to store packet counters.
DSCP	Assign a DSCP value to matching data packets.	**a.** In the **Action** conditions, click **DSCP**. **b.** In the **DSCP** field, type the DSCP value, a number from 0 through 63.
Forwarding Class	Assign a forwarding class to matching data packets.	**a.** In the **Match** conditions, click **Forwarding Class**. **b.** In the **Forwarding Class** field, type the class value, which can be up to 32 characters long.
Log	Minimum release: Cisco vManage Release 20.11.1 and Cisco IOS XE Release 17.11.1a Click **Log** to enable logging. When (DP, AAR or ACL) data policy packets are configured with log action, logs generated and logged to syslog. Due to the global **log-rate-limit**, not all logs are logged. A syslog message is generated the first time a packet header is logged and then every 5 minutes thereafter, as long as the flow is active.	**a.** In the **Action** conditions, click **Log** to enable logging.
Policer	Apply a policer to matching data packets.	**a.** In the **Match** conditions, click **Policer**. **b.** In the **Policer** drop-down field, select the name of a policer.

Action Condition	Description	Procedure
Loss Correction	Apply loss correction to matching data packets. Forward Error Correction (FEC) recovers lost packets on a link by sending redundant data, enabling the receiver to correct errors without the need to request retransmission of data. FEC is supported only for IPSEC tunnels, it is not supported for GRE tunnels. • **FEC Adaptive** – Corresponding packets are subjected to FEC only if the tunnels that they go through have been deemed unreliable based on measured loss. Adaptive FEC starts to work at 2% packet loss; this value is hard-coded and is not configurable. If you choose **FEC Adaptive**, an additional field, **Loss Threshold**, displays that allows you to specify the packet loss threshold for automatically enabling FEC. Adaptive FEC starts to work at 2% packet loss; this value is configurable. You can specify a loss threshold of 1 to 5%. The default packet loss threshold is 2%. • **FEC Always** – Corresponding packets are always subjected to FEC. • **Packet Duplication** – Sends duplicate packets over a single tunnel. If more than one tunnel is available, duplicated packets will be sent over the tunnel with the best parameters.	**a.** In the **Match** conditions, click **Loss Correction**. **b.** In the **Loss Correction** field, select **FEC Adaptive**, **FEC Always**, or **Packet Duplication**.
Click **Save Match and Actions**.		

16. Create additional sequence rules as desired. Drag and drop to re-arrange them.
17. Click **Save Data Policy**.
18. Click **Next** to move to **Apply Policies to Sites and VPNs** in the wizard.

Match Parameters - Control Policy

For OMP and TLOC routes , you can match the following attributes:

Match Condition	Description
Color List	One or more colors. The available colors are: 3g, biz-internet, blue, bronze, custom1,custom2, custom3, default, gold, green, lte, metro-ethernet, mpls, private1 through private6, public-internet, red and silver.

Match Condition	Description
Community List	List of one or more BGP communities. In the **Community List** field, you can specify: • *aa*:*nn*: AS number and network number. Each number is a 2-byte value with a range from 1 to 65535. • **internet**: Routes in this community are advertised to the internet community. This community comprises all BGP-speaking networking devices. • **local-as**: Routes in this community are not advertised outside the local AS. • **no-advertise**: Attach the NO_ADVERTISE community to routes. Routes in this community are not advertised to other BGP peers. • **no-export**: Attach the NO_EXPORT community to routes. Routes in this community are not advertised outside the local AS or outside a BGP confederation boundary. To configure multiple BGP communities in a single list, include multiple **community** options, specifying one community in each option.
Types	Specifies the community type. Choose **Standard** to specify communities and community numbers or, **Expanded** to filter communities using a regular expression. Regular expressions are used to specify patterns to match community attributes.
OMP Tag	Tag value associated with the route or prefix in the routing database on the device. The range is 0 through 4294967295.
Origin	Protocol from which the route was learned.
Originator	IP address from which the route was learned.

Match Condition	Description
Path Type	In a Hierarchical SD-WAN architecture, match a route by its path type, which can be one of the following: • **Hierarchical Path**: A route that includes hops from an access region to a border router, through region 0, to another border router, then to an edge router in a different access region • **Direct Path**: A direct path route from one edge router to another edge router. • **Transport Gateway Path**: A route that is re-originated by a router that has transport gateway functionality enabled. **Note** This option is available beginning with Cisco vManage Release 20.8.1.
Preference	How preferred a prefix is. This is the preference value that the route or prefix has in the local site, that is, in the routing database on the device. A higher preference value is more preferred.The range is 0 through 255.
Prefix List	One or more prefixes. Specifies the name of a prefix list.
Not available in Cisco SD-WAN Manager.	Individual site identifier. The range is 0 through 4294967295.
Site	One or more overlay network site identifiers.
Region	Region defined for Hierarchical SD-WAN. The range is 1 to 63. **Note** This option is available beginning with Cisco vManage Release 20.7.1.
Role	In a Hierarchical SD-WAN architecture, match by the device type, which can be **Border Router** or **Edge Router**. **Note** This option is available beginning with Cisco vManage Release 20.8.1.
TLOC	Individual TLOC address. **Note** To use the `set tloc` and `set tloc-list` commands, you must use the set-vpn command.

Match Condition	Description
VPN	Individual VPN identifier. The range is 0 through 65535.
Carrier	Carrier for the control traffic. Values are: default, carrier1 through carrier8.
Domain ID	Domain identifier associated with a TLOC. The range is 0 through 4294967295.
OMP Tag	Tag value associated with the TLOC route in the route table on the device. The range is 0 through 4294967295.
Site	Individual site contributor or more overlay network site identifiers.. The range is 0 through 4294967295.

In the CLI, you configure the OMP route attributes to match with the **policy control-policy sequence match route** command, and you configure the TLOC attributes to match with the **policy control-policy sequence match tloc** command.

Match Parameters - Data Policy

A centralized data policy can match IP prefixes and fields in the IP headers, as well as applications. You can also enable split DNS.

Each sequence in a policy can contain one or more match conditions.

Table 9:

Match Condition	Description
Omit	Match all packets.
Applications/Application Family List	Applications or application families.
Destination Data Prefix	Group of destination prefixes, IP prefix and prefix length. The range is 0 through 65535; specify a single port number, a list of port numbers (with numbers separated by a space), or a range of port numbers (with the two numbers separated with a hyphen [-]).

Match Condition	Description
Destination Region	Choose one of the following: • **Primary**: Match traffic if the destination device is in the same primary region (also called access region) as the source. This traffic reaches the destination using a multi-hop path, through the core region. • **Secondary**: Match traffic if the destination device is not in the same primary region as the source but is within the same secondary region as the source. This traffic can reach the destination using a direct tunnel, as described for secondary regions. • **Other**: Match traffic if the destination device is not in the same primary region or secondary region as the source. This traffic requires a multi-hop path from the source to the destination. **Note** Minimum releases: Cisco vManage Release 20.9.1, Cisco IOS XE Catalyst SD-WAN Release 17.9.1a
DNS Application List	Enables split DNS, to resolve and process DNS requests and responses on an application-by-application basis. Name of an **app-list** list . This list specifies the applications whose DNS requests are processed.
DNS	Specify the direction in which to process DNS packets. To process DNS requests sent by the applications (for outbound DNS queries), specify **dns request**. To process DNS responses returned from DNS servers to the applications, specify **dns response**.
DSCP	Specifies the DSCP value.
Packet length	Specifies the packet length. The range is 0 through 65535; specify a single length, a list of lengths (with numbers separated by a space), or a range of lengths (with the two numbers separated with a hyphen [-]).
Packet Loss Priority (PLP)	Specifies the packet loss priority. By default, packets have a PLP value of **low**. To set the PLP value to **high**, apply a policer that includes the **exceed remark** option.
Protocol	Specifies Internet protocol number. The range is 0 through 255.
ICMP Message	For Protocol IPv4 when you enter a Protocol value as 1, the **ICMP Message** field displays where you can select an ICMP message to apply to the data policy. Likewise, the **ICMP Message** field displays for Protocol IPv6 when you enter a Protocol value as 58. When you select Protocol as Both, the **ICMP Message or ICMPv6 Message** field displays. **Note** This field is available from , Cisco SD-WAN Release 20.4.1 Cisco vManage Release 20.4.1.
Source Data Prefix	Specifies the group of source prefixes or an individual source prefix.
Source Port	Specifies the source port number. The range is 0 through 65535; specify a single port number, a list of port numbers (with numbers separated by a space), or a range of port numbers (with the two numbers separated with a hyphen [-]).
TCP Flag	Specifies the TCP flag, syn.
Traffic To	In a Multi-Region Fabric architecture, match border router traffic flowing to the access region that the border router is serving, the core region, or a service VPN. **Note** Minimum release: Cisco vManage Release 20.8.1

Note If IPv4 packet contains non-initial fragment of UDP or TCP datagram, it has no L4 ports information available because there is no UDP or TCP header. For such fragments destination-port or source-port match is ignored.

In the following example, all the UDP packets to destination port 161 and any other IPv4 packets having protocol ID field in IPv4 header set to 17 with IPv4 header having fragment-offset set will be dropped.

```
policy
 app-visibility
 access-list SDWAN_101
  sequence 100
   match
    destination-port 161
    protocol         17
   !
   action drop
   !
  !
```

Table 10: ICMP Message Types/Codes and Corresponding Enumeration Values

Type	Code	Enumeration
0	0	echo-reply
3		unreachable
	0	net-unreachable
	1	host-unreachable
	2	protocol-unreachable
	3	port-unreachable
	4	packet-too-big
	5	source-route-failed
	6	network-unknown
	7	host-unknown
	8	host-isolated
	9	dod-net-prohibited
	10	dod-host-prohibited
	11	net-tos-unreachable
	12	host-tos-unreachable
	13	administratively-prohibited
	14	host-precedence-unreachable
	15	precedence-unreachable

5		redirect
	0	net-redirect
	1	host-redirect
	2	net-tos-redirect
	3	host-tos-redirect
8	0	echo
9	0	router-advertisement
10	0	router-solicitation
11		time-exceeded
	0	ttl-exceeded
	1	reassembly-timeout
12		parameter-problem
	0	general-parameter-problem
	1	option-missing
	2	no-room-for-option
13	0	timestamp-request
14	0	timestamp-reply
40	0	photuris
42	0	extended-echo
43		extended-echo-reply
	0	echo-reply-no-error
	1	malformed-query
	2	interface-error
	3	table-entry-error
	4	multiple-interface-match

Table 11: ICMPv6 Message Types/Codes and Corresponding Enumeration Values

Type	Code	Enumeration

1		unreachable
	0	no-route
	1	no-admin
	2	beyond-scope
	3	destination-unreachable
	4	port-unreachable
	5	source-policy
	6	reject-route
	7	source-route-header
2	0	packet-too-big
3		time-exceeded
	0	hop-limit
	1	reassembly-timeout
4		parameter-problem
	0	Header
	1	next-header
	2	parameter-option
128	0	echo-request
129	0	echo-reply
130	0	mld-query
131	0	mld-report
132	0	mld-reduction
133	0	router-solicitation
134	0	router-advertisement
135	0	nd-ns
136	0	nd-na
137	0	redirect
138		router-renumbering
	0	renum-command
	1	renum-result
	255	renum-seq-number

139		ni-query
	0	ni-query-v6-address
	1	ni-query-name
	2	ni-query-v4-address
140		ni-response
	0	ni-response-success
	1	ni-response-refuse
	2	ni-response-qtype-unknown
141	0	ind-solicitation
142	0	ind-advertisement
143		mldv2-report
144	0	dhaad-request
145	0	dhaad-reply
146	0	mpd-solicitation
147	0	mpd-advertisement
148	0	cp-solicitation
149	0	cp-advertisement
151	0	mr-advertisement
152	0	mr-solicitation
153	0	mr-termination
155	0	rpl-control

Action Parameters - Control Policy

For each match condition, you configure a corresponding action to take if the route or TLOC matches for a control policy.

In the CLI, you configure actions with the **policy control-policy action** command.

Each sequence in a centralized control policy can contain one action condition.

In the action, you first specify whether to accept or reject a matching route or TLOC:

Table 12:

Description	Cisco SD-WAN Manager
Accept the route. An accepted route is eligible to be modified by the additional parameters configured in the **action** portion of the policy configuration.	Click **Accept**.

Description	Cisco SD-WAN Manager
Discard the packet.	Click **Reject**.

Then, for a route or TLOC that is accepted, you can configure the following actions:

Action Condition	Description
Export To	Export the route the the specified VPN or list of VPNs (for a **match route** match condition only). The range is 0 through 65535 or list name.
OMP Tag	Change the tag string in the route, prefix, or TLOC. The range is 0 through 4294967295.
Preference	Change the preference value in the route, prefix, or TLOC to the specified value. A higher preference value is more preferred. The range is 0 through 255.
Service	Specify a service to redirect traffic to before delivering the traffic to its destination. The TLOC address or list of TLOCs identifies the TLOCs to which the traffic should be redirected to reach the service. In the case of multiple TLOCs, the traffic is load-balanced among them. The VPN identifier is where the service is located. Standard services: **FW**, **IDS**, **IDP** Custom services: **netsvc1**, **netsvc2**, **netsvc3**, **netsvc4** Configure the services themselves on the Cisco SD-WAN devices that are collocated with the service devices, using the **vpn service** configuration command.
TLOC	Change the TLOC address, color, and encapsulation to the specified address and color. For each TLOC, specify its address, color, and encapsulation. *address* is the system IP address. *color* can be one of **3g**, **biz-internet**, **blue**, **bronze**, **custom1**, **custom2**, **custom3**, **default**, **gold**, **green**, **lte**, **metro-ethernet**, **mpls**, **private1** through **private6**, **public-internet**, **red**, and **silver**. *encapsulation* can be **gre** or **ipsec**. Optionally, set a preference value (from 0 to 232 – 1) to associate with the TLOC address. When you apply a TLOC list in an **action accept** condition, when multiple TLOCs are available and satisfy the match conditions, the TLOC with the highest preference value is used. If two or more of TLOCs have the highest preference value, traffic is sent among them in an ECMP fashion.
TLOC Action	Direct matching routes or TLOCs using the mechanism specified by *action*, and enable end-to-end tracking of whether the ultimate destination is reachable. Setting the TLOC action option enables the Cisco Catalyst SD-WAN Controller to perform end-to-end tracking of the path to the ultimate destination device.

Note The **preference** command controls the preference for directing inbound and outbound traffic to a tunnel. The preference can be a value from 0 through 4294967295 (232 – 1), and the default value is 0. A higher value is preferred over a lower value.

When a Cisco vEdge device has two or more tunnels, if all the TLOCs have the same preference and no policy is applied that affects traffic flow, all the TLOCs are advertised into OMP. When the router transmits or receives traffic, it distributes traffic flows evenly among the tunnels, using ECMP.

Action Parameters - Data Policy

Table 13: Feature History

Feature Name	Release Information	Description
Path Preference Support for Cisco IOS XE Catalyst SD-WAN Devices	Cisco IOS XE Catalyst SD-WAN Release 17.2.1r	This feature extends to Cisco IOS XE Catalyst SD-WAN devices, support for selecting one or more local transport locators (TLOCs) for a policy action.
Traffic Redirection to SIG Using Data Policy	Cisco SD-WAN Release 20.4.1 Cisco vManage Release 20.4.1	With this feature, while creating a data policy, you can define an application list along with other match criteria and redirect the application traffic to a Secure Internet Gateway (SIG).
Next Hop Action Enhancement in Data Policies	Cisco SD-WAN Release 20.5.1 Cisco vManage Release 20.5.1	This feature enhances match action conditions in a centralized data policy for parity with the features configured on Cisco vEdge devices. When you are setting up next-hop-loose action, this feature helps to redirect application traffic to an available route when next-hop address is not available.

When data traffic matches the conditions in the match portion of a centralized data policy, the packet can be accepted or dropped. Then, you can associate parameters with accepted packets.

In the CLI, you configure the action parameters with the **policy data-policy vpn-list sequence action** command.

Each sequence in a centralized data policy can contain one action condition.

In the action, you first specify whether to accept or drop a matching data packet, and whether to count it:

Action Condition	Description
Click **Accept**	Accepts the packet. An accepted packet is eligible to be modified by the additional parameters configured in the action portion of the policy configuration.
Cflowd	Enables cflowd traffic monitoring.
Counter	Counts the accepted or dropped packets. Specifies the name of a counter. Use the **show policy access-lists counters** command on the Cisco vEdge device.
Click **Drop**	Discards the packet. This is the default action.

Action Condition	Description
Log	Minimum release: Cisco IOS XE Catalyst SD-WAN Release 17.11.1a and Cisco vManage Release 20.11.1 Click **Log** to enable logging. When (DP, AAR or ACL) data policy packets are configured with log action, logs generated and logged to syslog. Due to the global **log-rate-limit**, not all logs are logged. A syslog message is generated the first time a packet header is logged and then every 5 minutes thereafter, as long as the flow is active. For information on **policy log-rate-limit** CLI, see **policy log-rate-limit** command in the Cisco Catalyst SD-WAN Qualified Command Reference Guide.
Redirect DNS	Redirects DNS requests to a particular DNS server. Redirecting requests is optional, but if you do so, you must specify both actions. For an inbound policy, **redirect-dns host** allows the DNS response to be correctly forwarded back to the requesting service VPN. For an outbound policy, specify the IP address of the DNS server. **Note** When you upgrade to releases later than Cisco IOS XE Catalyst SD-WAN Release 17.7.1a, you must configure redirect DNS through **nat use-vpn 0** to redirect DNS to Direct Internet Interface (DIA). **Note** You can set only local TLOC preferences with redirect-dns as actions on the same sequence, but not remote TLOC. **Note** You cannot configure Redirect DNS and SIG at the same time. NAT DIA fallback and DNS redirection are not supported at the same time in data policy.
TCP Optimization	Fine-tune TCP to decrease round-trip latency and improve throughout for matching TCP traffic.
Secure Internet Gateway	Redirect application traffic to a SIG. **Note** Before you apply a data policy for redirecting application traffic to a SIG, you must have configured the SIG tunnels. For more information on configuring Automatic SIG tunnels, see Automatic Tunnels. For more information on configuring Manual SIG tunnels, see Manual Tunnels.

Then, for a packet that is accepted, the following parameters can be configured:

Action Condition	Description
Cflowd	Enables cflowd traffic monitoring.
NAT Pool or **NAT VPN**	Enables NAT functionality, so that traffic can be redirected directly to the internet or other external destination. You can configure up to 31 (1–31) NAT pools per router.
DSCP	DSCP value. The range is 0 through 63.
Forwarding Class	Name of the forwarding class.
Local TLOC	Enables sending packets to one of the TLOCs that matches the color and encapsulation. The available colors are: 3g, biz-internet, blue, bronze, custom1,custom2, custom3, default, gold, green, lte, metro-ethernet, mpls, private1 through private6, public-internet, red and silver. The encapsulation options are: **ipsec** and **gre**. By default, if the TLOC is not available, traffic is forwarded using an alternate TLOC. To drop traffic if a TLOC is unavailable, include the **restrict** option. By default, encapsulation is **ipsec**.
Next Hop	Sets the next hop IP address to which the packet should be forwarded. **Note** Starting from Cisco SD-WAN Release 20.5.1 and Cisco vManage Release 20.5.1, the **Use Default Route when Next Hop is not available** field is available next to the **Next Hop** action parameter. This option is available only when the sequence type is **Traffic Engineering** or **Custom**, and the protocol is either **IPv4** or **IPv6**, but not both.
Policer	Applies a policer. Specifies the name of policer configured with the **policy policer** command.
Service	Specifies a service to redirect traffic to before delivering the traffic to its destination. The TLOC address or list of TLOCs identifies the remote TLOCs to which the traffic should be redirected to reach the service. In the case of multiple TLOCs, the traffic is load-balanced among them. The VPN identifier is where the service is located. Standard services: **FW**, **IDS**, **IDP** Custom services: **netsvc1**, **netsvc2**,**netsvc3**, **netsvc4** TLOC list is configured with a **policy lists tloc-list** list. Configure the services themselves on the Cisco vEdge devices that are collocated with the service devices, using the **vpn service** command.

Action Condition	Description
TLOC	Direct traffic to a remote TLOC that matches the IP address, color, and encapsulation of one of the TLOCs in the list. If a preference value is configured for the matching TLOC, that value is assigned to the traffic.
Click **Accept**, then action **VPN**.	Set the VPN that the packet is part of. The range is 0 through 65530.

Note Data policies are applicable on locally generated packets, including routing protocol packets, when the match conditions are generic.

Example configuration:

```
sequence 21
   match
    source-ip 10.0.0.0/8
   action accept
```

In such situations, it may be necessary to add a sequence in the data policy to escape the routing protocol packets. For example to skip OSPF, use the following configuration:

```
sequence 20
   match
    source-ip 10.0.0.0/8
    protocol  89
   action accept
sequence 21
   match
    source-ip 10.0.0.0/8
   action accept
```

The following table describes the IPv4 and IPv6 actions.

Table 14:

IPv4 Actions	IPv6 Actions
drop, dscp, next-hop (from-service only)/vpn, count, forwarding class, policer (only in interface ACL), App-route SLA (only)	N/A
App-route preferred color, app-route sla strict, cflowd, nat, redirect-dns	N/A
N/A	drop, dscp, next-hop/vpn, count, forwarding class, policer (only in interface ACL) App-route SLA (only), App-route preferred color, app-route sla strict
policer (DataPolicy), tcp-optimization, fec-always,	policer (DataPolicy)
tloc, tloc-list (set tloc, set tloc-list)	tloc, tloc-list (set tloc, set tloc-list)
App-Route backup-preferred color, local-tloc, local-tloc-list	App-Route backup-preferred color, local-tloc, local-tloc-list

Apply Policies to Sites and VPNs

In the **Apply Policies to Sites and VPNs** page, apply a policy to sites and VPNs:

1. In the **Policy Name** field, enter a name for the policy. This field is mandatory and can contain only uppercase and lowercase letters, the digits 0 through 9, hyphens (–), and underscores (_). It cannot contain spaces or any other characters.
2. In the **Policy Description** field, enter a description of the policy. It can contain up to 2048 characters. This field is mandatory, and it can contain any characters and spaces.
3. Associate the policy with VPNs and sites. The choice of VPNs and sites depends on the type of policy block:
   a. For a **Topology** policy block, click **New Site List**, **Inbound Site List**, **Outbound Site List**, or **VPN List**. Some topology blocks might have no **Add** buttons. Choose one or more site lists, and choose one or more VPN lists. Click **Add**.
   b. For an **Application-Aware Routing** policy block, click **New Site List** and **VPN list**. Choose one or more site lists, and choose one or more VPN lists. Click **Add**.
   c. For a **Traffic Data** policy block, click **New Site List and VPN List**. Choose the direction for applying the policy (**From Service**, **From Tunnel**, or **All**), choose one or more site lists, and choose one or more VPN lists. Click **Add**.
   d. For a cflowd policy block, click **New Site List**. Choose one or more site lists, and click **Add**.
4. Click **Preview** to view the configured policy. The policy appears in CLI format.
5. Click **Save Policy**. The **Configuration** > **Policies** page appears, and the policies table includes the newly created policy.

NAT Fallback on Cisco IOS XE Catalyst SD-WAN Devices

	Release Information	
NAT Fallback on Cisco IOS XE Catalyst SD-WAN Devices	Cisco IOS XE Release 17.3.2 Cisco vManage Release 20.3.2	Cisco IOS XE Catalyst SD-WAN devices support the NAT fallback feature for Direct Internet Access (DIA). The NAT fallback feature provides a routing-based mechanism for all traffic that is sent to the DIA route to use an alternative route when required. With this release, fallback is supported on the service and tunnel side.

Note To use Cisco SD-WAN Manager to configure NAT DIA fallback, Cisco SD-WAN Manager must manage your Cisco Catalyst SD-WAN Controller.

To enable NAT fallback using Cisco SD-WAN Manager, create and configure a data policy by doing the following:

1. From the Cisco SD-WAN Manager menu, choose **Configuration** > **Policies**.
2. From the **Custom** options drop-down, under **Centralized Policy**, select **Traffic Policy**.
3. Click **Traffic Data**.
4. From the **Add Policy** drop-down, click**Create New**.
5. Click **Sequence Type** and select **Custom**.
6. Click **(+) Sequence Rule** to create a new sequence rule.
7. After adding match conditions, click **Actions** and click **Accept**.
8. Click **NAT VPN** and select the **Fallback** checkbox.
9. Click **Save and Match Actions**.
10. Click **Save Data Policy**.

Edit your existing centralized policy and import the policy:

1. Click **Centralized Policy** and for the required centralized policy, click **...** and select **Edit**.
2. Click **Traffic Rules** and select **Traffic Data**.
3. From the **Add Policy** drop-down, select **Import Existing**.
4. Select the NAT policy that you created from the **Policy** drop-down.
5. Click **Policy Application** and select **Traffic Data**.
6. Click **+ New Site List and VPN List**.
7. Select the direction, VPN, and site as required.
8. Click **Add**.
9. Click **Save Policy Changes**.
10. Click to select **VPN**, and **Site** from the drop-down.

Note Policy configured for the **from-tunnel** traffic is also applied to the return DIA (Underlay) traffic apart from the return traffic coming over the tunnel. If none of the sequences in that policy match, it matches the default sequence in that policy.

Note NAT DIA fallback and DNS redirection are not supported at the same time in data policy.

The following NAT fallback actions/commands are now supported:

- Action: `nat fallback`

- When applying a policy: `direction from-tunnel`

Activate a Centralized Policy

Activating a centralized policy sends that policy to all connected Cisco SD-WAN Controllers. To activate a centralized policy:

1. From the Cisco SD-WAN Manager menu, choose **Configuration** > **Policies**. **Centralized Policy** is selected and displayed by default.
2. For the required policy, click **...** and select **Activate**. The **Activate Policy** popup appears. It lists the IP addresses of the reachable Cisco SD-WAN Controllers to which the policy must be applied.
3. Click **Activate**.

View Centralized Policies

To view centralized policies:

1. From the **Centralized Policy**, select a policy.
2. For a policy created using the UI policy builder or using the CLI, click **...** and select **View**. The policy created using the UI policy builder is displayed in graphical format while the policy created using the CLI method is displayed in text format.
3. For a policy created using Cisco SD-WAN Manager policy configuration wizard, click **...** and select **Preview**. This policy is displayed in text format.

Copy, Edit, and Delete Policies

To copy a policy:

1. From the **Centralized Policy**, select a policy.
2. For the desired policy, click **...** and select **Copy**.
3. In the Policy Copy popup window, enter the policy name and a description of the policy.

Note If you are upgrading to 18.4.4 version, data policy names need to be under 26 characters.

Note Starting with the Cisco SD-WAN release 19.3, 127 characters are supported for policy names for the following policy types:

- Central route policy
- Local route policy
- Local Access Control lOst (ACL)
- Local IPv6 ACL
- Central data policy
- Central app route policy
- QoS map
- Rewrite rule

All other policy names support 32 characters.

4. Click **Copy**.

To edit policies created using the Cisco SD-WAN Manager policy configuration wizard:

1. For the desired policy, click **...** and select **Edit**.
2. Edit the policy as needed.
3. Click **Save Policy Changes**.

To edit polices created using the CLI method:

1. In the **Custom Options** drop-down, click **CLI Policy**.
2. For the desired policy, click **...** and select **Edit**.
3. Edit the policy as needed.
4. Click **Update**.

To delete policies:

1. From the **Centralized Policy**, select a policy.
2. For the desired policy, click **...** and select **Delete**.
3. Click **OK** to confirm deletion of the policy.

Configure Centralized Policies Using the CLI

To configure a centralized control policy using the CLI:

1. Create a list of overlay network sites to which the centralized control policy is to be applied (in the **apply-policy** command):

```
vSmart(config)# policy
vSmart(config-policy)# lists site-list list-name
vSmart(config-lists-list-name)# site-id site-id
```

The list can contain as many site IDs as necessary. Include one **site-id** command for each site ID. For contiguous site IDs, you can specify a range of numbers separated with a dash (–). Create additional site lists, as needed.

2. Create lists of IP prefixes, TLOCs, and VPNs as needed:

```
vSmart(config)# policy lists
vSmart(config-lists)# prefix-list list-name
vSmart(config-lists-list-name)# ip-prefix prefix/length
vSmart(config)# policy lists
vSmart(config-lists)# tloc-list list-name
vSmart(config-lists-list-name)# tloc address
color color
encap encapsulation
[preference value]
vSmart(config)# policy lists
vSmart(config-lists)# vpn-list list-name
vSmart(config-lists-list-name)# vpn vpn-id
```

3. Create a control policy instance:

```
vSmart(config)# policy control-policy policy-name
vSmart(config-control-policy-policy-name)#
```

4. Create a series of match–action pair sequences:

```
vSmart(config-control-policy-policy-name)# sequence
number
vSmart(config-sequence-number)#
```

The match–action pairs are evaluated in order, by sequence number, starting with the lowest numbered pair and ending when the route matches the conditions in one of the pairs. Or if no match occurs, the default action is taken (either rejecting the route or accepting it as is).

5. Define match parameters for routes and for TLOCs:

```
vSmart(config-sequence-number)# match route route-parameter
vSmart(config-sequence-number)# match tloc tloc-parameter
```

6. Define actions to take when a match occurs:

```
vSmart(config-sequence-number)# action reject
vSmart(config-sequence-number)# action accept export-to (vpn
vpn-id | vpn-list list-name)
vSmart(config-sequence-number)# action accept set omp-tag
number
```

```
vSmart(config-sequence-number)# action accept set
preference value
```

```
vSmart(config-sequence-number)# action accept set
service service-name
(tloc ip-address |
tloc-list list-name)
[vpn vpn-id]
```

```
vSmart(config-sequence-number)# action accept set tloc
ip-address
color color
[encap encapsulation]
```

```
vSmart(config-sequence-number)# action accept set tloc-action
action

vSmart(config-sequence-number)# action accept set tloc-list list-name
```

7. Create additional numbered sequences of match–action pairs within the control policy, as needed.
8. If a route does not match any of the conditions in one of the sequences, it is rejected by default. If you want nonmatching routes to be accepted, configure the default action for the policy:

```
vSmart(config-policy-name)# default-action accept
```

9. Apply the policy to one or more sites in the Cisco Catalyst SD-WAN overlay network:

```
vSmart(config)# apply-policy site-list
list-name
control-policy
policy-name (in | out)
```

10. If the action you are configuring is a service, configure the required services on the Cisco SD-WAN devices so that the Cisco Catalyst SD-WAN Controller knows how to reach the services:

```
Device(config)# vpn vpn-id
service service-name
address ip-address
```

Specify the VPN is which the service is located and one to four IP addresses to reach the service device or devices. If multiple devices provide the same service, the device load-balances the traffic among them. Note that the Cisco SD-WAN device keeps track of the services, advertising them to the Cisco Catalyst SD-WAN Controller only if the address (or one of the addresses) can be resolved locally, that is, at the device's local site, and not learned through OMP. If a previously advertised service becomes unavailable, the Cisco SD-WAN device withdraws the service advertisement.

Following are the high-level steps for configuring a VPN membership data policy:

1. Create a list of overlay network sites to which the VPN membership policy is to be applied (in the **apply-policy** command):

```
vSmart(config)# policy
vSmart (config-policy)# lists site-list list-name
vSmart(config-lists-list-name)# site-id site-id
```

The list can contain as many site IDs as necessary. Include one **site-id** command for each site ID. For contiguous site IDs, you can specify a range of numbers separated with a dash (–). Create additional site lists, as needed.

2. Create lists of IP prefixes and VPNs, as needed:

```
vSmart(config)# policy lists
vSmart(config-lists)# data-prefix-list list-name
vSmart(config-lists-list-name)# ip-prefix prefix/length

vSmart(config)# policy lists
vSmart(config-lists)# vpn-list list-name
vSmart(config-lists-list-name)# vpn vpn-id
```

3. Create lists of TLOCs, as needed.

```
vSmart(config)# policy
vSmart(config-policy)# lists tloc-list list-name
vSmart(config-lists-list-name)# tloc ip-address color color encap encapsulation
[preference number}
```

4. Define policing parameters, as needed:

```
vSmart(config-policy)# policer policer-name
vSmart(config-policer)# rate bandwidth
vSmart(config-policer)# burst bytes
vSmart(config-policer)# exceed action
```

5. Create a data policy instance and associate it with a list of VPNs:

```
vSmart(config)# policy data-policy policy-name
vSmart(config-data-policy-policy-name)# vpn-list list-name
```

6. Create a series of match–pair sequences:

```
vSmart(config-vpn-list)# sequence number
vSmart(config-sequence-number)#
```

The match–action pairs are evaluated in order, by sequence number, starting with the lowest numbered pair and ending when the route matches the conditions in one of the pairs. Or if no match occurs, the default action is taken (either rejecting the route or accepting it as is).

7. Define match parameters for packets:

```
vSmart(config-sequence-number)# match parameters
```

8. Define actions to take when a match occurs:

```
vSmart(config-sequence-number)# action (accept | drop) [count counter-name] [log]
[tcp-optimization]
vSmart(config-sequence-number)# action acccept nat [pool number] [use-vpn 0]
vSmart(config-sequence-number)# action accept redirect-dns (host | ip-address)
vSmart(config-sequence-number)# action accept set parameters
```

9. Create additional numbered sequences of match–action pairs within the data policy, as needed.

10. If a route does not match any of the conditions in one of the sequences, it is rejected by default. To accept nonmatching prefixed, configure the default action for the policy:

```
vSmart(config-policy-name)# default-action accept
```

11. Apply the policy to one or more sites in the overlay network:

```
vSmart(config)# apply-policy site-list list-name data-policy policy-name (all
|from-service | from-tunnel)
```

Centralized Policies Configuration Examples

This topic provides some examples of configuring a centralized data policy to influence traffic flow across the Cisco Catalyst SD-WAN domain and to configure a Cisco Catalyst SD-WAN device to be an internet exit point.

General Centralized Policy Example

This section shows a general example of a centralized data policy to illustrate that you configure centralized data policy on a Cisco Catalyst SD-WAN Controller and that after you commit the configuration, the policy itself is pushed to the required Cisco Catalyst SD-WAN device.

Here we configure a simple data policy on the Cisco Catalyst SD-WAN Controller vm9:

```
vm9# show running-config policy
policy
 data-policy test-data-policy
```

```
  vpn-list test-vpn-list
   sequence 10
    match
     destination-ip 209.165.201.0/27
    !
    action drop
     count test-counter
    !
   !
   default-action drop
  !
 !
 lists
  vpn-list test-vpn-list
   vpn 1
  !
  site-list test-site-list
   site-id 500
  !
 !
!
```

Immediately, after you activate the configuration on the Cisco Catalyst SD-WAN Controller, it pushes the policy configuration to the Cisco vEdge devices in site 500. One of these devices is vm5, where you can see that the policy has been received:

```
vm5# show policy from-vsmart
policy-from-vsmart
 data-policy test-data-policy
  vpn-list test-vpn-list
   sequence 10
    match
     destination-ip 209.165.201.0/27
    !
    action drop
     count test-counter
    !
   !
   default-action drop
  !
 !
 lists
  vpn-list test-vpn-list
   vpn 1
  !
 !
!
```

Control Access

This example shows a data policy that limits the type of packets that a source can send to a specific destination. Here, the host at source address 192.0.2.1 in site 100 and VPN 100 can send only TCP traffic to the destination host at 203.0.113.1. This policy also specifies the next hop for the TCP traffic sent by 192.0.2.1, setting it to be TLOC 209.165.200.225, color gold. All other traffic is accepted as a result of the **default-action** statement.

```
policy
  lists
     site-list north
       site-id 100
     vpn-list vpn-north
       vpn 100
  !
  data-policy tcp-only
```

```
    vpn-list vpn-north
      sequence 10
        match
          source-ip 192.0.2.1/32
          destination-ip 198.51.100.1/32
          protocol tcp
        action accept
          set tloc 203.0.113.1 gold
      !
      default-action accept
  !
!
apply-policy
  site north data-policy tcp-only
```

Restrict Traffic

This examples illustrates how to disallow certain types of data traffic from being sent from between VPNs. This policy drops data traffic on port 25, which carries SMTP mail traffic, that originates in 209.165.201.0/27. However, the policy accepts all other data traffic, including non-SMTP traffic from 209.165.201.0/27.

```
policy
  lists
    data-prefix-list north-ones
      ip-prefix 209.165.201.0/27
      port 25
    vpn-list all-vpns
      vpn 1
      vpn 2
    site-list north
      site-id 100
  !
  data-policy no-mail
   vpn-list all-vpns
     sequence 10
       match
         source-data-prefix-list north-ones
       action drop
     !
     default-action accept
  !
!
apply-policy
  site north data-policy no-mail
```

Allow Traffic to Exit from a Cisco vEdge Device to the Internet

The following example allows data traffic destined for two prefixes on the Internet to exit directly from the local Cisco vEdge device to the internet destination. Configure this policy on the Cisco Catalyst SD-WAN Controller.

```
polcy
 lists
  vpn-list vpn-1
    vpn 1
  !
  site-list nat-sites
    site-id 100,200
  !
data-policy accept-nat
  vpn-list vpn-1
   sequence 100
    match
```

```
     source-ip      10.20.24.0/24
     destination-ip 10.0.12.12/32
    !
    action accept
     count nat
     nat use-vpn 0
    !
   !
   sequence 101
    match
     source-ip      10.20.24.0/24
     destination-ip 10.1.15.13/32
    !
    action accept
     count nat_inet
     nat use-vpn 0
    !
   !
   default-action accept
  !
 !
apply-policy
  site-list nat-sites data-policy accept-nat
```

Using the destination port instead of a destination IP prefix allows greater flexibility for traffic exiting to the internet. Here, traffic can go to all HTTP and HTTPS sites (ports 80 and 443, respectively). Configure this policy on a Cisco Catalyst SD-WAN Controller.

```
data-policy accept-nat
  vpn-list vpn-1
   sequence 100
    match
     source-ip      10.20.24.0/24
     destination-port 80
    !
    action accept
     count nat
     nat use-vpn 0
    !
   !
   sequence 101
    match
     source-ip      10.20.24.0/24
     destination-port 443
    !
    action accept
     count nat_inet
     nat use-vpn 0
    !
   !
   default-action accept
  !
 !
```

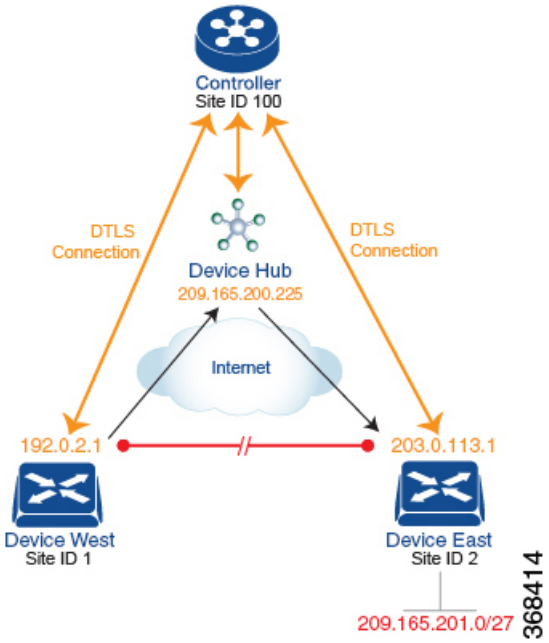
Traffic Engineering

This example of traffic engineering forces all traffic to come to a Cisco SD-WAN device using a device hub instead of directly.

One common way to design a domain in a Cisco SD-WAN overlay network is to route all traffic destined for branches through a hub router, which is typically located in a data center, rather than sending the traffic directly from one Cisco SD-WAN device to another. You can think of this as a hub-and-spoke design, where one device is acting as a hub and the devices are the spokes. With such a design, traffic between local branches

travels over the IPsec connections that are established between the spoke routers and the hub routers when the devices are booted up. Using established connections means that the devices do not need to expend time and CPU cycles to establish IPsec connections with each other. If you were to imagine that this were a large network with many devices, having a full mesh of connections between each pair of routers would require a large amount of CPU from the routers. Another attribute of this design is that, from an administrative point of view, it can be simpler to institute coordinated traffic flow policies on the hub routers, both because there are fewer of them in the overlay network and because they are located in a centralized data center.

One way to direct all the device spoke router traffic to a Cisco hub router is to create a policy that changes the TLOC associated with the routes in the local network. Let's consider the topology in the figure here:



This topology has two devices in different branches:

- The Device West in site ID 1. The TLOC for this device is defined by its IP address (192.0.2.1), a color (gold), and an encapsulation (here, IPsec). We write the full TLOC address as {192.0.2.1, gold, ipsec}. The color is simply a way to identify a flow of traffic and to separate it from other flows.
- The Device East in site ID 2 has a TLOC address of {203.0.113.1, gold, ipsec}.

The devices West and East learn each other's TLOC addresses from the OMP routes distributed to them by the Cisco Catalyst SD-WAN Controller. In this example, the Device East advertises the prefix 209.165.201.0/27 as being reachable at TLOC {203.0.113.1, gold, }. In the absence of any policy, the Device West could route traffic destined for 209.165.201.0/27 to TLOC {203.0.113.1, gold, ipsec}, which means that the Device West would be sending traffic directly to the Device East.

However, our design requires that all traffic from West to East be routed through the hub router, whose TLOC address is {209.165.200.225, gold, ipsec}, before going to the Device East. To effect this traffic flow, you define a policy that changes the route's TLOC. So, for the prefix 209.165.201.0/27, you create a policy that changes the TLOC associated with the prefix 209.165.201.0/27 from {203.0.113.1, gold, ipsec}, which is the TLOC address of the Device East, to {209.165.200.225, gold, ipsec}, which is the TLOC address of the hub router. The result is that the OMP route for the prefix 209.165.201.0/27 that the Cisco Catalyst SD-WAN Controller advertises to the Device West that contains the TLOC address of the hub router instead of the

TLOC address of the Device East. From a traffic flow point of view, the Device West then sends all traffic destined for 209.165.201.0/27 to the hub router.

The device also learns the TLOC addresses of the West and East devices from the OMP routes advertised by the Cisco Catalyst SD-WAN Controller. Because, devices must use these two TLOC addresses, no policy is required to control how the hub directs traffic to the devices.

Here is a policy configuration on the Cisco Catalyst SD-WAN Controller that directs the Device West (and any other devices in the network domain) to send traffic destined to prefix 209.165.201.0/27 to TLOC 209.165.200.225, gold, which is the device:

```
policy
  lists
    prefix-list east-prefixes
      ip-prefix 209.165.201.0/27
    site-list west-sites
      site-id 1
  control-policy change-tloc
    sequence 10
      match route
        prefix-list east-prefixes
        site-id 2
      action accept
        set tloc 209.165.200.225 color gold encap ipsec
apply-policy
  site west-sites control-policy change-tloc out
```

A rough English translation of this policy is:

```
Create a list named “east-prefixes” that contains the IP prefix “209.165.201.0/27”
  Create a list named “west-sites” that contains the site-id “1”
  Define a control policy named “change-tloc”
    Create a policy sequence element that:
      Matches a prefix from list “east-prefixes”, that is, matches “209.165.201.0/27”
      AND matches a route from site-id “2”
    If a match occurs:
      Accept the route
      AND change the route’s TLOC to “209.165.200.225” with a color of "gold" and an
encapsulation of "ipsec"
  Apply the control policy “change-tloc” to OMP routes sent by the vSmart
    controller to “west-sites”, that is, to site ID 1
```

This control policy is configured on the Cisco Catalyst SD-WAN Controller as an outbound policy, as indicated by the **out** option in the apply-policy site command. This option means the Cisco Catalyst SD-WAN Controller applies the TLOC change to the OMP route after it distributes the route from its route table. The OMP route for prefix 209.165.201.0/27 that the Cisco Catalyst SD-WAN Controller distributes to the Device West associates 209.165.201.0/27 with TLOC 209.165.200.225, gold. This is the OMP route that the Device West installs it in its route table. The end results are that when the Device West sends traffic to 209.165.201.0/27, the traffic is directed to the hub; and the Device West does not establish a DTLS tunnel directly with the Device East.

If the West side of the network had many sites instead of just one and each site had its own device, it would be straightforward to apply this same policy to all the sites. To do this, you simply add the site IDs of all the sites in the **site-list west-sites** list. This is the only change you need to make in the policy to have all the West-side sites send traffic bound for the prefix 209.165.201.0/27 through the device. For example:

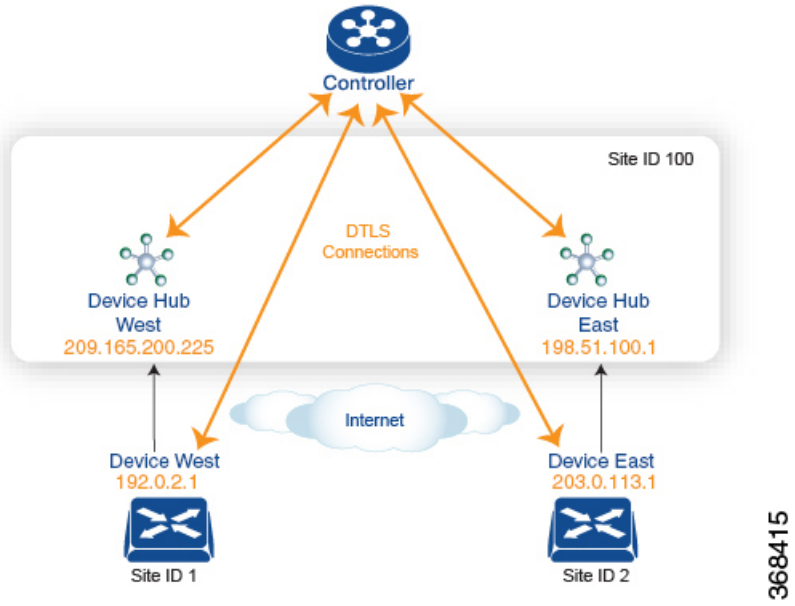
```
policy
  lists
    prefix-list east-prefixes
      ip-prefix 209.165.201.0/27
    site-list west-sites
      site-id 1
```

```
      site-id 11
      site-id 12
      site-id 13
  control-policy change-tloc
    sequence 10
      match route
        prefix-list east-prefixes
        site-id 2
      action accept
        set tloc 209.165.200.225 color gold encap ipsec
apply-policy
  site west-sites control-policy change-tloc out
```

Creating Arbitrary Topologies

To provide redundancy in the hub-and-spoke-style topology discussed in the previous example, you can add a second Cisco hub to create a dual-homed hub site. The following figure shows that site ID 100 now has two Device hubs. We still want all inter-branch traffic to be routed through a device hub. However, because we now have dual-homed hubs, we want to share the data traffic between the two hub routers.

- Device Hub West, with TLOC 209.165.200.225, gold. We want all data traffic from branches on the West side of the overlay network to pass through and be processed by this device.
- Device Hub East, with TLOC 198.51.100.1, gold. Similarly, we all East-side data traffic to pass through the Device Hub East.



Here is a policy configuration on the Cisco Catalyst SD-WAN Controller that would send West-side data traffic through the Cisco hub, and West and East-side traffic through the Device Hub East:

```
policy
  lists
    site-list west-sites
      site-id 1
    site-list east-sites
      site-id 2
    tloc-list west-hub-tlocs
      tloc-id 209.165.200.225 gold
```

```
    tloc-list east-hub-tlocs
      tloc-id 198.51.100.1 gold
  control-policy prefer-west-hub
    sequence 10
      match tloc
        tloc-list west-hub-tlocs
      action accept
        set preference 50
  control-policy prefer-east-hub
    sequence 10
      match tloc
        tloc-list east-hub-tlocs
      action accept
        set preference 50
apply-policy
  site west-sites control-policy prefer-west-hub out
  site east-sites control-policy prefer-east-hub out
```

Here is an explanation of this policy configuration:

Create the site lists that are required for the **apply-policy** configuration command:

- **site-list west-sites** lists all the site IDs for all the devices in the West portion of the overlay network.
- **site-list east-sites** lists the site IDs for the devices in the East portion of the network.

Create the TLOC lists that are required for the match condition in the control policy:

- **west-hub-tlocs** lists the TLOC for the Device West Hub, which we want to service traffic from the West-side device.
- **east-hub-tlocs** lists the TLOC for the Device East Hub, to service traffic from the East devices.

Define two control policies:

- **prefer-west-hub** affects OMP routes destined to TLOC 209.165.200.225, gold, which is the TLOC address of the Device West hub router. This policy modifies the preference value in the OMP route to a value of 50, which is large enough that it is likely that no other OMP routes will have a larger preference. So setting a high preference value directs traffic destined for site 100 to the Device West hub router.
- Similarly, **prefer-east-hub** sets the preference to 50 for OMP routes destined TLOC 198.51.100.1, gold, which is the TLOC address of the Device East hub router, thus directing traffic destined for site 100 site to the Device East hub 198.51.100.1 router.

Apply the control policies:

- The first line in the **apply-policy** configuration has the Cisco Catalyst SD-WAN Controller apply the **prefer-west-hub** control policy to the sites listed in the **west-sites** list, which here is only site ID 1, so that the preference in their OMP routes destined to TLOC 209.165.200.225 is changed to 50 and traffic sent from the Device West to the hub site goes through the Device West hub router.
- The Cisco Catalyst SD-WAN Controller applies the **prefer-east-hub** control policy to the OMP routes that it advertises to the devices in the **east-sites** list, which changes the preference to 50 for OMP routes destined to TLOC 198.51.100.1, so that traffic from the Device East goes to the Device East hub router.

Verify Centralized Control Policies Configuration

Table 15: Feature History

Feature Name	Release Information	Description
Troubleshoot and test policies on a Cisco Catalyst SD-WAN Controller	Cisco SD-WAN Release 20.8.1	This feature provides a method for displaying policy sequences that match a particular input variable and policy name. This is useful for troubleshooting large policies that have numerous sequences.

The following is a sample output from the **test policy match control-policy** command displaying the sequence that matches a particular input variable and policy name. For more information, see the test policy match control-policy command page.

The following sample output displays the sequence of the control_policy1 for vpn 2:

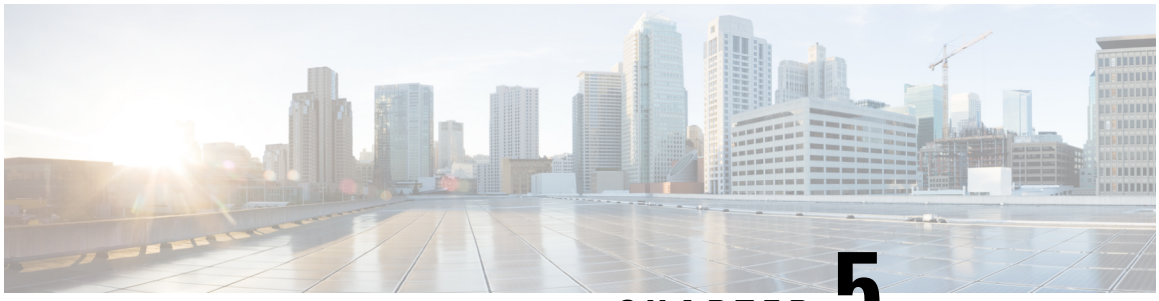
```
Device# test policy match control-policy control_policy1 vpn 2
Found: vpn 2 matches policy control_policy1 sequence 111
    sequence: 111
        match route [VPN-ID  (0x100) ]
            vpn-id: 2
        action: reject
        set: [ (0x0) ]
```

The following sample output displays the sequence of the cp1 policy for the prefix 10.1.1.1/32:

```
Device# test policy match control-policy cp1 prefix 10.1.1.1/32
Found: prefix 10.1.1.1/32 matches policy cp1 sequence 111
    sequence: 111
        match route [PFX-LIST  (0x10) ]
            IPv4 prefix-list: pf1 (0x7f04292bfa00)
        action: reject
        set: [ (0x0) ]
```

The following sample output displays the sequence of the cp1 policy for ipv6-prefix a:a:a:a:a:a:a:a/128:

```
Device# test policy match control-policy cp1 ipv6-prefix a:a:a:a:a:a:a:a/128
Found: ipv6-prefix a:a:a:a:a:a:a:a/128 matches policy cp1 sequence 600
    sequence: 600
        match route [PFX-LIST  (0x10) ]
            IPv6 prefix-list: pfv61 (0x7ff7be6cb080)
        action: reject
        set: [ (0x0) ]
```



CHAPTER 5

Localized Policy

Note To achieve simplification and consistency, the Cisco SD-WAN solution has been rebranded as Cisco Catalyst SD-WAN. In addition, from Cisco IOS XE SD-WAN Release 17.12.1a and Cisco Catalyst SD-WAN Release 20.12.1, the following component changes are applicable: **Cisco vManage** to **Cisco Catalyst SD-WAN Manager**, **Cisco vAnalytics** to **Cisco Catalyst SD-WAN Analytics**, **Cisco vBond** to **Cisco Catalyst SD-WAN Validator**, **Cisco vSmart** to **Cisco Catalyst SD-WAN Controller**, and **Cisco Controllers** to **Cisco Catalyst SD-WAN Control Components**. See the latest Release Notes for a comprehensive list of all the component brand name changes. While we transition to the new names, some inconsistencies might be present in the documentation set because of a phased approach to the user interface updates of the software product.

The topics in this section provide overview information about the different types of localized policies, the components of localized policies, and how to configure localized policies using Cisco SD-WAN Manager or the CLI.

Overview of Localized Policies

Localized policy refers to a policy that is provisioned locally through the CLI on the Cisco vEdge devices, or through a Cisco SD-WAN Manager device template.

Types of Localized Policies

Localized Control Policy

Control policy operates on the control plane traffic in the Cisco SD-WAN overlay network, influencing the determination of routing paths through the overlay network. Localized control policy is policy that is configured on a Cisco vEdge device (hence, it is local) and affects BGP and OSPF routing decisions on the site-local network that the device is part of.

In addition to participating in the overlay network, a Cisco vEdge device participates in the network at its local site, where it appears to the other network devices to be simply a regular router. As such, you can provision routing protocols, such as BGP and OSPF, on the Cisco vEdge device so that it can exchange route information with the local-site routers. To control and modify the routing behavior on the local network, you configure a type of control policy called route policy on the devices. Route policy applies only to routing performed at the local branch, and it affects only the route table entries in the local device's route table.

Localized control policy, which you configure on the devices, lets you affect routing policy on the network at the local site where the device is located. This type of control policy is called route policy. This policy is similar to the routing policies that you configure on a regular driver, allowing you to modify the BGP and OSPF routing behavior on the site-local network. Whereas, centralized control policy affects the routing behavior across the entire overlay network, route policy applies only to routing at the local branch.

Localized Data Policy

Data policy operates on the data plane in the Cisco Catalyst SD-WAN overlay network and affects how data traffic is sent among the Cisco vEdge devices in the network. The Cisco Catalyst SD-WAN architecture defines two types of data policy, centralized data policy, which controls the flow of data traffic based on the IP header fields in the data packets and based on network segmentation, and localized data policy, which controls the flow of data traffic into and out of interfaces and interface queues on a Cisco vEdge device.

Localized data policy, so called because it is provisioned on the local Cisco vEdge device, is applied on a specific router interface and affects how a specific interface handles the data traffic that it is transmitting and receiving. Localized data policy is also referred to as access lists (ACLs). With access lists, you can provision class of service (CoS), classifying data packets and prioritizing the transmission properties for different classes. You can configure policing and provision packet mirroring.

For IPv4, you can configure QoS actions.

You can apply IPv4 access lists in any VPN on the router, and you can create access lists that act on unicast and multicast traffic. You can apply IPv6 access lists only to tunnel interfaces in the transport VPN (VPN 0).

You can apply access lists either in the outbound or inbound direction on the interface. Applying an IPv4 ACL in the outbound direction affects data packets traveling from the local service-side network into the IPsec tunnel toward the remote service-side network. Applying an IPv4 ACL in the inbound direction affects data packets exiting from the IPsec tunnel and being received by the local Cisco vEdge device. For IPv6, an outbound ACL is applied to traffic being transmitted by the router, and an inbound ACL is applied to received traffic.

Explicit and Implicit Access Lists

Access lists that you configure using localized data policy are called *explicit* ACLs. You can apply explicit ACLs in any VPN on the router.

Router tunnel interfaces also have *implicit ACLs*, which are also referred to as *services*. Some of these are present by default on the tunnel interface, and they are in effect unless you disable them. Through configuration,

you can also enable other implicit ACLs. On Cisco vEdge devices, the following services are enabled by default: DHCP (for DHCPv4 and DHCPv6), DNS, and ICMP. You can also enable services for BGP, Netconf, NTP, OSPF, SSHD, and STUN.

Perform QoS Actions

With access lists, you can provision quality of service (QoS) which allows you to classify data traffic by importance, spread it across different interface queues, and control the rate at which different classes of traffic are transmitted. See Forwarding and QoS Overview.

Mirror Data Packets

Once packets are classified, you can configure access lists to send a copy of data packets seen on a Cisco vEdge device to a specified destination on another network device. The Cisco vEdge devices support 1:1 mirroring; that is, a copy of every packet is sent to the alternate destination.

Configure Localized Policy Using Cisco SD-WAN Manager

To configure localized policies, use the Cisco SD-WAN Manager policy configuration wizard. The wizard is a UI policy builder that consists of five windows to configure and modify the following localized policy components:

- Groups of interest, also called lists
- Forwarding classes to use for QoS
- Access control lists (ACLs)
- Route policies
- Policy settings

You configure some or all these components depending on the specific policy you are creating. To skip a component, click **Next** at the bottom of the window. To return to a component, click **Back** at the bottom of the window.

To configure localized policies using Cisco SD-WAN Manager, use the steps identified in the procedures that follow this section.

Start the Policy Configuration Wizard

To start the policy configuration wizard:

1. From the Cisco SD-WAN Manager menu, choose **Configuration** > **Policies**.
2. Select **Localized Policy**.
3. Click **Add Policy**.

The **Create Groups of Interest** page is displayed.

Configure Groups of Interest for Localized Policy

In **Create Groups of Interest**, create lists of groups to use in a localized policy:

In Create Groups of Interest, create new groups of list types as described in the following sections to use in a localized policy:

Configure As Path

1. In the group of interest list, click **AS Path**.
2. Click **New AS Path List**.
3. Enter a name for the list.
4. Enter the AS path, separating AS numbers with a comma.
5. Click **Add**.

   AS Path list specifies one or more BGP AS paths. You can write each AS as a single number or as a regular expression. To specify more than one AS in a single path, include the list separated by commas. To configure multiple AS paths in a single list, include multiple **as-path** options, specifying one AS path in each option.

Configure Community

A community list is used to create groups of communities to use in a match clause of a route map. A community list can be used to control which routes are accepted, preferred, distributed, or advertised. You can also use a community list to set, append, or modify the communities of a route.

1. In the group of interest list, click **Community**.
2. Click **New Community List**.
3. Enter a name for the community list.
4. In the **Add Community** field, enter one or more data prefixes separated by commas in any of the following formats:
   - *aa*:*nn*: Autonomous System (AS) number and network number. Each number is a 2-byte value with a range from 1 to 65535.
   - **internet**: Routes in this community are advertised to the Internet community. This community comprises all BGP-speaking networking devices.
   - **local-as**: Routes in this community are not advertised outside the local AS number.
   - **no-advertise**: Attaches the NO_ADVERTISE community to routes. Routes in this community are not advertised to other BGP peers.
   - **no-export**: Attaches the NO_EXPORT community to routes. Routes in this community are not advertised outside the local AS or outside a BGP confederation boundary. To configure multiple BGP communities in a single list, include multiple **community** options, specifying one community in each option.
5. Click **Add**.

Configure Data Prefix

1. In the **Group of Interest** list, click **Data Prefix**.
2. Click **New Data Prefix List**.
3. Enter a name for the list.
4. Enter one or more IP prefixes.
5. Click **Add**.

   A data prefix list specifies one or more IP prefixes. You can specify both unicast and multicast addresses. To configure multiple prefixes in a single list, include multiple **ip-prefix** options, specifying one prefix in each option.

Configure Extended Community

1. In the group of interest list, click **Extended Community**.
2. Click **New Extended Community List**.
3. Enter a name for the list.
4. Enter the BGP extended community in the following formats:
   - **rt** (*aa*:*nn* | *ip-address*): Route target community, which is one or more routers that can receive a set of routes carried by BGP. Specify this as the AS number and network number, where each number is a 2-byte value with a range from 1 to 65535, or as an IP address.
   - **soo** (*aa*:*nn* | *ip-address*): Route origin community, which is one or more routers that can inject a set of routes into BGP. Specify this as the AS number and network number, where each number is a 2-byte value with a range from 1 to 65535, or as an IP address. To configure multiple extended BGP communities in a single list, include multiple **community** options, specifying one community in each option.
5. Click **Add**.

Configure Class Map

1. In the group of interest list, click **Class Map**.
2. Click **New Class List**.
3. Enter a name for the class.
4. Select a required queue from the **Queue** drop-down list.
5. Click **Save**.

Configure Mirror

1. In the group of interest list, click **Mirror**.
2. Click **New Mirror List**. The Mirror List popup displays.
3. Enter a name for the list.

4. In the **Remote Destination IP** field, enter the IP address of the destination for which to mirror the packets.
5. In the **Source IP** field, enter the IP address of the source of the packets to mirror.
6. Click **Add**.

   To configure mirroring parameters, define the remote destination to which to mirror the packets, and define the source of the packets. Mirroring applies to unicast traffic only. It does not apply to multicast traffic.

Configure Policer

1. In the group of interest list, click **Policer**.
2. Click **New Policer List**.
3. Enter a name for the list.
4. In the **Burst (bps)** field, enter maximum traffic burst size. It can be a value from 15000 to 10000000 bytes.
5. In the **Exceed** field, select the action to take when the burst size or traffic rate is exceeded. Select **Drop** (the default) to set the packet loss priority (PLP) to low. Select **Remark** to set the PLP to high.
6. In the **Rate (bps)** field, enter the maximum traffic rate. It can be value from 8 through 2^{64} bps (8 through 100000000000).
7. Click **Add**.

Configure Prefix

1. In the group of interest list, click **Prefix**.
2. Click **New Prefix List**.
3. Enter a name for the list.
4. In the **Internet Protocol** field, click either **IPv4** or **IPv6**.
5. Under **Add Prefix**, enter the prefix for the list. (An example is displayed.) Optionally, click the green **Import** link on the right-hand side to import a prefix list.
6. Click **Add**.

Click **Next** to move to **Configure Forwarding Classes/QoS** in the wizard.

Configure Forwarding Classes/QoS

When you first open the **Forwarding Classes/QoS** page, **QoS Map** is selected by default:

QoS Map

To create a new QoS mapping:

1. In **QoS**, click the **Add QoS Map** drop-down.
2. Select **Create New**.

3. Enter a name and description for the QoS mapping.
4. Click **Add Queue**. The **Add Queue** popup appears.
5. Select the queue number from the **Queue** drop-down.
6. Select the maximum bandwidth and buffer percentages, and the scheduling and drop types.
7. Enter the **Forwarding Class**.
8. Click **Save Queue**.

To import an existing QoS mapping:

1. In **QoS**, click the **Add QoS Map** drop-down.
2. Select **Import Existing**. The **Import Existing Application QoS Map Policy** popup displays.
3. Select a **QoS Map** policy.
4. Click **Import**.

To view or copy a QoS mapping or to remove the mapping from the localized policy, click **...** and select the desired action.

For hardware, each interface has eight queues, numbered from 0 through 7. Queue 0 is reserved for low-latency queuing (LLQ), so any class that is mapped to queue 0 must be configured to use LLQ. The default scheduling method for all is weighted round-robin (WRR).

For Cisco vEdge devices, each interface has eight queues, numbered from 0 through 7. Queue 0 is reserved for control traffic, and queues 1, 2, 3, 4, 5, 6 and 7 are available for data traffic. The scheduling method for all eight queues is WRR. LLQ is not supported.

To configure QoS parameters on a Cisco vEdge device, you must enable QoS scheduling and shaping. To enable QoS parameters for traffic that the Cisco vEdge device receives from transport-side interfaces:

To enable QoS parameters for traffic that the Cisco vEdge device receives from service-side interfaces:

Policy Rewrite

To configure policy rewrite rules for the QoS mapping:

1. In **Policy Rewrite**, click the **Add Rewrite Policy** drop-down.
2. Select **Create New**.
3. Enter a name and description for the rewrite rule.
4. Click **Add Rewrite Rule**. The **Add Rule** popup appears.
5. Select a class from the **Class** drop-down.
6. Select the priority (**Low** or **High**) from the Priority drop-down.
7. Enter the DSCP value (0 through 63) in the **DSCP** field.
8. Enter the class of service (CoS) value (0 through 7) in the **Layer 2 Class of Service** field.
9. Click **Save Rule**.

To import an existing rewrite rule:

1. In **QoS**, click the **Add Rewrite Policy** drop-down..
2. Select **Import Existing**. The **Import Existing Policy Rewrite** popup appears.
3. Select a rewrite rule policy.
4. Click **Import**.

Click **Next** to move to **Configure Access Lists** page.

Configure ACLs

1. In the **Configure Access Control Lists** page, configure ACLs.
2. To create a new ACL, click the **Add Access Control List Policy** drop-down. Select one from the following options:
   - **Add IPv4 ACL Policy**: Configure IPv4 ACL policy.
   - **Add IPv6 ACL Policy**: Configure IPv6 ACL policy.
   - **Import Existing**: Import existing ACL policy.
3. If you click **Add IPv4 ACL Policy**, the **Add IPv4 ACL Policy** page appears.

   or

   If you click **Add IPv6 ACL Policy**, the **Add IPv6 ACL Policy** page appears.
4. Enter a name and description for the ACL in the **ACL Policy** page.
5. In the left pane, click **Add ACL Sequence**. An **Access Control List** box is displayed in the left pane.
6. Double-click the **Access Control List** box, and type a name for the ACL.
7. In the right pane, click **Add Sequence Rule** to create a single sequence in the ACL. **Match** is selected by default.
8. Click a match condition.
9. On the left, enter the values for the match condition.
   a. On the right enter the action or actions to take if the policy matches.
10. Repeat Steps 6 through 8 to add match–action pairs to the ACL.
11. To rearrange match–action pairs in the ACL, in the right pane drag them to the desired position.
12. To remove a match–action pair from the ACL, click the **X** in the upper right of the condition.
13. Click **Save Match and Actions** to save a sequence rule.
14. To rearrange sequence rules in an ACL, in the left pane drag the rules to the desired position.
15. To copy, delete, or rename an ACL sequence rule, in the left pane, click **...** next to the rule's name and select the desired option.

Default Action

If a packet being evaluated does not match any of the match conditions in a access list, a default action is applied to this packet. By default, the packet is dropped. To change the default action:

1. Click **Default Action** in the left pane.
2. Click the **Pencil** icon.
3. Change the default action to **Accept**.
4. Click **Save Match and Actions**.
5. Click **Save Access Control List Policy**.

To configure **Device Access Policy**, see Device Access Policy.

Click **Next** to move to Configure Route Policy page.

Explicit and Implicit Access Lists

Access lists that you configure through localized data policy using the **policy access-list** command are called *explicit* ACLs. You can apply explicit ACLs to any interface in any VPN on the device.

The device's tunnel interfaces in VPN 0 also have *implicit ACLs*, which are also referred to as *services*. Some services are enabled by default on the tunnel interface, and are in effect unless you disable them. Through configuration, you can also enable other services. You configure and modify implicit ACLs with the **allow-service** command:

```
Device(config)# vpn 0
Device(config-vpn)# interface interface-name
Device(config-interface)# tunnel-interface
Device(config-tunnel-interface)# allow-service service-name
Device(config-tunnel-interface)# no allow-service service-name
```

On Cisco vEdge devices, the following services are enabled by default: DHCP (for DHCPv4 and DHCPv6), DNS, and ICMP. These three services allow the tunnel interface to accept DHCP, DNS, and ICMP packets. You can also enable services for BGP, Netconf, NTP, OSPF, SSHD, and STUN.

Note If a connection is initiated from a device, and if NAT is enabled on the device (for example, Direct Internet Access (DIA) is configured), return traffic is allowed by the NAT entry even if the implicit ACL has been configured as **no allow-service**. You can still block this traffic with an explicit ACL.

When data traffic matches both an explicit ACL and an implicit ACL, how the packets are handled depends on the ACL configuration. Specifically, it depends on:

- Whether the implicit ACL is configured as allow (**allow-service** *service-name*) or deny (**no allow-service** *service-name*). Allowing a service in an implicit ACL is the same as specifying the **accept** action in an explicit ACL, and a service that is not allowed in an implicit ACL is the same as specifying the **drop** action in an explicit ACL
- Whether, in an explicit ACL, the **accept** or **deny** action is configured in a policy sequence or in the default action.

The following table explains how traffic matching both an implicit and an explicit ACL is handled:

Table 16:

Implicit ACL	Explicit ACL: Sequence	Explicit ACL: Default	Result
Allow (accept)	Deny (drop)	—	Deny (drop)
Allow (accept)	—	Deny (drop)	Allow (accept)
Deny (drop)	Allow (accept)	—	Allow (accept)
Deny (drop)	—	Allow (accept)	Deny (drop)

Configure Route Policies

In **Configure Route Policies**, configure the routing policies:

1. In **Add Route Policy**, select **Create New**.
2. Enter a name and description for the route policy.
3. In the left pane, click **Add Sequence Type**. A **Route** box is displayed in the left pane.
4. Double-click the **Route** box, and type a name for the route policy.
5. In the right pane, click **Add Sequence Rule** to create a single sequence in the policy. **Match** is selected by default.
6. Select a desired protocol from the **Protocol** drop-down list. The options are: IPv4, IPv6, or both.
7. Click a match condition.
8. On the left, enter the values for the match condition.
9. On the right enter the action or actions to take if the policy matches.
10. Repeat Steps 6 through 8 to add match–action pairs to the route policy.
11. To rearrange match–action pairs in the route policy, in the right pane drag them to the desired position.
12. To remove a match–action pair from the route policy, click the X in the upper right of the condition.
13. Click **Save Match and Actions** to save a sequence rule.
14. To rearrange sequence rules in an route policy, in the left pane drag the rules to the desired position.
15. To copy, delete, or rename the route policy sequence rule, in the left pane, click **...** next to the rule's name and select the desired option.
16. If no packets match any of the route policy sequence rules, the default action is to drop the packets. To change the default action:
    a. Click **Default Action** in the left pane.
    b. Click the Pencil icon.

c. Change the default action to **Accept**.

d. Click **Save Match and Actions**.

17. Click **Save Route Policy**.
18. Click **Next** to move to **Policy Overview** page.

Match Parameters

Access List Parameters

Access lists can match IP prefixes and fields in the IP headers.

In the CLI, you configure the match parameters with the **policy access-list sequence match** command.

Each sequence in an access-list must contain one match condition.

Match class in ACL is not supported. You can use rewrite policy to configure DSCP values.

For access lists, you can match these parameters:

Match Condition	Description
Class	Name of a class defined with a **policy class-map** command.
Destination Data Prefix	Name of a data-prefix-list list.
Destination Port	Specifies a single port number, a list of port numbers (with numbers separated by a space), or a range of port numbers (with the two numbers separated with a hyphen [-]). The range is 0 through 65535.
DSCP	Specifies the DSCP value. The range is 0 through 63.
Protocol	Specifies the internet protocol number. The range is 0 through 255.
ICMP Message	When you select a Protocol value as 1 the **ICMP Message** field displays where you can select an ICMP message to apply to the data policy. When you select a Next Header value as 58 the **ICMP Message** field displays where you can select an ICMP message to apply to the data policy. **Note** This field is available from , Cisco SD-WAN Release 20.4.1 Cisco vManage Release 20.4.1.
Packet Length	Specifies the length of the packet. The range can be from 0 through 65535. Specify a single length, a list of lengths (with numbers separated by a space), or a range of lengths (with the two numbers separated with a hyphen [-]).
Source Data Prefix	Specifies the name of a **data-prefix-list** list.
PLP	Specifies the Packet Loss Priority (PLP) (**high** | **low**). By default, packets have a PLP value of **low**. To set the PLP value to **high**, apply a policer that includes the **exceed remark** option.

Match Condition	Description
Source Port	Specifies a single port number, a list of port numbers (with numbers separated by a space), or a range of port numbers (with the two numbers separated with a hyphen [-]). The range is 0 through 65535.
TCP	**syn**

Route Policy Parameters

For route policies, you can match these parameters:

Match Condition	Description
Address	Specifies the name of a **Prefix-List** list.
AS Path List	Specifies one or more BGP AS path lists. You can write each AS as a single number or as a regular expression. To specify more than one AS number in a single path, include the list in quotation marks (" "). To configure multiple AS numbers in a single list, include multiple **AS Path** options, specifying one AS path in each option.
Community List	List of one of more BGP communities. In **Community List**, you can specify: • *aa*:*nn*: AS number and network number. Each number is a 2-byte value with a range from 1 to 65535. • **internet**: Routes in this community are advertised to the Internet community. This community comprises all BGP-speaking networking devices. • **local-as**: Routes in this community are not advertised outside the local AS. • **no-advertise**: Attach the NO_ADVERTISE community to routes. Routes in this community are not advertised to other BGP peers. • **no-export**: Attach the NO_EXPORT community to routes. Routes in this community are not advertised outside the local AS or outside a BGP confederation boundary. To configure multiple BGP communities in a single list, include multiple **community** options, specifying one community in each option.
Extended Community List	Specifies the list of one or more BGP extended communities. In **community**, you can specify: • **rt** (*aa*:*nn* | *ip-address*): Route target community, which is one or more routers that can receive a set of routes carried by BGP. Specify this as the AS number and network number, where each number is a 2-byte value with a range from 1 to 65535, or as an IP address. • **soo** (*aa*:*nn* | *ip-address*): Route origin community, which is one or more routers that can inject a set of routes into BGP. Specify this as the AS number and network number, where each number is a 2-byte value with a range from 1 to 65535, or as an IP address. To configure multiple extended BGP communities in a single list, include multiple **community** options, specifying one community in each option.
BGP Local Preference	Specifies the BGP local preference number. The range is 0 through 4294967295.
Metric	Specifies the route metric value. The range is 0 through 4294967295.

Match Condition	Description
Next Hop	Specifies the name of an IP prefix list.
OMP Tag	Specifies the OMP tag number. The range is 0 through 4294967295.
Origin	Specifies the BGP origin code. The optionss are: EGP (default), IGP, Incomplete.
OSPF Tag	Specifies the OSPF tag number. The range is 0 through 4294967295.
Peer	Specifies the peer IP address.

Action Parameters

Access List Parameters

When a packet matches the conditions in the match portion of an access list, the packet can be accepted or dropped, and it can be counted. Then, you can classify, mirror, or police accepted packets.

In the CLI, you configure the action parameters with the **policy access-list sequence action** command.

Each sequence in an access list can contain one action condition.

In the action, you first specify whether to accept or drop a matching data packet, and whether to count it:

Action Condition	Description
Accept	Accepts the packet. An accepted packet is eligible to be modified by the additional parameters configured in the **action** portion of the access list.
Counter	Name of a counter. To display counter information, use the **show policy access-lists counters** command on the Cisco vEdge device.
Drop	Discards the packet. This is the default action.

For a packet that is accepted, the following actions can be configured:

Description	Value or Range
Class	Specifies the name of a QoS class. It can also be defined with a **policy class-map** command.
Mirror List	Specifies the name of mirror . It is defined with a **policy mirror** command.
Policer	Specifies the name of a policer defined with a **policy policer** command.
DSCP	Specifies the packet's DSCP value. The range is 0 through 63.
Next Hop	Specifies the IPv4 address. It sets the next hop IP address to which the packet should be forwarded. **Note** Starting from Cisco vManage Release 20.5.1 and Cisco SD-WAN Release 20.5.1, **Use Default Route when Next Hop is not available** field is available next to Next Hop action parameter.

Route Policy Parameters

Each sequence in a localized control policy can contain one action condition.

When a route matches the conditions in the match portion of a route policy, the route can be accepted or rejected:

For a packet that is accepted, the following actions can be configured:

Description	Value or Range
Aggregator	Set sthe AS number in which a BGP route aggregator is located and the IP address of the route aggregator. The range is 1 through 65535.
As Path	Sets an AS number or a series of AS numbers to exclude from the AS path or to prepend to the AS path. The range is 1 through 65535.
Atomic Aggregate	Sets the BGP atomic aggregate attribute.
Community	Sets the BGP community value.
Local Preference	Sets the BGP local preference. The range is 0 through 4294967295.
Metric	Sets the metric value. The range is 0 through 4294967295.
Metric Type	Sets the metric type. The options are type1 or type2.
Next Hop	Sets the IPv4 address. It sets the next hop IP address to which the packet should be forwarded. **Note** Starting from Cisco vManage Release 20.5.1 and Cisco SD-WAN Release 20.5.1, **Use Default Route when Next Hop is not available** field is available next to Next Hop action parameter.
OMP Tag	Sets the OMP tag for OSPF to use. The range is 0 through 4294967295.
Origin	Sets the BGP origin code. The options are: EGP (default), IGP, Incomplete.
Originator	Sets the IP address from which the route was learned.
OSPF Tag	Sets the OSPF tag value. The range is 0 through 4294967295.
Weight	Sets the BGP weight. The range is 0 through 4294967295.

Configure Policy Settings

In **Policy Overview**, configure the policy settings:

1. In the **Enter name and description for your localized master policy** pane, enter name and description for the policy.
2. In the **Policy Settings** pane, select the policy application checkboxes that you want to confirgure. The options are:
   - **Netflow**: Perform traffic flow monitoring on IPv4 traffic.
   - **Netflow IPv6**: Perform traffic flow monitoring on IPv6 traffic.

- **Application**: Track and monitor IPv4 applications.
- **Application IPv6**: Track and monitor IPv6 applications.
- **Cloud QoS**: Enable QoS scheduling.
- **Cloud QoS Service Side**: Enable QoS scheduling on the service side.
- **Implicit ACL Logging**: Log the headers of all the packets that are dropped because they do not match a service perform traffic flow monitoring.

3. To configure how often packets flows are logged, click **Log Frequency**.

   Packet flows are those that match an access list (ACL), a cflowd flow, or an application-aware routing flow.

4. Click **Preview** to view the full policy in CLI format.
5. Click **Save Policy**.

Apply Localized Policy in a Device Template

1. From the Cisco SD-WAN Manager menu, choose **Configuration** > **Templates**.
2. If you are creating a new device template:
   a. Click **Device Templates**.

Note In Cisco vManage Release 20.7.1 and earlier releases, **Device Template** is titled as **Device**.

   b. From the **Create Template** drop-down, select **From Feature Template**.
   c. From the **Device Model** drop-down, select one of the Cisco vEdge devices.
   d. In the **Template Name** field, enter a name for the device template. This field is mandatory and can contain only uppercase and lowercase letters, the digits 0 through 9, hyphens (–), and underscores (_). It cannot contain spaces or any other characters.
   e. In the **Description** field, enter a description for the device template. This field is mandatory, and it can contain any characters and spaces.
   f. Continue with Step 4.

3. If you are editing an existing device template:
   a. Click **Device Templates**, and for the desired template, click **...** and select **Edit**.

Note In Cisco vManage Release 20.7.1 and earlier releases, **Device Template** is titled as **Device**.

   b. Click **Additional Templates**. The screen scrolls to the **Additional Templates** section.
   c. From the **Policy** drop-down, select the name of a policy that you have configured.

4. Click **Additional Templates** located directly beneath the **Description** field. The screen scrolls to the **Additional Templates** section.
5. From the **Policy** drop-down, select the name of the policy you configured in the above procedure.
6. Click **Create** (for a new template) or **Update** (for an existing template).

Activate a Localized Policy

1. Click **Localized Policy**, and select a policy.
2. For the desired policy, click **...** and select **Activate**.
3. In the **Activate Policy** popup, click **Activate** to push the policy to all reachable Cisco SD-WAN Controllers in the network.
4. Click **OK** to confirm activation of the policy on all Cisco SD-WAN Controllers.
5. To deactivate the localized policy, select =, and then select a policy.
6. For the desired policy, click **...** and select **Deactivate**.
7. In the **Deactivate Policy** popup, click **Deactivate** to confirm that you want to remove the policy from all reachable Cisco SD-WAN Controllers.

View Localized Policies

To view localized policies:

1. Click **Localized Policy**, and select a policy.
2. For a policy created using the UI policy builder or using the CLI, click **...** and select **View**. The policy created using the UI policy builder is displayed in graphical format while the policy created using the CLI method is displayed in text format.
3. For a policy created using the Cisco SD-WAN Manager policy configuration wizard, click **...** and select **Preview**. This policy is displayed in text format.

Copy, Edit, and Delete Policies

To copy a policy:

1. Click **Localized Policy**, and select a policy.
2. For the desired policy, click **...** and select **Copy**.
3. In the Policy Copy popup window, enter the policy name and a description of the policy.

Note If you are upgrading to 18.4.4 version, data policy names need to be under 26 characters.

Note Starting with the Cisco SD-WAN release 19.3, 127 characters are supported for policy names for the following policy types:

- Central route policy
- Local route policy
- Local Access Control lOst (ACL)
- Local IPv6 ACL
- Central data policy
- Central app route policy
- QoS map
- Rewrite rule

All other policy names support 32 characters.

4. Click **Copy**.

To edit policies created using the Cisco SD-WAN Manager policy configuration wizard:

1. For the desired policy, click **...** and select **Edit**.
2. Edit the policy as needed.
3. Click **Save Policy Changes**.

To edit polices created using the CLI method:

1. From the **Custom Options** drop-down, under Localized Policy, select **CLI Policy**.
2. For the desired policy, click **...** and select **Edit**.
3. Edit the policy as needed.
4. Click **Update**.

To delete policies:

1. Click **Localized Policy**, and select a policy.
2. For the desired policy, click **...** and select **Delete**.
3. Click **OK** to confirm deletion of the policy.

Configure Localized Policy for IPv4 Using the CLI

Following are the high-level steps for configuring an access list using the CLI for Cisco vEdge devices:

1. Create lists of IP prefixes as needed:

```
vEdge(config)# policy
vEdge(config-policy)# lists data-prefix-list list-name
vEdge(config-data-prefix-list)# ip-prefix prefix/length
```

2. If you configure a logging action, configure how often to log packets to the syslog files:

```
vEdge(config)# policy log-frequency number
```

3. For QoS, map each forwarding class to an output queue, configure a QoS scheduler for each forwarding class, and group the QoS schedulers into a QoS map:

```
vEdge(config)# policy class-map
vEdge(config-class-map)# class class-name queue number
vEdge(config)# policy qos-scheduler scheduler-name
vEdge(config-qos-scheduler)# class class-name
vEdge(config-qos-scheduler)# bandwidth-percent percentage
vEdge(config-qos-scheduler)# buffer-percent percentage
vEdge(config-qos-scheduler)# drops drop-type
vEdge(config-qos-scheduler)# scheduling type
```

```
vEdge(config)# policy qos-map map-name qos-scheduler scheduler-name
```

4. For QoS, define rewrite rules to overwrite the DSCP field of a packet's outer IP header, if desired:

```
vEdge(config)# policy rewrite-rule rule-name
vEdge(config-rewrite-rule)# class class-name loss-priority
dscp dscp-value layer-2-cos number
```

class-name is one of the classes defined under a **qos-scheduler** command.

5. Define mirroring parameters (for unicast traffic only):

```
vEdge(config)# policy mirror mirror-name
vEdge(config-mirror)# remote-dest ip-address source ip-address
```

6. Define policing parameters:

```
vEdge(config)# policy policer policer-name
vEdgeconfig-policer)# rate bandwidth
vEdge(config-policer)# burst bytes
vEdge(config-policer)# exceed action
```

7. Create an access list instance:

```
vEdge(config)# policy access-list list-name
```

8. Create a series of match–action pair sequences:

```
vEdge(config-access-list)# sequence number
vEdge(config-sequence)#
```

The match–action pairs are evaluated in order, by sequence number, starting with the lowest numbered pair and ending when the route matches the conditions in one of the pairs. Or if no match occurs, the default action is taken (either rejecting the route or accepting it as is).

9. Define match parameters for packets:

```
vEdge(config-sequence-number)
# match match-parameter
```

10. Define actions to take when a match occurs:

```
vEdge(config-sequence)# action drop
vEdge(config-sequence)# action count counter-name
vEdge(config-sequence)# action log
vEdge(config-sequence)# action accept class class-name
```

```
vEdge(config-sequence)# action accept mirror mirror-name
vEdge(config-sequence)# action accept policer policer-name
vEdge(config-sequence)# action accept set dscp value
vEdge(config-sequence)# action accept set next-hop ipv4-address
```

11. Create additional numbered sequences of match–action pairs within the access list, as needed.
12. If a packet does not match any of the conditions in one of the sequences, it is rejected by default. If you want nonmatching packets to be accepted, configure the default action for the access list:

    ```
    vEdge(config-policy-name)
    # default-action accept
    ```

13. Apply the access list to an interface:

    ```
    vEdge(config)# vpn vpn-id interface interface-name
    vEdge(config-interface)# access-list list-name (in | out)
    ```

    Applying the access list in the inbound direction (**in**) affects packets being received on the interface. Applying it in the outbound direction (**out**) affects packets being transmitted on the interface. For QoS, apply a DSCP rewrite rule to the same egress interface:

    ```
    vEdge(config)# vpn vpn-id interface interface-name rewrite-rule rule-name
    ```

14. You can apply a policer directly to an interface, which has the effect of policing all packets transiting the interface, rather than policing only the selected packets that match the access list. You can apply the policer to either inbound or outbound packets:

    ```
    vEdge(config)# vpn vpn-id  interface interface-name
    vEdge(config-interface)# policer policer-name (in | out)
    ```

Configure Localized Policy for IPv6 Using the CLI

Following are the high-level steps for configuring an access list using the CLI:

1. Define mirroring parameters (for unicast traffic only):

   ```
   vEdge(config)# policy mirror mirror-name
   vEdge(config-mirror)# remote-dest ip-address source ip-address
   ```

2. Define policing parameters:

   ```
   vEdge(config)# policy policer policer-name
   vEdge(config-policer)# rate bandwidth
   vEdge(config-policer)# burst bytes
   vEdge(config-policer)# exceed action
   ```

3. Create an access list instance:

   ```
   vEdge(config)# policy ipv6 access-list list-name
   ```

4. Create a series of match–action pair sequences:

   ```
   vEdge(config-ipv6-access-list)# sequence number
   vEdge(config-sequence)#
   ```

   The match–action pairs are evaluated in order, by sequence number, starting with the lowest numbered pair and ending when the route matches the conditions in one of the pairs. Or if no match occurs, the default action is taken (either rejecting the route or accepting it as is).

5. Define match parameters for packets:

```
vEdge(config-sequence-number)# match match-parameter
```

6. Define actions to take when a match occurs:

```
vEdge(config-sequence)# action drop
vEdge(config-sequence)# action count counter-name
vEdge(config-sequence)# action log
vEdge(config-sequence)# action accept class class-name
vEdge(config-sequence)# action accept mirror mirror-name
vEdge(config-sequence)# action accept policer policer-name
```

7. Create additional numbered sequences of match–action pairs within the access list, as needed.

8. If a packet does not match any of the conditions in one of the sequences, it is rejected by default. If you want nonmatching packets to be accepted, configure the default action for the access list:

```
vEdge(config-policy-name)# default-action accept
```

9. Apply the access list to an interface:

```
vEdge(config)# vpn vpn-id interface interface-name
vEdge(config-interface)# ipv6 access-list list-name (in | out)
```

   Applying the access list in the inbound direction (**in**) affects packets being received on the interface. Applying it in the outbound direction (**out**) affects packets being transmitted on the interface.

Localized Data Policy Configuration Examples

This topic provides some straightforward examples of configuring localized data policy to help you get an idea of how to use policy to influence traffic flow across the Cisco Catalyst SD-WAN domain. Localized data policy, also known as access lists, is configured directly on the local Cisco vEdge devices.

QoS

You can configure quality of service (QoS) to classify data packets and control how traffic flows out of and in to the interfaces on a Cisco vEdge device and on the interface queues. For examples of how to configure a QoS policy, see Forwarding and QoS Configuration Examples.

Mirroring Example

This example illustrates how to configure a mirror instance to automatically send a copy of certain types of data packet to a specified destination for analysis. After you configure the mirror instance, include it in an access list. Here, "mirror-m1" is configured with the host at source address 10.20.23.16 and destination host at 10.2.2.11. The mirror instance is then included in the access list "acl2," which is configured so that data packets originating from the host at source address 10.20.24.17 and going to the destination host at 10.20.25.18 are mirrored to the destination host at 10.2.2.11 with the source address of the originating host as 10.20.23.16.

```
policy
 mirror m1
  remote-dest 10.2.2.11 source 10.20.23.16
 !
!

vm5# show running-config policy access-list acl2
policy
 access-list acl2
  sequence 1
   match
```

```
    source-ip      10.20.24.17/32
    destination-ip 10.20.25.18/32
   !
   action accept
    mirror m1
   !
  !
  default-action drop
 !
!
```

ICMP Message Example

This example displays the configuration for localized data policy for ICMP messages.

```
policy
access-list acl_1
 sequence 100
  match
   protocol 1
   icmp-msg administratively-prohibited
  !
  action accept
   count administratively-prohibited
  !
  !
```

QoS For Router Generated Cisco SD-WAN Manager Traffic

Table 17: Feature History

Feature Name	Release Information	Description
QoS for Router Generated Cisco SD-WAN Manager Traffic	Cisco IOS XE Catalyst SD-WAN Release 17.11.1a	This feature helps you to prioritize or queue router-generated Cisco SD-WAN Manager traffic based on your specific requirements. Use QoS policies and class maps to route Cisco SD-WAN Manager traffic through a queue of your choice.

Information About QoS For Router-Generated Cisco SD-WAN Manager Traffic

Quality of Service (QoS) is a technique used to manage and prioritize network traffic to ensure that certain types of traffic are given priority over others. QoS is particularly important for router-generated Cisco SD-WAN Manager traffic, which is used for managing and monitoring network devices. For more information see, Forwarding and QoS.

You can prioritize or queue router-generated traffic based on your specific requirements. The prioritization can be achieved through the use of QoS policies and class maps.

Use the following steps to put router-generated traffic into the queue of your choice:

1. Define a class map using a CLI template: Identifies the type of traffic you want to prioritize. In this case, you create a class map to identify the router-generated traffic to queue.
2. Define a policy map using a CLI template: Defines the actions that you want to take on the traffic identified in the class map. Create a policy map that assigns a priority or places the router-generated traffic into a specific queue.

Benefits of Qos For Router Generated Cisco SD-WAN Manager Traffic

- Improved network performance: By prioritizing critical router-generated traffic over less important traffic, ensure that your network management functions operate smoothly and monitor and control network devices effectively.
- Better user experience: Queing router-generated traffic helps preventing congestion on the network and ensure that user-generated traffic does not negatively impact network management functions. The queueing can result in a better user experience.
- Increased network availability: Reduces the risk of network downtime caused by network management issues. This improves network availability and reduce the impact of any network issues on your business operations.
- Simplified network management: Simplifies network management and reduces the need for manual intervention. The simplification can save time and reduce the risk of human error.
- Efficient use of network resources: QoS policies and class maps allow you to allocate network resources efficiently, ensuring that critical router-generated traffic flow efficiently, minimizing the impact on other network traffic.

Restrictions For QoS For Router Generated Cisco SD-WAN Manager Traffic

- The QoS for router generated Cisco SD-WAN Manager traffic feature is supported only on Cisco IOS XE Catalyst SD-WAN devices.
- Configuring QoS for router generated Cisco SD-WAN Manager traffic is possible only using a CLI template.
- With this feature, you can prioritize, using a queue, only for the traffic that devices generate forCisco SD-WAN Manager. Other data and management plane traffic continue to take Queue 0 by default.

Configure QoS for Router Generated Cisco SD-WAN Manager Traffic Using a CLI Template

For more information about using CLI templates, see CLI Add-On Feature Templates and CLI Templates.

Note By default, CLI templates execute commands in global config mode.

Define a Class Map and Map to a Queue Number

1. Using a localized policy, define a class-map and map the class-map to a queue number :

   policy class-map class *Queue_1* **queue** *2*
2. Commit the changes.

Here's the complete configuration example for defining a class map and mapping it to a queue number:

```
config-t
policy class-map class Queue_1 queue 2
!
```

Enable Qos For Router Generated Cisco SD-WAN Manager Traffic

This section provides example CLI configurations to enable QoS for router generated Cisco SD-WAN Manager traffic:

1. Enter config-policy mode:

   policy
2. Use a forwarding class and use the class map that you mapped to a queue that you want to prioritize:

   vmanage-forwarding-class *queue_name*
3. Commit the changes.

   QoS for router generated Cisco SD-WAN Manager traffic is enabled.

Here's the complete configuration example for enabling Qos for router generated Cisco SD-WAN Manager traffic:

```
config-t
policy
vmanage-forwarding-class Queue_1
!
```

Verify QoS for Router Generated Cisco SD-WAN Manager Traffic Using CLI

The following is sample output from the **show policy-map interface** command using the **GigabitEthernet 1** keyword:

```
Device# show policy-map interface GigabitEthernet 1

  Service-policy output: shape_GigabitEthernet1

    Class-map: class-default (match-any)
      8619 packets, 5056404 bytes
      5 minute offered rate 113000 bps, drop rate 0000 bps
      Match: any
      Queueing
```

```
queue limit 64 packets
(queue depth/total drops/no-buffer drops) 0/0/0
(pkts output/bytes output) 8619/5056404
shape (average) cir 4200000, bc 16800, be 16800
target shape rate 4200000

Service-policy : qosmap

  queue stats for all priority classes:
    Queueing
    priority level 1
    queue limit 512 packets
    (queue depth/total drops/no-buffer drops) 0/0/0
    (pkts output/bytes output) 565/95064

  Class-map: Queue0 (match-any)
    565 packets, 95064 bytes
    5 minute offered rate 4000 bps, drop rate 0000 bps
    Match: qos-group 0
    police:
        rate 30 %
        rate 1260000 bps, burst 39375 bytes
      conformed 565 packets, 95064 bytes; actions:
        transmit
      exceeded 0 packets, 0 bytes; actions:
        drop
      conformed 4000 bps, exceeded 0000 bps
    Priority: Strict, b/w exceed drops: 0

    Priority Level: 1

  Class-map: Queue_1 (match-any)
    8050 packets, 4961100 bytes                                  ------------------->
    5 minute offered rate 111000 bps, drop rate 0000 bps
    Match: qos-group 1
    Queueing
    queue limit 64 packets
    (queue depth/total drops/no-buffer drops) 0/0/0
    (pkts output/bytes output) 8050/4961100
    bandwidth remaining ratio 10

  Class-map: Queue_2 (match-any)
    4 packets, 240 bytes
    5 minute offered rate 0000 bps, drop rate 0000 bps
    Match: qos-group 2
    Queueing
    queue limit 64 packets
    (queue depth/total drops/no-buffer drops) 0/0/0
    (pkts output/bytes output) 4/240
    bandwidth remaining ratio 10
```

In this example, **Class-map** for the respective queues displays the number, size, and the rate of packet transfer from the router to the destination. You can see a change in the Queue_1 and keep track of the packet transfer.

Troubleshooting QoS For Router Generated Cisco SD-WAN Manager Traffic

Problem

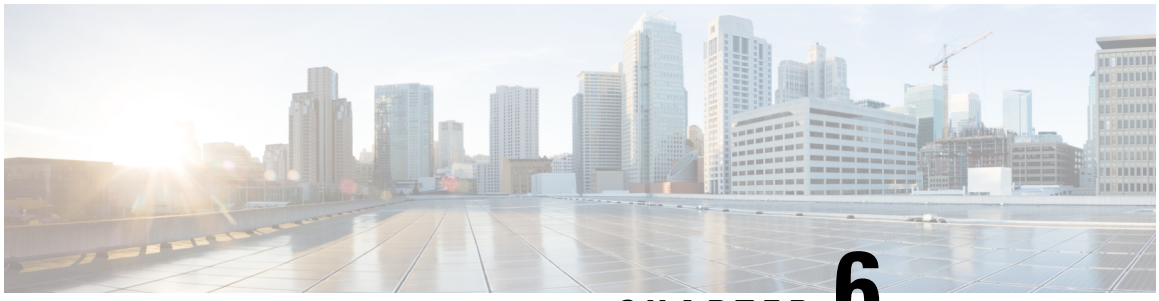
Unable to commit changes using the CLI

Possible Causes

There could be typos or incorrect queue names entered while committing the changes. For example, if you type queuee 2 instead of queue 2, the following error is displayed: Aborted: illegal reference 'policy vmanage-traffic-forwarding-class'

Solution

Enter the right queue name that you want the Cisco SD-WAN Manager traffic from the router to flow through.



CHAPTER 6

Default AAR and QoS Policies

Note To achieve simplification and consistency, the Cisco SD-WAN solution has been rebranded as Cisco Catalyst SD-WAN. In addition, from Cisco IOS XE SD-WAN Release 17.12.1a and Cisco Catalyst SD-WAN Release 20.12.1, the following component changes are applicable: **Cisco vManage** to **Cisco Catalyst SD-WAN Manager**, **Cisco vAnalytics** to **Cisco Catalyst SD-WAN Analytics**, **Cisco vBond** to **Cisco Catalyst SD-WAN Validator**, **Cisco vSmart** to **Cisco Catalyst SD-WAN Controller**, and **Cisco Controllers** to **Cisco Catalyst SD-WAN Control Components**. See the latest Release Notes for a comprehensive list of all the component brand name changes. While we transition to the new names, some inconsistencies might be present in the documentation set because of a phased approach to the user interface updates of the software product.

Table 18: Feature History

Feature Name	Release Information	Description
Configure Default AAR and QoS Policies	Cisco SD-WAN Release 20.7.1 Cisco vManage Release 20.7.1	This feature enables you to efficiently configure default application-aware routing (AAR), data, and quality of service (QoS) policies for Cisco SD-WAN devices. The feature provides a step-by-step workflow for categorizing the business relevance, path preference, and other parameters for network applications, and applying those preferences as traffic policy.

Information About Default AAR and QoS Policies

It is often helpful to create an AAR policy, a data policy, and a QoS policy for devices in a network. These policies route and prioritize traffic for best performance. When creating these policies, it is helpful to distinguish

among the applications producing network traffic, based on the likely business relevance of the applications, and to give higher priority to business-relevant applications.

Cisco SD-WAN Manager provides an efficient workflow to help you create a default set of AAR, data, and QoS policies to apply to devices in the network. The workflow presents a set of more than 1000 applications that can be identified by network-based application recognition (NBAR), an application recognition technology built into Cisco Catalyst SD-WAN devices. The workflow groups the applications into one of three business-relevance categories:

- Business-relevant: Likely to be important for business operations, for example, Webex software.
- Business-irrelevant: Unlikely to be important for business operations, for example, gaming software.
- Default: No determination of relevance to business operations.

Within each of the business-relevance categories, the workflow groups the applications into application lists, such as broadcast video, multimedia conferencing, VoIP telephony, and so on.

Using the workflow, you can accept the predefined categorization of each application's business relevance or you can customize the categorization of specific applications by moving them from one of the business-relevance categories to another. For example, if, by default, the workflow predefines a specific application as business-irrelevant, but that application is important for your business operations, then you can recategorize the application as Business-relevant.

The workflow provides a step-by-step procedure for configuring the business relevance, path preference, and service level agreement (SLA) category.

After you complete the workflow, Cisco SD-WAN Manager produces a default set of the following:

- AAR policy
- QoS policy
- Data policy

After you attach these policies to a centralized policy, you can apply these default policies to Cisco Catalyst SD-WAN devices in the network.

Background Information About NBAR

NBAR is an application recognition technology included in Cisco Catalyst SD-WAN devices. NBAR uses a set of application definitions called protocols to identify and categorize traffic. One of the categories that it assigns to traffic is the business-relevance attribute. The values of this attribute are Business-relevant, Business-irrelevant, and Default. In developing protocols to identify applications, Cisco estimates whether an application is likely to be important for typical business operations, and assigns a business-relevance value to the application. The default AAR and QoS policy feature uses the business-relevance categorization provided by NBAR.

Benefits of Default AAR and QoS Policies

- Manage and customize bandwidth allocations.
- Prioritize applications based on their relevance to your business.

Prerequisites for Default AAR and QoS Policies

- Knowledge about the relevant applications.
- Familiarity with the SLAs and QoS markings to prioritize traffic.

Restrictions for Default AAR and QoS Policies

- When you customize a business-relevant application group, you cannot move all the applications from that group to another section. Application groups of business-relevant section need to have at least one application in them.
- Default AAR and QoS policies do not support IPv6 addressing.

Supported Devices for Default AAR and QoS Policies

- Cisco 1000 Series Integrated Services Routers (ISR1100-4G and ISR1100-6G)
- Cisco 4000 Series Integrated Services Routers (ISR44xx)
- Cisco Catalyst 8000V Edge Software
- Cisco Catalyst 8300 Series Edge Platforms
- Cisco Catalyst 8500 Series Edge Platforms
- Cisco C1100 Series Integrated Services Router

Use Cases for Default AAR and QoS Policies

If you are setting up a Cisco Catalyst SD-WAN network and want to apply an AAR and a QoS policy to all the devices in a network, use this feature to create and deploy these policies quickly.

Configure Default AAR and QoS Policies Using Cisco SD-WAN Manager

Follow these steps to configure default AAR, data, and QoS policies using Cisco SD-WAN Manager:

1. From the Cisco SD-WAN Manager menu, choose **Configuration** > **Policies**.
2. Click **Add Default AAR & QoS**.

   The **Process Overview** page is displayed.
3. Click **Next**.

The **Recommended Settings based on your selection** page is displayed.

4. Based on the requirements of your network, move the applications between the **Business Relevant**, **Default**, and **Business Irrelevant** groups.

Note When customizing the categorization of applications as Business-relevant, Business-irrelevant, or Default, you can only move individual applications from one category to another. You cannot move an entire group from one category to another.

5. Click **Next**.

   On the **Path Preferences (optional)** page, choose the **Preferred** and **Preferred Backup** transports for each traffic class.

6. Click **Next**.

   The **App Route Policy Service Level Agreement (SLA) Class** page is displayed.

   This page shows the default settings for **Loss**, **Latency**, and **Jitter** values for each traffic class. If necessary, customize **Loss**, **Latency**, and **Jitter** values for each traffic class.

7. Click **Next**.

   The **Enterprise to Service Provider Class Mapping** page is displayed.

   a. Select a service provider class option, based on how you want to customize bandwidth for different queues. For further detials on QoS queues, refer to the section **Mapping of Application Lists to Queues**

   b. If necessary, customize the bandwidth percentage values for each queues.

8. Click **Next**.

   The **Define prefixes for the default policies and applications lists** page is displayed.

   For each policy, enter a prefix name and description.

9. Click **Next**.

   The **Summary** page is displayed. On this page, you can view the details for each configuration.

   You can click **Edit** to edit the options that appeared earlier in the workflow. Clicking edit returns you to the relevant page.

10. Click **Configure**.

    Cisco SD-WAN Manager creates the AAR, data, and QoS policies and indicates when the process is complete.

    The following table describes the workflow steps or actions and their respective effects:

Table 19: Workflow Steps and Effects

Workflow Step	Affects the Following
Recommended Settings based on your selection	AAR and data policies
Path Preferences (optional)	AAR policies

Workflow Step	Affects the Following
App Route Policy Service Level Agreement (SLA) Class: • Loss • Latency • Jitter	AAR policies
Enterprise to Service Provider Class Mapping	Data and QoS policies
Define prefixes for the default policies and applications	AAR, data, QoS policies, forwarding classes, application lists, SLA class lists

11. To view the policy, click **View Your Created Policy**.

Note To apply the default AAR and QoS policies to the devices in the network, create a centralized policy that attaches the AAR and data policies to the required site lists. To apply the QoS policy to the Cisco SD-WAN devices, attach it to a localized policy through device templates.

Mapping of Application Lists to Queues

The following lists show each service provider class option, the queues in each option, and the application lists included in each queue. The application lists are named here as they appear on the Path Preferences page in this workflow.

4 QoS class

- Voice
  - Internetwork control
  - VoIP telephony
- Mission critical
  - Broadcast video
  - Multimedia conferencing
  - Real-Time interactive
  - Multimedia streaming
- Business data
  - Signaling
  - Transactional data
  - Network management
  - Bulk data

- Default
  - Best effort
  - Scavenger

5 QoS class

- Voice
  - Internetwork control
  - VoIP telephony
- Mission critical
  - Broadcast video
  - Multimedia conferencing
  - Real-Time interactive
  - Multimedia streaming
- Business data
  - Signaling
  - Transactional data
  - Network management
  - Bulk data
- General data
  - Scavenger
- Default
  - Best effort

6 QoS class

- Voice
  - Internetwork control
  - VoIP telephony
- Video
  - Broadcast video
  - Multimedia conferencing
  - Real-Time interactive
- Mission Critical

- Multimedia streaming

- Business data
  - Signaling
  - Transactional data
  - Network management
  - Bulk data

- General data
  - Scavenger

- Default
  - Best effort

8 QoS class

- Voice
  - VoIP telephony

- Net-ctrl-mgmt
  - Internetwork control

- Interactive video
  - Multimedia conferencing
  - Real-Time interactive

- Streaming video
  - Broadcast video
  - Multimedia streaming

- Call signaling
  - Signaling

- Critical data
  - Transactional data
  - Network management
  - Bulk data

- Scavengers
  - Scavenger

- Default
  - Best effort

Monitor Default AAR and QoS Policies

Monitor Default AAR Policies

1. From the Cisco SD-WAN Manager menu, choose **Configuration** > **Policies**.
2. Click **Custom Options**.
3. Choose **Traffic Policy** from **Centralized Policy**.
4. Click **Application Aware Routing**.

   A list of AAR policies is displayed.
5. Click **Traffic Data**.

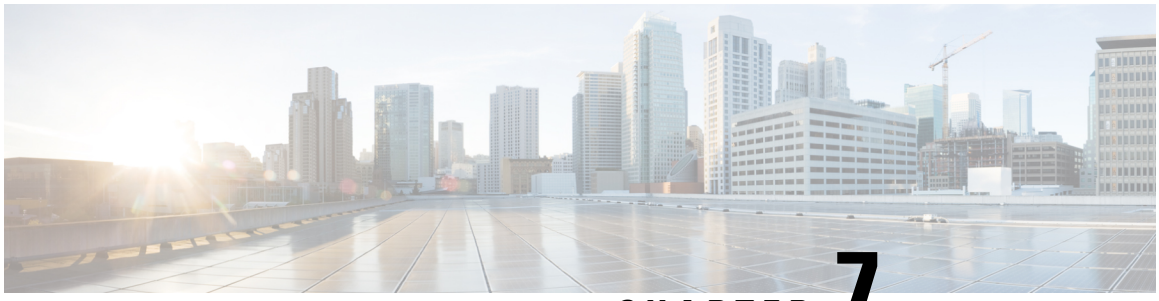
   A list of traffic data policies is displayed.

Monitor QoS Policies

1. From the Cisco SD-WAN Manager menu, choose **Configuration** > **Policies**.
2. Click **Custom Options**.
3. Choose **Forwarding Class/QoS** from **Localized Policy**.
4. Click **QoS Map**.

   A list of QoS policies is displayed.

Note To verify QoS polices, refer to Verify QoS Policy.



CHAPTER 7

Device Access Policy

Note To achieve simplification and consistency, the Cisco SD-WAN solution has been rebranded as Cisco Catalyst SD-WAN. In addition, from Cisco IOS XE SD-WAN Release 17.12.1a and Cisco Catalyst SD-WAN Release 20.12.1, the following component changes are applicable: **Cisco vManage** to **Cisco Catalyst SD-WAN Manager**, **Cisco vAnalytics** to **Cisco Catalyst SD-WAN Analytics**, **Cisco vBond** to **Cisco Catalyst SD-WAN Validator**, **Cisco vSmart** to **Cisco Catalyst SD-WAN Controller**, and **Cisco Controllers** to **Cisco Catalyst SD-WAN Control Components**. See the latest Release Notes for a comprehensive list of all the component brand name changes. While we transition to the new names, some inconsistencies might be present in the documentation set because of a phased approach to the user interface updates of the software product.

Table 20: Feature History

Feature Name	Release Information	Description
Device Access Policy on SNMP and SSH	Cisco SD-WAN Release 20.1.1	This feature defines the rules that traffic must meet to pass through an interface. When you define rules for incoming traffic, they are applied to the traffic before any other policies are applied. The control plane of a Cisco vEdge device processes the data traffic for local services (like SSH and SNMP) from a set of sources. Routing packets are required to form the overlay.
Device Access Policy on SNMP and SSH	Cisco SD-WAN Release 19.3.x	This feature defines the rules that traffic must meet to pass through an interface. When you define rules for incoming traffic on Cisco vEdge devices, they are applied to the traffic before any other policies are applied.

Device Access Policy Overview

Starting from Cisco SD-WAN Release 19.3, the Cisco SD-WAN Manager user interface is enhanced to configure device access policy on all the Cisco Catalyst SD-WAN devices.

The control plane of Cisco vEdge devices process the data traffic for local services like, SSH and SNMP, from a set of sources. It is important to protect the CPU from device access traffic by applying the filter to avoid malicious traffic.

Access policies define the rules that traffic must meet to pass through an interface. When you define rules for incoming traffic, they are applied to the traffic before any other policies are applied. You can use access policies, in routed and transparent firewall mode to control IP traffic. An access rule permits or denies traffic based on the protocol used, the source and destination IP address or network, and optionally, the users and user groups. Each incoming packet at an interface is analyzed to determine if it must be forwarded or dropped based on criteria you specify. If you define access rules for the outgoing traffic, packets are also analyzed before they are allowed to leave an interface. Access policies are applied in order. That is, when the device compares a packet to the rules, it searches from top to bottom in the access policies list, and applies the policy for the first matched rule, ignoring all subsequent rules (even if a later rule is a better match). Thus, you should place specific rules above more general rules to ensure the specific rules are not skipped.

Configure Device Access Policy Using Cisco SD-WAN Manager

Cisco vEdge devices support device access policy configuration to handle SNMP and SSH traffic directed towards the control plane. Use Cisco SD-WAN Manager to configure destination ports based on the device access policy.

Note In order to allow connections to devices from **Tools** > **SSH Terminal** in Cisco SD-WAN Manager, create a rule to accept **Device Access Protocol** as SSH and **Source Data Prefix** as 192.168.1.5/32.

To configure localized device access control policies, use the Cisco SD-WAN Manager policy configuration wizard.

Configure specific or all components depending on the specific policy you are creating. To skip a component, click the **Next** button. To return to a component, click the **Back** button at the bottom of the screen.

To configure a device access policy:

1. From the Cisco SD-WAN Manager menu, choose **Configuration** > **Policies**.
2. Click **Localized Policy** and from the **Custom Options** drop-down, under **Localized Policy**, select **Access Control Lists.**
3. From the **Add Device Access Policy** drop-down list, select **Add IPv4 Device Access Policy** or **Add IPv6 Device Access Policy** option to add a device.

Note Beginning with Cisco IOS XE Catalyst SD-WAN Release 17.9.1a and Cisco vManage Release 20.9.1, if you configure an IPv4 or an IPv6 device access policy with no policy sequences and only a default action of **Accept** or **Drop**, the device access policy creates an SSH and an SNMP configuration. You can now create a device access policy with only a default action and with no policy sequences to create a device configuration or a Cisco SD-WAN Manager configuration for both SSH and SNMP.

If you do not create an SNMP server configuration, the SNMP configuration created by the device access policy is unused.

Prior to Cisco IOS XE Catalyst SD-WAN Release 17.9.1a and Cisco vManage Release 20.9.1, if you configured a device access policy with only a default action of **Accept** or **Drop** and with no policy sequences, the device access policy would not create a device configuration or a Cisco SD-WAN Manager configuration.

4. Select **Add IPv4 Device Access Policy** from the drop-down list to add an **IPv4 ACL Policy**. The edit **Device IPv4 ACL Policy** page appears.
5. Enter the name and the description for the new policy.
6. Click **Add ACL Sequence** to add a sequence. The **Device Access Control List** page is displayed.
7. Click **Sequence Rule**. **Match** and **Actions** options are displayed.
8. Click **Match**, select and configure the following conditions for your ACL policy:

Match Condition	Description
Device Access Protocol (required)	Select a carrier from the drop-down list. For example SNMP, SSH.
Source Data Prefix	Enter the source IP address. For example, 10.0.0.0/12.
Source Port	Enter the list of source ports. The range is 0-65535.
Destination Data Prefix	Enter the destination IP address. For example, 10.0.0.0/12.
VPN	Enter the VPN ID. The range is 0-65536.

9. Click **Actions**, configure the following conditions for your ACL policy:

Action Condition	Description
Accept	
Counter Name	Enter the counter name to be accepted. The maximum length can be 20 characters.
Drop	
Counter Name	Enter the counter name to drop. The maximum length can be 20 characters.

10. Click **Save Match And Actions** to save all the conditions for the ACL policy.

11. Click **Save Device Access Control List Policy** to apply the selected match conditions to an action.
12. If no packets match, then any of the route policy sequence rules. The **Default Action** in the left pane is to drop the packets.

Note IPv6 prefix match is not supported on Cisco vEdge devices. When you try to configure IPv6 prefix matches on these devices, Cisco SD-WAN Manager fails to generate device configuration.

Configure Device Access Policy Using the CLI

Configuration:

```
Device(config)# policy
Device(config-policy) policy device-access-policy <name>
    sequence 1
        match
            destination-data-prefix-list    Destination prefix list
            destination-ip                  List of destination addresses
            destination-port                List of destination ports
            dscp                            List of DSCP values
            packet-length                   Packet length
            protocol                        List of protocols
            source-data-prefix-list         Source prefix list
            source-ip                       List of source addresses
            source-port                     List of source ports
            destination-vpn                 List of VPN-ID
        action
            accept
            count                           Number of packets/bytes matching this rule
            drop
    default-action       Accept or drop

system
    device-access-policy ipv4 <pol-name>
```

Note IPv6 prefix match is not supported on Cisco SD-WANs.

The following example shows the sample configuration for device access policy:

```
policy device-access-policy dev_pol
  sequence 1
   match
    destination-port 22
   !
   action drop
    count ssh_packs
   !
  !
  default-action drop
 !
 device-access-policy snmp_policy
  sequence 2
   match
    destination-port 161
```

```
   !
   action drop
    count snmp_packs
   !
  !
  default-action accept
 !
!
system
 device-access-policy ipv4 snmp_policy
!
```

Verifying Device Access Policy Configuration

Cisco vEdge devices support the following operational commands to provide information for a device-access-policy. These commands provide a visual for the counters and the names of the configured device-access-policy. The two commands and the respective yang models are shown in the following sections.

Yang model for the command **device-access-policy-counters**:

```
list device-access-policy-counters {
        tailf:info "IPv6 Device Access Policy counters";
        when "/viptela-system:system/viptela-system:personality = 'vedge'";
        tailf:callpoint device-access-policy-counters-v6; // _nfvis_exclude_line_
        key "name";
        tailf:hidden cli;

        leaf name {
          tailf:info "Device Access Policy name";
          type viptela:named-type-127;
        }
        config false;
        list device-access-policy-counter-list {
          tailf:info "Device access policy counter list";
          tailf:callpoint device-access-policy-counter-list-v6; // _nfvis_exclude_line_
          tailf:cli-no-key-completion;
          tailf:cli-suppress-show-match;
          key "counter-name";
          tailf:hidden cli;

          leaf counter-name {
            tailf:info "Counter name";
            tailf:cli-suppress-show-match;
            type viptela:named-type;
          }
          leaf packets {
            type yang:counter64;
            tailf:cli-suppress-show-match;
          }
          leaf bytes {
            type yang:counter64;
            tailf:cli-suppress-show-match;
          }
        }
```

The following example shows the policy details of a counter.

```
show policy device-access-policy-counters

                  COUNTER
NAME          NAME        PACKETS   BYTES
--------------------------------------------------------------
```

```
dev_pol       ssh_packs      -       -
snmp_policy   snmp_packs     0       0
```

Yang model for the command **device-access-policy**:

```
list device-access-policy {
      tailf:info "Configure IPv4 device-access policy";
      key "name";
      when "/viptela-system:system/viptela-system:personality = 'vedge'";

      leaf name {
        tailf:info "Name of IPv4 device-access policy";
        type viptela:named-type-127;
      }

      list sequence {
        tailf:info "List of sequences";
        key "seq-value";

        leaf seq-value {
          tailf:info "Sequence value";
          type uint16 {
            tailf:info "<0..65530>";
            range "0..65530";
          }
        }

        container match {
          tailf:info "Match criteria";
          tailf:cli-add-mode;

          choice source {
            case prefix {
              leaf-list source-ip {
                tailf:info "List of source addresses";
                tailf:cli-flat-list-syntax;
                tailf:cli-replace-all;
                type inet:ipv4-prefix;
              }
            }

            case prefix-list {
              leaf source-data-prefix-list {
                tailf:info "Source prefix list";

                type leafref {
                  path "../../../../lists/data-prefix-list/name";
                }
              }
            }

          choice destination {
            case prefix {
              leaf-list destination-ip {
                tailf:info "List of destination addresses";
                tailf:cli-flat-list-syntax;
                tailf:cli-replace-all;
                type inet:ipv4-prefix;
              }
            }

            case prefix-list {
              leaf destination-data-prefix-list {
                tailf:info "Destination prefix list";
```

```
        type leafref {
          path "../../../../lists/data-prefix-list/name";
        }
      }
    }
  }

  leaf-list source-port {
    tailf:info "List of source ports";
    tailf:validate port_range {
      tailf:call-once 'true';
      tailf:dependency '.';
    }
    tailf:cli-flat-list-syntax;
    tailf:cli-replace-all;
    type viptela:range-type {
      tailf:info "<0..65535> or range";
    }
  }

  leaf-list destination-port {
    tailf:info "List of destination ports";
    tailf:validate port_range {
      tailf:call-once 'true';
      tailf:dependency '.';
    }
    tailf:cli-flat-list-syntax;
    tailf:cli-replace-all;
    type viptela:range-type {
      tailf:info "<0..65535> or range";
    }
  }

  leaf-list destination-vpn {
    tailf:info "List of VPN ID";
    tailf:validate port_range {
      tailf:call-once 'true';
      tailf:dependency '.';
    }
    tailf:cli-flat-list-syntax;
    tailf:cli-replace-all;
    type viptela:range-type {
      tailf:info "<0..65535> or range";
    }
  }

}

container action {
  tailf:cli-add-mode;
  tailf:cli-incomplete-command;
  tailf:info "Accept or drop";

  leaf action-value {
    tailf:cli-hide-in-submode;
    tailf:cli-drop-node-name;
    tailf:cli-show-with-default;
    type action-data-enum;
  }

  leaf count {
    tailf:info "Number of packets/bytes matching this rule";
    type string {
      tailf:info "<1..32 characters>";
```

```
            length '1..32';
          }
        }
      }
    }

    leaf default-action {
      tailf:cli-show-with-default;
      tailf:info "Accept or drop";
      type action-data-enum;
    }
  }
```

Yang model for the command **device-access-policy-names**:

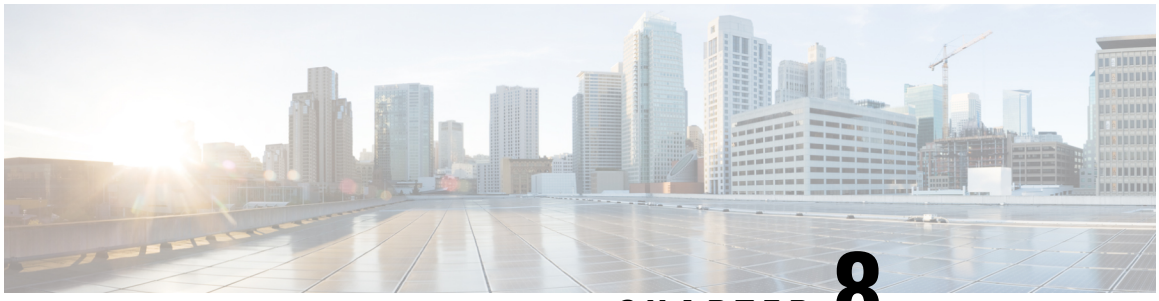
```
list device-access-policy-names {
        tailf:info "IPv6 device access policy names";
        when "/viptela-system:system/viptela-system:personality = 'vedge'";
        tailf:callpoint device-access-policy-names-v6; // _nfvis_exclude_line_
        tailf:cli-no-key-completion;
        key "name";
        tailf:hidden cli;

        leaf name {
          tailf:info "Device Access Policy name";
          type viptela:named-type-127;
        }
        config false;
      }
```

The following example shows the list of configured policies:

```
show policy device-access-policy-names

NAME
-------------
dev_pol
snmp_policy
```



CHAPTER 8

Cisco Catalyst SD-WAN Application Intelligence Engine Flow

Note To achieve simplification and consistency, the Cisco SD-WAN solution has been rebranded as Cisco Catalyst SD-WAN. In addition, from Cisco IOS XE SD-WAN Release 17.12.1a and Cisco Catalyst SD-WAN Release 20.12.1, the following component changes are applicable: **Cisco vManage** to **Cisco Catalyst SD-WAN Manager**, **Cisco vAnalytics** to **Cisco Catalyst SD-WAN Analytics**, **Cisco vBond** to **Cisco Catalyst SD-WAN Validator**, **Cisco vSmart** to **Cisco Catalyst SD-WAN Controller**, and **Cisco Controllers** to **Cisco Catalyst SD-WAN Control Components**. See the latest Release Notes for a comprehensive list of all the component brand name changes. While we transition to the new names, some inconsistencies might be present in the documentation set because of a phased approach to the user interface updates of the software product.

The topics in this section provide overview information about the Cisco Catalyst SD-WAN Application Intelligence Engine (SAIE) flow, and how to configure the flow using Cisco SD-WAN Manager or the CLI.

Cisco Catalyst SD-WAN Application Intelligence Engine Flow Overview

The Cisco Catalyst SD-WAN Application Intelligence Engine (SAIE) flow provides the ability to look into the packet past the basic header information. The SAIE flow determines the contents of a particular packet, and then either records that information for statistical purposes or performs an action on the packet.

Note In Cisco vManage Release 20.7.1 and earlier releases, the SAIE flow is called the deep packet inspection (DPI) flow.

Benefits include increased visibility into the network traffic, which enables network operators to understand usage patterns and to correlate network performance information along with providing usage base billing or even acceptable usage monitoring. The SAIE flow can also reduce the overall costs on the network.

You can configure the SAIE flow using a centralized data policy. You define the applications of interest in a Cisco SD-WAN Manager policy list or with the **policy lists app-list** CLI command, and you call these lists in a **policy data-policy** command. You can control the path of the application traffic through the network by defining, in the **action** portion of the data policy, the local TLOC or the remote TLOC, or for strict control, you can define both.

The following list of protocols are not supported in SAIE flow:

- Open Shortest Path First (OSPF)
- Border Gateway Protocol (BGP)
- Internet Control Message Protocol (ICMP)
- Bidirectional Forwarding Detection (BFD)

Configure Cisco Catalyst SD-WAN Application Intelligence Engine Flow Using Cisco SD-WAN Manager

To configure the Cisco Catalyst SD-WAN Application Intelligence Engine (SAIE) flow, use the Cisco SD-WAN Manager policy configuration wizard. The wizard consists of the following sequential screens that guide you through the process of creating and editing policy components:

- Create Applications or Groups of Interest—Create lists that group together related items and that you call in the match or action components of a policy. For configuration details, see Configure Groups of Interest.
- Configure Traffic Rules—Create the match and action conditions of a policy. For configuration details, see Configure Traffic Rules.
- Apply Policies to Sites and VPNs—Associate policy with sites and VPNs in the overlay network.

Apply Centralized Policy for SD-WAN Application Intelligence Engine Flow

To ensure that a centralized data policy for the SD-WAN Application Intelligence Engine (SAIE) flow takes effect, you must apply it to a list of sites in the overlay network.

To apply a centralized policy in Cisco SD-WAN Manager, see *Configure Centralized Policy Using Cisco SD-WAN Manager*.

To apply a centralized policy in the CLI:

```
vSmart(config)# apply-policy site-list list-name data-policy policy-name (all | from-service
 | from-tunnel)
```

By default, data policy applies to all data traffic passing through the Cisco Catalyst SD-WAN Controller: the policy evaluates all data traffic going from the local site (that is, from the service side of the router) into the tunnel interface, and it evaluates all traffic entering to the local site through the tunnel interface. You can explicitly configure this behavior by including the **all** option. To have the data policy apply only to policy

exiting from the local site, include the **from-service** option. To have the policy apply only to incoming traffic, include the **from-tunnel** option.

You cannot apply the same type of policy to site lists that contain overlapping site IDs. That is, all data policies cannot have overlapping site lists among themselves. If you accidentally misconfigure overlapping site lists, the attempt to commit the configuration on the Cisco Catalyst SD-WAN Controller fails.

As soon as you successfully activate the configuration by issuing a **commit** command, the Cisco Catalyst SD-WAN Controller pushes the data policy to the Cisco vEdge devices located in the specified sites. To view the policy as configured on the Cisco Catalyst SD-WAN Controller, use the **show running-config** command on the Cisco Catalyst SD-WAN Controller:

```
vSmart# show running-config policy
vSmart# show running-config apply-policy
```

To view the policy that has been pushed to the Cisco vEdge device, use the **show policy from-vsmart** command on the Cisco vEdge device.

```
vEdge# show policy from-vsmart
```

Monitor Running Applications

To enable the SD-WAN Application Intelligence Engine (SAIE) infrastructure on Cisco vEdge devices, you must enable application visibility on the devices:

Note In Cisco vManage Release 20.7.x and earlier releases, the SAIE flow is called the deep packet inspection (DPI) flow.

```
vEdge(config)# policy app-visibility
```

To display information about the running applications, use the **show app dpi supported-applications**, **show app dpi applications**, and **show app dpi flows** commands on the device.

View SAIE Applications

You can view the list of all the application-aware applications supported by the Cisco Catalyst SD-WAN software on the router using the following steps:

1. From the Cisco SD-WAN Manager menu, choose **Monitor** > **Devices**.

   Cisco vManage Release 20.6.x and earlier: From the Cisco SD-WAN Manager menu, choose **Monitor** > **Network**.

2. Click **WAN-Edge**, select the **Device** that supports the SD-WAN Application Intelligence Engine (SAIE) flow. The Cisco SD-WAN Manager Control Connections page is displayed.

3. In the left pane, select **Real Time** to view the device details.

4. From the **Device Options** drop-down, choose **SAIE Applications** to view the list of applications running on the device.

5. From the **Device Options** drop-down, choose **SAIE Supported Applications** to view the list of applications that are supported on the device.

Action Parameters for Configuring SD-WAN Application Intelligence Engine Flow

When data traffic matches the conditions in the match portion of a centralized data policy, the packet can be accepted or dropped, and it can be counted. Then, you can associate parameters with accepted packets.

From the Cisco SD-WAN Manager menu, you can configure match parameters from:

- **Configuration** > **Policies** > **Centralized Policy** > **Add Policy** > **Configure Traffic Rules** > **(Application-Aware Routing | Traffic Data | Cflowd)** > **Sequence Type** > **Sequence Rule** > **Action**
- **Configuration** > **Policies** > **Custom Options** > **Centralized Policy** > **Traffic Policy** > **(Application-Aware Routing | Traffic Data | Cflowd)** > **Sequence Type** > **Sequence Rule** > **Action**.

In the CLI, you configure the action parameters under the **policy data-policy vpn-list sequence action** command.

Each sequence in a centralized data policy can contain one action condition.

In the action, you first specify whether to accept or drop a matching data packet, and whether to count it:

Table 21:

Description	Cisco SD-WAN Manager	CLI Command	Value or Range
Accept the packet. An accepted packet is eligible to be modified by the additional parameters configured in the action portion of the policy configuration.	Click **Accept**.	**accept**	—
Count the accepted or dropped packets.	**Action Counter** Click **Accept**, then action **Counter**	**count** *counter-name*	Name of a counter. Use the **show policy access-lists counters** command on the Cisco device.
Discard the packet. This is the default action.	Click **Drop**	**drop**	—
Log the packet. Packets are placed into the messages and vsyslog system logging (syslog) files.	**Action Log** Click **Accept**, then **action Log**	**log**	To view the packet logs, use the **show app log flows** and **show log** commands.

To view the packet logs, use the **show app log flow** and **show log** commands.

Then, for a packet that is accepted, the following parameters can be configured.

Table 22:

Description	Cisco SD-WAN Manager	CLI Command	Value or Range
DSCP value.	Click **Accept**, then action **DSCP**.	**set dscp** *value*	0 through 63
Forwarding class.	Click **Accept**, then action **Forwarding Class**.	**set forwarding-class** *value*	Name of forwarding class

Description	Cisco SD-WAN Manager	CLI Command	Value or Range
Direct matching packets to a TLOC that matches the color and encapsulation By default, if the TLOC is not available, traffic is forwarded using an alternate TLOC.	Click **Accept**, then action **Local TLOC**.	**set local-tloc color** *color* [**encap** *encapsulation*]	*color* can be: **3g**, **biz-internet**, **blue**, **bronze**, **custom1**, **custom2**, **custom3**, **default**, **gold**, **green lte**, **metro-ethernet mpls**, **private1** through **private6**, **public-internet**, **red**, and **silver**. By default, *encapsulation* is **ipsec**. It can also be **gre**.
Direct matching packets to one of the TLOCs in the list if the TLOC matches the color and encapsulation By default, if the TLOC is not available, traffic is forwarded using an alternate TLOC. To drop traffic if a TLOC is unavailable, include the **restrict** option.	Click **Accept**, then action **Local TLOC**	**set local-tloc-list color** *color* **encap** *encapsulation* [**restrict**]	
Set the next hop to which the packet should be forwarded.	Click **Accept**, then action **Next Hop**.	**set next-hop** *ip-address*	IP address
Apply a policer.	Click **Accept**, then action **Policer**.	**set policer** *policer-name*	Name of policer configured with a **policy policer** command.
Direct matching packets to the name service, before delivering the traffic to its ultimate destination. The TLOC address or list of TLOCs identifies the remote TLOCs to which the traffic should be redirected to reach the service. In the case of multiple TLOCs, the traffic is load-balanced among them. The VPN identifier is where the service is located. Configure the services themselves on the Cisco devices that are collocated with the service devices, using the **vpn service** configuration command.	Click **Accept**, then action **Service**.	**set service** *service-name* [**tloc** *ip-address* | **tloc-list** *list-name*] [**vpn** *vpn-id*]	Standard services: **FW**, **IDS**, **IDP** Custom services: **netsvc1**, **netsvc2**,**netsvc3**, **netsvc4** TLOC list is configured with a **policy lists tloc-list** list.
Direct matching packets to the named service that is reachable using a GRE tunnel whose source is in the transport VPN (VPN 0). If the GRE tunnel used to reach the service is down, packet routing falls back to using standard routing. To drop packets when a GRE tunnel to the service is unreachable, include the restrict option. In the service VPN, you must also advertise the service using the **service** command. You configure the GRE interface or interfaces in the transport VPN (VPN 0).	Click **Accept**, then action **Service**.	**set service** *service-name* [**tloc** *ip-address* | **tloc-list** *list-name*] [**vpn** *vpn-id*]	Standard services: **FW**, **IDS**, **IDP** Custom services: **netsvc1**, **netsvc2**,**netsvc3**, **netsvc4**
Direct traffic to a remote TLOC. The TLOC is defined by its IP address, color, and encapsulation.	Click **Accept**, then action **TLOC**.	**set local-tloc color** *color* [**encap** *encapsulation*]	TLOC address, color, and encapsulation

Description	Cisco SD-WAN Manager	CLI Command	Value or Range
Direct traffic to one of the remote TLOCs in the TLOC list.	Click **Accept**, then action **TLOC**.	**set tloc-list** *list-name*	Name of a **policy lists tloc-list** list
Set the VPN that the packet is part of.	Click **Accept**, then action **VPN**.	**set vpn** *vpn-id*	0 through 65530

Default Action

If a data packet being evaluated does not match any of the match conditions in a data policy, a default action is applied to the packet. By default, the data packet is dropped.

From the Cisco SD-WAN Manager menu, you modify the default action from **Configuration** > **Policies** > **Centralized Policy** > **Add Policy** > **Configure Traffic Rules** > **Application-Aware Routing** > **Sequence Type** > **Sequence Rule** > **Default Action**.

In the CLI, you modify the default action with the **policy data-policy vpn-list default-action accept** command.

Configure SD-WAN Application Intelligence Engine Flow Using the CLI

Following are the high-level steps for configuring a centralized data policy for the SD-WAN Application Intelligence Engine (SAIE) flow.

Note In Cisco vManage Release 20.7.x and earlier releases, the SAIE flow is called the deep packet inspection (DPI) flow.

1. Create a list of overlay network sites to which the data policy is to be applied using the **apply-policy** command:

   ```
   vSmart(config)# policy
   vSmart(config-policy)# lists site-list list-name
   vSmart(config-lists-list-name)# site-id site-id
   ```

   The list can contain as many site IDs as necessary. Include one **site-id** command for each site ID. For contiguous site IDs, you can specify a range of numbers separated with a dash (–).

   Create additional site lists, as needed.

2. Create lists of applications and application families that are to be subject to the data policy. Each list can contain one or more application names, or one or more application families. A single list cannot contain both applications and application families.

   ```
   vSmart(config)# policy lists
   vSmart(config-lists)# app-list list-name
   vSmart(config-app-list)# app application-name

   vSmart(config)# policy lists
   vSmart(config-lists)# app-list list-name
   vSmart(config-applist)# app-family family-name
   ```

3. Create lists of IP prefixes and VPNs, as needed:

```
vSmart(config)# policy lists
vSmart(config-lists)# data-prefix-list list-name
vSmart(config-lists-list-name)# ip-prefix prefix/length

vSmart(config)# policy lists
vSmart(config-lists)# vpn-list list-name
vSmart(config-lists-list-name)# vpn vpn-id
```

4. Create lists of TLOCs, as needed:

```
vSmart(config)# policy
vSmart(config-policy)# lists tloc-list list-name
vSmart(config-lists-list-name)# tloc ip-address color color encap encapsulation
[preference number]
```

5. Define policing parameters, as needed:

```
vSmart(config-policy)# policer policer-name
vSmart(config-policer)# rate bandwidth
vSmart(config-policer)# burst bytes
vSmart(config-policer)# exceed action
```

6. Create a data policy instance and associate it with a list of VPNs:

```
vSmart(config)# policy data-policy policy-name
vSmart(config-data-policy-policy-name)# vpn-list list-name
```

7. Create a series of match–pair sequences:

```
vSmart(config-vpn-list)# sequence number
vSmart(config-sequence-number)#
```

   The match–action pairs are evaluated in order, by sequence number, starting with the lowest numbered pair and ending when the route matches the conditions in one of the pairs. Or if no match occurs, the default action is taken (either rejecting the route or accepting it as is).

8. Define match parameters based on applications:

```
vSmart(config-sequence-number)# match app-list list-name
```

9. Define additional match parameters for data packets:

```
vSmart(config-sequence-number)# match parameters
```

10. Define actions to take when a match occurs:

```
vSmart(config-sequence-number)# action (accept | drop) [count]
```

11. For packets that are accepted, define the actions to take. To control the tunnel over which the packets travels, define the remote or local TLOC, or for strict control over the tunnel path, set both:

```
vSmart(config-action)# set tloc ip-address color color encap encapsulation
vSmart(config-action)# set tloc-list list-name
vSmart(config-action)# set local-tloc color color encap encapsulation
vSmart(config-action)# set local-tloc-list color color encap encapsulation [restrict]
```

12. Define additional actions to take.

13. Create additional numbered sequences of match–action pairs within the data policy, as needed.

14. If a route does not match any of the conditions in one of the sequences, it is rejected by default. If you want nonmatching prefixes to be accepted, configure the default action for the policy:

```
vSmart(config-policy-name)# default-action accept
```

15. Apply the policy to one or more sites in the overlay network:

```
vSmart(config)# apply-policy site-list list-name data-policy policy-name (all |
from-service | from-tunnel)
```

```
vEdge(config)# policy app-visibility
```

Use the following show commands for visibility in to traffic classification:

- `show app dpi flows`
- `show support dpi flows active detail`
- `show app dpi application`
- `show support dpi flows expired detail`
- `show support dpi statistics`

Traffic Classification Using NBAR

Table 23: Feature History

Feature	Release Information	Description
Traffic Classification Using NBAR	Cisco SD-WAN Release 20.6.1 Cisco vManage Release 20.6.1	This feature extends Network-Based Application Recognition (NBAR) support to Cisco SD-WAN vEdge devices.

Information about NBAR

Starting from Cisco SD-WAN Release 20.6.1, Cisco vEdge devices use Network-Based Application Recognition as the SD-WAN Application Intelligence Engine (SAIE).

Note In Cisco vManage Release 20.7.x and earlier releases, the SAIE flow is called the deep packet inspection (DPI) flow.

Cisco NBAR is a classification engine that recognizes and classifies a wide variety of protocols and applications. It performs the SAIE flow on network traffic to identify network applications according to their traffic characteristics.

The specific traffic characteristics of a network application are called an application signatures. Cisco packages the signature for an application, together with other information, as a protocol. Cisco packages a large set of protocols, covering numerous commonly occurring network applications, as a Protocol Pack. Cisco updates and distributes Protocol Packs regularly. They provide a database of network application signatures for NBAR to use to identify network application traffic.

The term network applications is defined broadly, and may include all of the following, and more:

- Social media websites

- Voice over IP (VoIP) applications
- Streaming audio and video, such as Cisco Webex
- Cloud applications, such as for cloud storage
- SaaS applications
- Custom network applications specific to an organization

Identifying applications is useful for monitoring network traffic, configuring application-aware traffic policy, and more.

To summarize network application signatures, protocols, and Protocol Packs, and how NBAR uses them:

- The traffic of a network application has unique characteristics that can be used to identify the traffic as belonging to that specific application. These characteristics are called application signatures.
- Cisco packages the signature for a specific network application as a protocol.
- Cisco packages a large set of protocols, covering commonly occurring internet applications, as Protocol Packs.
- Cisco NBAR performs deep packet inspection on traffic to gather the information required to identify the sources of the traffic, and uses protocols, such as those provided in Protocol Packs, to match that information to specific network applications. The result is that NBAR identifies the network applications producing traffic in the network.

Integration with NBAR

Upgrading the Cisco SD-WAN controllers and Cisco vEdge devices to Cisco SD-WAN Release 20.6.1 enables use of NBAR. The introduction of NBAR as the SD-WAN Application Intelligence Engine (SAIE) in this release may affect application-aware centralized policy. We recommend upgrading in the following order. For each step, see the notes regarding NBAR and application-aware policy.

Note In Cisco vManage Release 20.7.x and earlier releases, the SAIE flow is called the deep packet inspection (DPI) flow.

1. Cisco vManage upgrade

Note During the upgrade to Cisco vManage Release 20.6.1, Cisco vManage indicates unsupported applications or any mappings that are missing in the configuration through a warning message. After the upgrade, when you log in, a dialog box displays the policy, app-list name, unsupported applications, and mappings that are missing in the app-list. You can download the information to review and modify the application-aware policies.

2. Cisco vSmart Controller and Cisco vBond Controller upgrade

Note The upgrade process for the Cisco SD-WAN controllers checks any existing application-aware centralized policies to determine whether they use applications not supported by NBAR. If so, the upgrade process prompts you with information about how to update the policies to be compatible with NBAR.

3. Cisco vEdge device upgrade

Note After the upgrade to Cisco SD-WAN Release 20.6.1, Cisco vEdge devices use NBAR for the SAIE flow. The application IDs in the cFlowd records are exported to external collectors from Cisco vEdge device. The application IDs correspond to NBAR and maps to application names accordingly. Application ID to application name mapping is available in NBAR Protocol Pack or in Cisco vEdge device show command, **show app dpi supported-applications app-id**.

For information about upgrading the software and best practices, see Upgrade the Software.

In Cisco vManage, you can configure policy using one of the following two methods:

1. Policy builder: You can choose the required applications from a list of supported applications to use in a policy. Cisco vManage maps these application names to the applications that are compatible with NBAR and generates SAIE-compatible application names.
2. Templates: With policy and device CLI templates, you can create custom policies that can include names of applications that are specific to SAIE.

When you upgrade Cisco vSmart Controller and push any policies or templates, the device generates syslogs, SNMP traps, and Netconf notifications for any application mismatch. You can view the notifications on the **Monitor** > **Events** page. The notification message lists the application names and the renamed application names.

You can view the alarms related to unsupported applications and the applications that need renaming on the **Monitor** > **Alarms** page. The alarm message lists the unsupported application names.

You can view the applications specified by application-aware policies using the `show app dpi applications` command. For information about this command, see show app dpi applications.

Supported Platforms for Traffic Classification Using NBAR

The following is a list of Cisco vEdge devices that support NBAR:

- vEdge 100b
- vEdge 100m
- vEdge 1000
- vEdge 2000
- vEdge 5000
- vEdge Cloud Router
- ISR1100- 4G

- ISR1100- 6G

Benefits of Using NBAR

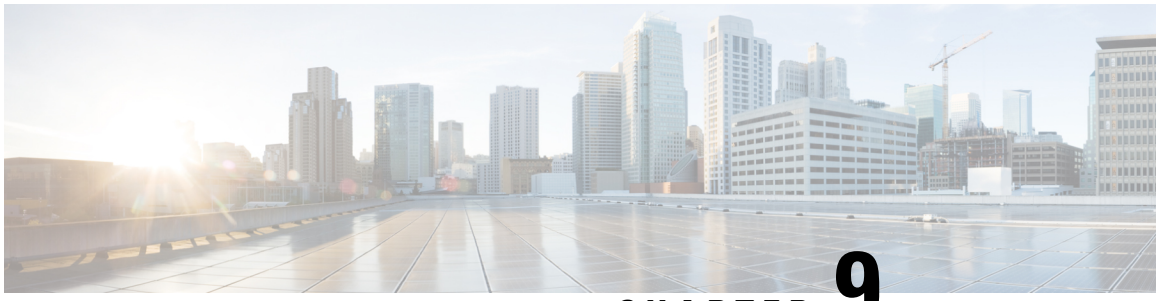
- Consistent application classification behavior across Cisco SD-WAN platforms and mixed deployments.
- Access to all NBAR supported applications through Cisco vManage.
- Better performance and throughput with NBAR.
- NBAR supports better sub-classification for enterprise grade applications.

Restrictions for Traffic Classification Using NBAR

- The following features are not supported on Cisco vEdge devices in Cisco SD-WAN Release 20.6.1:
  - Custom Applications
  - Cisco Software-Defined Application Visibility and Control (SD-AVC)

Note These features also were not supported on Cisco vEdge devices in earlier releases.

- Cisco vManage displays alarms on the **Monitor** > **Alarms** page for custom applications to indicate that Cisco vEdge devices do not support custom applications.



CHAPTER 9

Custom Applications

Note To achieve simplification and consistency, the Cisco SD-WAN solution has been rebranded as Cisco Catalyst SD-WAN. In addition, from Cisco IOS XE SD-WAN Release 17.12.1a and Cisco Catalyst SD-WAN Release 20.12.1, the following component changes are applicable: **Cisco vManage** to **Cisco Catalyst SD-WAN Manager**, **Cisco vAnalytics** to **Cisco Catalyst SD-WAN Analytics**, **Cisco vBond** to **Cisco Catalyst SD-WAN Validator**, **Cisco vSmart** to **Cisco Catalyst SD-WAN Controller**, and **Cisco Controllers** to **Cisco Catalyst SD-WAN Control Components**. See the latest Release Notes for a comprehensive list of all the component brand name changes. While we transition to the new names, some inconsistencies might be present in the documentation set because of a phased approach to the user interface updates of the software product.

Table 24: Feature History

Feature Name	Release Information	Description
Custom Applications	Cisco SD-WAN Release 20.9.1 Cisco vManage Release 20.9.1	This feature enables you to define custom applications using Cisco Software-Defined Application Visibility and Control (SD-AVC) support. This feature is available only on Cisco vEdge devices.

Information About Custom Applications

Cisco Network-Based Application Recognition (NBAR) is a Cisco technology that performs the SD-WAN Application Intelligence Engine (SAIE) flow on network traffic to identify network applications according to their traffic characteristics.

Note In Cisco vManage Release 20.7.1 and earlier releases, the SAIE flow is called the deep packet inspection (DPI) flow.

The specific traffic characteristics of a network application are called an application signatures. Cisco packages the signature for an application, together with other information, as a protocol. Cisco packages a large set of protocols, covering numerous commonly occurring network applications, as a Protocol Pack. Cisco updates and distributes Protocol Packs regularly. They provide a database of network application signatures for NBAR to use to identify network application traffic.

The term network applications is defined broadly, and may include all of the following, and more:

- Social media websites
- Voice over IP (VoIP) applications
- Streaming audio and video, such as Cisco Webex
- Cloud applications, such as for cloud storage
- SaaS applications
- Custom network applications specific to an organization

Identifying applications is useful for monitoring network traffic, configuring application-aware traffic policy, and more.

To summarize network application signatures, protocols, and Protocol Packs, and how NBAR uses them:

- The traffic of a network application has unique characteristics that can be used to identify the traffic as belonging to that specific application. These characteristics are called application signatures.
- Cisco packages the signature for a specific network application as a protocol.
- Cisco packages a large set of protocols, covering commonly occurring internet applications, as Protocol Packs.
- Cisco NBAR performs the SAIE flow on traffic to gather the information required to identify the sources of the traffic, and uses protocols, such as those provided in Protocol Packs, to match that information to specific network applications. The result is that NBAR identifies the network applications producing traffic in the network.

Cisco Software-Defined Application Visibility and Control (SD-AVC) uses Cisco NBAR application identification to provide information about application usage within a network.

Custom Applications

In addition to the standard protocols provided in a Protocol Pack, you can define protocols, called custom applications, to identify internet traffic, often for uncommon network applications that are of specific interest to their organization. Custom applications augment the protocols provided in a Protocol Pack.

You can use custom applications in the same way as any other protocol when configuring:

- Cisco Catalyst SD-WAN policies
- Application Quality of Experience (AppQoE) policies, such as application-aware routing, TCP acceleration, and Quality of Service (QoS)

Note The following terms are used in the documentation of related technologies, and are equivalent: custom applications, custom protocols, user-defined applications

Custom Applications in Cisco Catalyst SD-WAN

Cisco Software-Defined AVC (SD-AVC) is a component of Cisco Application Visibility and Control (AVC). It functions as a centralized network service, operating with specific participating devices in a network. One function of Cisco SD-AVC, which is included as a component of Cisco Catalyst SD-WAN, is to create and manage custom applications. Cisco Catalyst SD-WAN uses this Cisco SD-AVC functionality, through SD-AVC REST APIs, to enable you to define custom applications within Cisco Catalyst SD-WAN.

As a Cisco Catalyst SD-WAN user, you can use Cisco SD-WAN Manager to define custom applications. Cisco SD-AVC then pushes the custom applications to devices in the network. The devices in the network use the custom applications and other application protocols to analyze traffic traversing the devices.

The process of defining a custom protocol includes choosing criteria to identify network traffic as coming from a specific network application. The criteria can include characteristics of hosts originating the traffic, such as server names, IP addresses, and so on.

Priority of Protocols and Custom Applications

It is possible to define custom applications that match some of the same traffic as a protocol included in the Protocol Pack operating with Cisco NBAR. When matching traffic, custom applications have priority over Protocol Pack protocols. Deploying SD-AVC within an existing network does not require any changes to the network topology.

Restrictions for Custom Applications

- Maximum number of custom applications: 1100
- Maximum number of L3/L4 rules: 20000
- Maximum number of server names: 50000
- For server names, maximum instances of wildcard followed by a period (.): 50000

  Example: *.cisco.com matches www.cisco.com, developer.cisco.com
- For server names, maximum instances of prefix wildcard as part of server name: 256

  Example: *ample.com matches www.example.com
- Mapping the same domain to two different custom applications is not supported.
- DNS traffic and application traffic need to be in the same VRF for SD-AVC to perform first packet classification.
- Creating custom applications though CLI is not supported in Cisco Catalyst SD-WAN policy.
- Activation of custom applications:
  - A custom application created in Cisco SD-WAN Manager is activated immediately for application visibility functionality only (monitoring traffic), such as for protocol-discovery counters and Flexible NetFlow (FNF). When activated for visibility functionality only, custom applications do not affect traffic policy.
  - When the custom application is used by a policy, it becomes activated for control functionality (traffic policy) also.

Configure Custom Applications Using Cisco SD-WAN Manager

Prerequisites

Install Cisco SD-AVC as a component of Cisco Catalyst SD-WAN. For information on how to enable SD-AVC on Cisco SD-WAN Manager, see Information on how to enable SD-AVC for Cisco SD-WAN devices.

Perform the following steps to configure custom applications:

1. In Cisco SD-WAN Manager, select **Configuration** > **Policies**.
2. Select **Centralized Policy**.
3. Click **Custom Options** and select **Centralized Policy** > **Lists**.
4. Click **Custom Applications**, and then click **New Custom Application**.
5. To define the application, provide an application name and enter match criteria. The match criteria can include one or more of the attributes provided: server names, IP addresses, and so on. You do not need to enter match criteria for all fields.

   The match logic follows these rules:

   - Between all L3/L4 attributes, there is a logical AND. Traffic must match all conditions.
   - Between L3/L4 and Server Names, there is a logical OR. Traffic must match either the server name or the L3/L4 attributes.

Field	Description
Application Name	(mandatory) Enter a name for the custom application. Maximum length: 32 characters
Server Names	One or more server names, separated by commas. You can include an asterisk wildcard match character (*) only at the beginning of the server name. Examples: *cisco.com, *.cisco.com (match www.cisco.com, developer.cisco.com, …)
L3/L4 Attributes	
IP Address	Enter one or more IPv4 addresses, separated by commas. Example: 10.0.1.1, 10.0.1.2 **Note** The subnet prefix range is 24 to 32.

Field	Description
Ports	Enter one or more ports or port ranges, separated by commas. Example: 30, 45-47
L4 Protocol	Select one of the following: TCP, UDP, TCP-UDP

6. Click **Add**. The new custom application appears in the table of custom applications.

Note To check the progress of creating the new custom application, click **Tasks** (clipboard icon). A panel opens, showing active and completed processes.

Example Custom Application Criteria

Criteria	How to configure fields
Domain name	**Server Names**: cisco.com
Set of IP addresses, set of ports, and L4 protocol	**IP Address**: 10.0.1.1, 10.0.1.2 **Ports**: 20, 25-37 **L4 Protocol**: TCP-UDP
Set of ports and L4 protocol	**Ports**: 30, 45-47 **L4 Protocol**: TCP

Verify Custom Applications

Verify Custom Applications in Cisco SD-WAN Manager

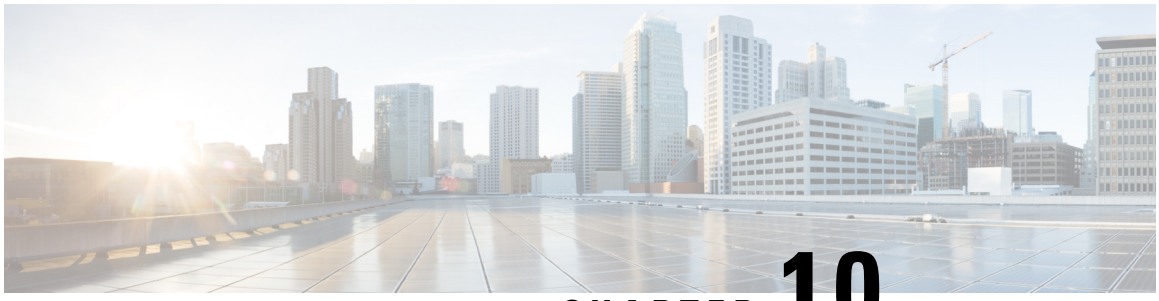
After you define a custom application, it appears in the **Custom Application List**, which shows all available protocols and custom applications. The **Custom Application List** is available here:

Configuration > **Policies** > **Centralized Policy** > **Add Policy** > **Custom Applications**.

Verify Protocols and Custom Applications on a Device

Use the **show app dpi supported-applications** command to display all protocols and custom applications that are loaded on the router. It is helpful to filter the results. For example, to display all protocols and custom applications with "custom" in the name, use this:

```
Device#show app dpi supported-applications | include custom
custom_amazon                    3899          PPDK LOCAL
custom_facebook                  3284          PPDK LOCAL
```



CHAPTER 10

Application-Aware Routing

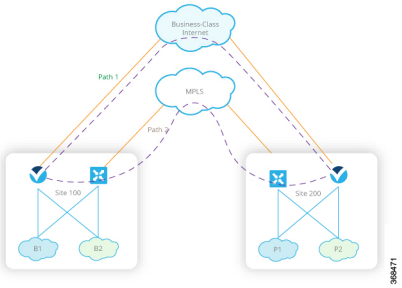
Note To achieve simplification and consistency, the Cisco SD-WAN solution has been rebranded as Cisco Catalyst SD-WAN. In addition, from Cisco IOS XE SD-WAN Release 17.12.1a and Cisco Catalyst SD-WAN Release 20.12.1, the following component changes are applicable: **Cisco vManage** to **Cisco Catalyst SD-WAN Manager**, **Cisco vAnalytics** to **Cisco Catalyst SD-WAN Analytics**, **Cisco vBond** to **Cisco Catalyst SD-WAN Validator**, **Cisco vSmart** to **Cisco Catalyst SD-WAN Controller**, and **Cisco Controllers** to **Cisco Catalyst SD-WAN Control Components**. See the latest Release Notes for a comprehensive list of all the component brand name changes. While we transition to the new names, some inconsistencies might be present in the documentation set because of a phased approach to the user interface updates of the software product.

Information About Application-Aware Routing

Application-aware routing tracks network and path characteristics of the data plane tunnels between Cisco SD-WAN devices and uses the collected information to compute optimal paths for data traffic. These characteristics include packet loss, latency, and jitter, and the load, cost and bandwidth of a link. The ability to consider factors in path selection other than those used by standard routing protocols—such as route prefixes, metrics, link-state information, and route removal on the Cisco SD-WAN device—offers a number of advantages to an enterprise:

- In normal network operation, the path taken by application data traffic through the network can be optimized, by directing it to WAN links that support the required levels of packet loss, latency, and jitter defined in an application's SLA.
- In the face of network brownouts or soft failures, performance degradation can be minimized. The tracking of network and path conditions by application-aware routing in real time can quickly reveal performance issues, and it automatically activates strategies that redirect data traffic to the best available path. As the network recovers from the soft failure conditions, application-aware routing automatically readjusts the data traffic paths.

- Network costs can be reduced because data traffic can be more efficiently load-balanced.
- Application performance can be increased without the need for WAN upgrades.



Each Cisco SD-WAN device supports up to eight TLOCs, allowing a single Cisco SD-WAN device to connect to up to eight different WAN networks. This capability allows path customization for application traffic that has different needs in terms of packet loss and latency.

Application-Aware Routing Support for Multicast Protocols

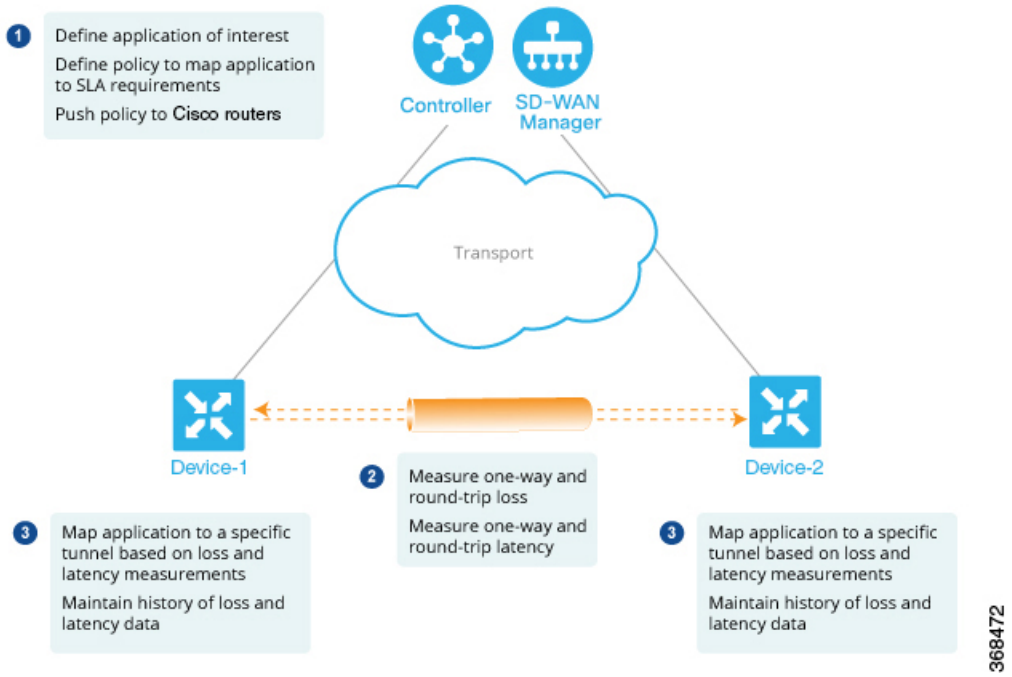
Starting from Cisco IOS XE Catalyst SD-WAN Release 17.3.1a, application-aware routing supports overlay multicast traffic on devices. In older releases, an application-route policy is supported only for unicast traffic.

The devices classify the multicast traffic based on the group address and sets the SLA class. The group address can be source IP, destination IP, source prefixes, and destination prefixes. In the forwarding plane, any traffic for group address must use only those TLOC paths that meet the SLA requirement. You can perform the path selection for a group based on the preferred color, backup color, or the default action.

Components of Application-Aware Routing

The Cisco SD-WAN Application-Aware Routing solution consists of three elements:

- **Identification**—You define the application of interest, and then you create a centralized data policy that maps the application to specific SLA requirements. You single out data traffic of interest by matching on the Layer 3 and Layer 4 headers in the packets, including source and destination prefixes and ports, protocol, and DSCP field. As with all centralized data policies, you configure them on a Cisco Catalyst SD-WAN Controller, which then passes them to the appropriate Cisco SD-WAN devices.
- **Monitoring and measuring**—The Cisco SD-WAN software uses BFD packets to continuously monitor the data traffic on the data plane tunnels between devices, and periodically measures the performance characteristics of the tunnel. To gauge performance, the Cisco SD-WAN device looks for traffic loss on the tunnel, and it measures latency by looking at the one-way and round-trip times of traffic traveling over the tunnel. These measurements might indicate suboptimal data traffic conditions.
- **Mapping application traffic to a specific transport tunnel**—The final step is to map an application's data traffic to the data plane tunnel that provides the desired performance for the application. The mapping decision is based on two criteria: the best-path criteria computed from measurements performed on the WAN connections and on the constraints specified in a policy specific to application-aware routing.



To create a data policy based on the Layer 7 application itself, configure the Cisco Catalyst SD-WAN Application Intelligence Engine (SAIE) flow with a centralized data policy. With the SAIE flow, you can direct traffic to a specific tunnel, based on the remote TLOC, the remote TLOC, or both. You cannot direct traffic to tunnels based on SLA classes.

Note In Cisco vManage Release 20.7.1 and earlier releases, the SAIE flow is called the deep packet inspection (DPI) flow.

SLA Classes

Table 25: Feature History

Feature	Release Information	Description
Support for SLA Classes	Cisco SD-WAN Release 20.1.1	This feature allows you to configure up to a maximum of eight SLA classes on Cisco SD-WAN Controller. Using this feature, you can configure additional options in an application-aware routing policy.

Feature	Release Information	Description
Support for six SLA Classes per Policy	Cisco vManage Release 20.3.1 Cisco SD-WAN Release 20.3.1	This feature allows you to configure up to four SLA classes per policy on Cisco SD-WAN devices. This enhancement allows additional options in an application-aware routing policy.
SLA Class Support Enhancement	Cisco SD-WAN Release 20.6.1 Cisco vManage Release 20.6.1	This feature is an enhancement to support more than six SLA classes per policy on Cisco SD-WAN devices.
Application Aware Routing and Data Policy SLA Preferred Colors	Cisco SD-WAN Release 20.6.1 Cisco vManage Release 20.6.1	This feature provides different behaviors to choose preferred colors based on the SLA requirements when both application-aware routing policy and data policies are configured.

A service-level agreement (SLA) determines actions taken in application-aware routing. The SLA class defines the maximum jitter, maximum latency, maximum packet loss, or a combination of these values for data plane tunnels in Cisco SD-WAN devices. Each data plane tunnel comprises a local transport locators (TLOC) and a remote TLOC pair. You can configure SLA classes under the **policy sla-class** command hierarchy on Cisco SD-WAN Controllers. From Cisco SD-WAN Release 20.1.x, you can configure a maximum of eight SLA classes on Cisco SD-WAN Validator. However, you can define only four unique SLA classes in an application-aware route policy. In releases earlier than Cisco SD-WAN Release 20.1.x, you can configure a maximum of four SLA classes.

Note In Cisco SD-WAN Release 20.3.1, you cannot configure more than four different SLA classes on Cisco SD-WAN devices. If you configure more than four different SLA classes, the application-aware routing policy gets rejected.

You can configure the following parameters in an SLA class.

Table 26: SLA Components

Description	Command	Value or Range
Maximum acceptable packet jitter on the data plane tunnel	**jitter** *milliseconds*	1–1000 milliseconds
Maximum acceptable packet latency on the data plane tunnel.	**latency** *milliseconds*	1–1000 milliseconds
Maximum acceptable packet loss on the data plane tunnel.	**loss** *percentage*	1–100 percent

SLA Support Enhancement

From Cisco SD-WAN Release 20.6.1 and Cisco vManage Release 20.6.1, you can configure more than six SLA classes per policy on Cisco SD-WAN devices.

vEdge Cloud Router platforms require another 250-MB RAM to support 7+1 SLA classes. vEdge Cloud Routers can be small or medium depending on the RAM size. Small vEdge Cloud Routers have RAM size less than 1.75 GB and the medium vEdge Cloud Routers have RAM size from 1.75 GB to 2.5 GB. (For example, when you upgrade to Cisco SD-WAN Release 20.6.1, vEdge Cloud Router with RAM size 1.5 GB is a small device.

This feature enhancement increases the number of SLA classes supported on Cisco SD-WAN Controller and SD-WAN Edge devices. With the increase in the SLA class support, you can align SLA classes to IP Virtual Private Networks (IP-VPN) on Multi-Protocol Label Switching (MPLS) networks for transporting traffic to a global network.

The SLA enhancement helps in multitenancy, where you can push different SLA classes for different tenants. The multitenancy feature requires the Cisco SD-WAN Controller to support more than eight SLA classes. To allocate SLA classes to different tenants, the global limit for policies must be 64.

Note You cannot configure the default SLA. The default SLA is configured in all the devices to forward traffic when no user-defined SLA is met.

Table 27: Maximum SLA Classes Supported on Cisco SD-WAN Devices

Platform	User-configurable SLA Classes Prior to Cisco SD-WAN Release 20.6.1 (+1 Default SLA Class)	User-configurable SLA Classes from Cisco SD-WAN Release 20.6.1 (+1 Default SLA Class)
Cisco SD-WAN Controller	8	64
Cisco vEdge 5000	4	15
vEdge Cloud Router	4	7
Any other Cisco vEdge devices	4	4

SLA-Preferred Colors

From Cisco SD-WAN Release 20.6.1, when you configure both application-aware routing policy and data policy, and if data flow matches the app-route and data policy sequences, the following expected behaviors occur:

- If the preferred colors that you configure in application-aware routing meet the SLA requirements, and these preferred colors have some colors that are common with data policy, the common preferred colors are chosen over others for forwarding. (Prior to Cisco SD-WAN Release 20.6.1, the data policy-preferred colors were forwarded and the application-aware routing policy preferences were ignored.)
- If preferred colors in application-aware routing do not meet the SLA, but there are colors that are common with the data policy, and these colors meet the SLA in application-aware routing, then these colors take precedence and are chosen for forwarding.

- If no tunnels or colors meet the SLA in application-aware routing, the data policy takes precedence and is chosen for forwarding. If the data policy has preferred colors, these colors are chosen. Otherwise, load balance occurs across all the colors in the data policy.

Classification of Tunnels into SLA Classes

The process of classifying tunnels into one or more SLA classes for application-aware routing has three parts:

- Measure loss, latency, and jitter information for the tunnel.
- Calculate the average loss, latency, and jitter for the tunnel.
- Determine the SLA classification of the tunnel.

Measure Loss, Latency, and Jitter

When a data plane tunnel in the overlay network is established, a BFD session automatically starts on the tunnel. In the overlay network, each tunnel is identified with a color that identifies a specific link between a local TLOC and a remote TLOC. The BFD session monitors the liveness of the tunnel by periodically sending Hello packets to detect whether the link is operational. Application-aware routing uses the BFD Hello packets to measure the loss, latency, and jitter on the links.

By default, the BFD Hello packet interval is 1 second. This interval is user-configurable (with the **bfd color interval** command). Note that the BFD Hello packet interval is configurable per tunnel.

Calculate Average Loss, Latency, and Jitter

BFD periodically polls all the tunnels on the Cisco SD-WAN devices to collect packet latency, loss, jitter, and other statistics for use by application-aware routing. At each poll interval, application-aware routing calculates the average loss, latency, and jitter for each tunnel, and then calculates or recalculates each tunnel's SLA. Each poll interval is also called a "bucket."

By default, the poll interval is 10 minutes. With the default BFD Hello packet interval at 1 second, this means that information from about 600 BFD Hello packets is used in one poll interval to calculate loss, latency, and jitter for the tunnel. The poll interval is user-configurable (with the **bfd app-route poll-interval** command). Note that the application-aware routing poll interval is configurable per Cisco SD-WAN device; that is, it applies to all tunnels originating on a device.

Reducing the poll interval without reducing the BFD Hello packet interval may affect the quality of the loss, latency, and jitter calculation. For example, setting the poll interval to 10 seconds when the BFD Hello packet interval is 1 second means that only 10 Hello packets are used to calculate the loss, latency, and jitter for the tunnel.

The loss, latency, and jitter information from each poll interval is preserved for six poll intervals. At the seventh poll interval, the information from the earliest polling interval is discarded to make way for the latest information. In this way, application-aware routing maintains a sliding window of tunnel loss, latency, and jitter information.

The number of poll intervals (6) is not user-configurable. Each poll interval is identified by an index number (0 through 5) in the output of the **show app-route statistics** command.

Determine SLA Classification

To determine the SLA classification of a tunnel, application-aware routing uses the loss, latency, and jitter information from the latest poll intervals. The number of poll intervals used is determined by a multiplier. By default, the multiplier is 6, so the information from all the poll intervals (specifically, from the last six poll intervals) is used to determine the classification. For the default poll interval of 10 minutes and the default multiplier of 6, the loss, latency, and jitter information collected over the last hour is considered when classifying the SLA of each tunnel. These default values have to be chosen to provide damping of sorts, as a way to prevent frequent reclassification (flapping) of the tunnel.

The multiplier is user-configurable (with the **bfd app-route multiplier** command). Note that the application-aware routing multiplier is configurable per Cisco SD-WAN device; that is, it applies to all tunnels originating on a device.

If there is a need to react quickly to changes in tunnel characteristics, you can reduce the multiplier all the way down to 1. With a multiplier of 1, only the latest poll interval loss and latency values are used to determine whether this tunnel can satisfy one or more SLA criteria.

Based on the measurement and calculation of tunnel loss and latency, each tunnel may satisfy one or more user-configured SLA classes. For example, a tunnel with a mean loss of 0 packets and mean latency of 10 milliseconds would satisfy a class that has been defined with a maximum packet loss of 5 and a minimum latency of 20 milliseconds, and it would also satisfy a class that has been defined with a maximum packet loss of 0 and minimum latency of 15 milliseconds.

Regardless of how quickly a tunnel is reclassified, the loss, latency, and jitter information is measured and calculated continuously. You can configure how quickly application-aware routing reacts to changes by modifying the poll interval and multiplier.

Per-Class Application-Aware Routing

Table 28: Feature History

Feature Name	Release Information	Description
Per-Class Application-Aware Routing	Cisco SD-WAN Release 20.4.1 Cisco vManage Release 20.4.1	This feature enhances the capabilities of directing traffic to next-hop addresses based on the service level agreement (SLA) definitions. These SLA definitions along with the policy to match and classify traffic types can be used to direct traffic over specific Cisco Catalyst SD-WAN tunnels. The SLA definition comprises of values of loss, latency, and jitter, which are measured using the Bidirectional Forwarding Detection (BFD) channel that exists between two transport locators (TLOCs).

Per-Class Application-Aware Routing Overview

The SLA definition comprises of values of loss, latency, and jitter, which are measured using the BFD channel that exists between two TLOCs. These values collectively represent the status of the network and the BFD link. The BFD control messages are sent with a high priority Differentiated Services Code Point (DSCP) marking of 48.

The SLA metrics based on the high priority packet does not reflect the priority that is received by the actual data that flows through the edge device. The data, depending on the application class, can have different DSCP values in the network. Therefore, a more accurate representation of the loss, latency, and jitter for the traffic profiles is required for the networks to use such measurements to direct traffic types to the right tunnels.

Application-aware routing uses policies that constrain paths that can be used for forwarding the application. These constraints are usually expressed in terms of SLA classes that contain loss, latency, and jitter requirements that must be met. This requires that these metrics be measured on all the paths to the destination of the traffic using active probing or by passive monitoring.

Active probing methods include generation of synthetic traffic that is injected along with real traffic. The expectation is that the probes and the real traffic is forwarded in the same way. BFD probing, ICMP, periodic HTTP requests and IP SLA measurements are some examples of active probing mechanisms. The Cisco Catalyst SD-WAN solution uses BFD based probes for active measurements. Passive monitoring methods rely on the Cisco Catalyst SD-WAN Application Intelligence Engine (SAIE) flow and monitoring actual traffic. For example, RTP/TCP traffic is monitored for loss, latency, and jitter.

Note In Cisco vManage Release 20.7.1 and earlier releases, the SAIE flow is called the deep packet inspection (DPI) flow.

Application Probe Class

An application probe class (app-probe-class) comprises of a forwarding class, color, and DSCP. This defines the marking per color of applications that are forwarded. The color or DSCP mapping is local to a Cisco SD-WAN network site. However, a few colors and the DSCP mapping for a color does not change per site. The forwarding class determines the QoS queue in which the BFD echo request is queued at the egress tunnel port. This is applicable only for BFD echo request packets. The packet-loss-priority for BFD packets is fixed to low. When BFD packets are sent with SLA class, they use the same DSCP value. When BFD packets are sent with app-probe-class along with SLA class, the BFD packets are sent for each SLA app-probe-class separately in a round-robin manner.

Default DSCP Values

The default DSCP value that is used in the DSCP control traffic is 48. However, there is a provision to change the default value along with the option to configure on the edge devices. All the network service providers may not necessarily use DSCP 48.

The BFD packet having the default DSCP can also be used for other features such as PMTU. A change in the default DSCP means that the other features are affected by the new default DSCP value. Therefore, we recommend that you configure the highest priority DSCP marking that the service provider provides (usually 48, but can be different based on the SLA agreement of the service provider). The color level overrides the global level default DSCP marking.

Configure Application-Aware Routing

Table 29: Feature History

Feature Name	Release Information	Description
Application-Aware Routing for Hub and Spoke	Cisco SD-WAN Release 20.9.1 Cisco vManage Release 20.9.1	This feature enables you to configure an application-aware routing policy on a hub device and apply the policy to packets that the hub device receives from a spoke device and sends to another spoke device. Applying the application-aware routing policy helps you realize the required SLA for spoke-hub-spoke traffic.

This topic provides general procedures for configuring application-aware routing. Application-aware routing policy affects only traffic that is flowing from the service side (the local/WAN side) to the tunnel (WAN) side of the Cisco SD-WAN device.

An application-aware routing policy matches applications with an SLA, that is, with the data plane tunnel performance characteristics that are necessary to transmit the applications' data traffic. The primary purpose of application-aware routing policy is to optimize the path for data traffic being transmitted by Cisco SD-WAN devices.

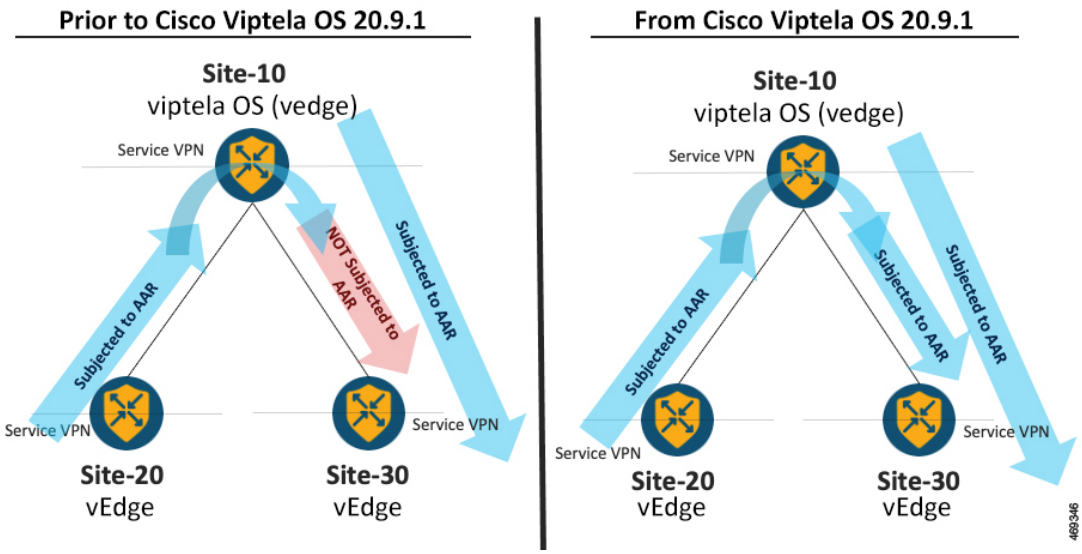
An application-aware routing policy is a type of centralized data policy: you configure it on the Cisco SD-WAN Controller, and the controller automatically pushes it to the affected Cisco SD-WAN devices. As with any policy, an application-aware routing policy consists of a series of numbered (ordered) sequences of match-action pairs that are evaluated in order, from lowest sequence number to highest sequence number. When a data packet matches one of the match conditions, an SLA action is applied to the packet to determine the data plane tunnel to use to transmit the packet. If a packet matches no parameters in any of the policy sequences, and if no SLA class is configured for the default-action, the packet is accepted and forwarded with no consideration of SLA. Because application-aware routing policy accepts nonmatching traffic by default, it is considered as a positive policy. Other types of policies in the Cisco SD-WAN software are negative policies, because by default they drop nonmatching traffic.

Hub and Spoke Topology

Minimum releases: Cisco SD-WAN Release 20.9.1 and Cisco vManage Release 20.9.1

You can configure an application-aware routing policy on a hub device and apply the policy to packets that the hub device receives from a spoke device and sends to another spoke device.

Figure 12: Hub and Spoke Topology



Configure Application-Aware Routing Policies Using Cisco SD-WAN Manager

To configure application-aware routing policy, use the Cisco SD-WAN Manager policy configuration wizard. For Centralized Policy configuration details, see Configure Centralized Policies. The wizard consists of four sequential windows that guide you through the process of creating and editing policy components:

- Create Applications or Groups of Interest: Create lists that group together related items and that you call in the match or action components of a policy. For configuration details, see Configure Groups of Interest.
- Configure Topology: Create the network structure to which the policy applies. For topology configuration details, see Configure Topology and VPN Membership.
- Configure Traffic Rules: Create the match and action conditions of a policy.
- Apply Policies to Sites and VPNs: Associate policy with sites and VPNs in the overlay network.

In the first three policy configuration wizard windows, you are creating policy components or blocks. In the last window, you are applying policy blocks to sites and VPNs in the overlay network.

For an application-aware routing policy to take effect, you must activate the policy.

Configure Best Tunnel Path

Table 30: Feature History

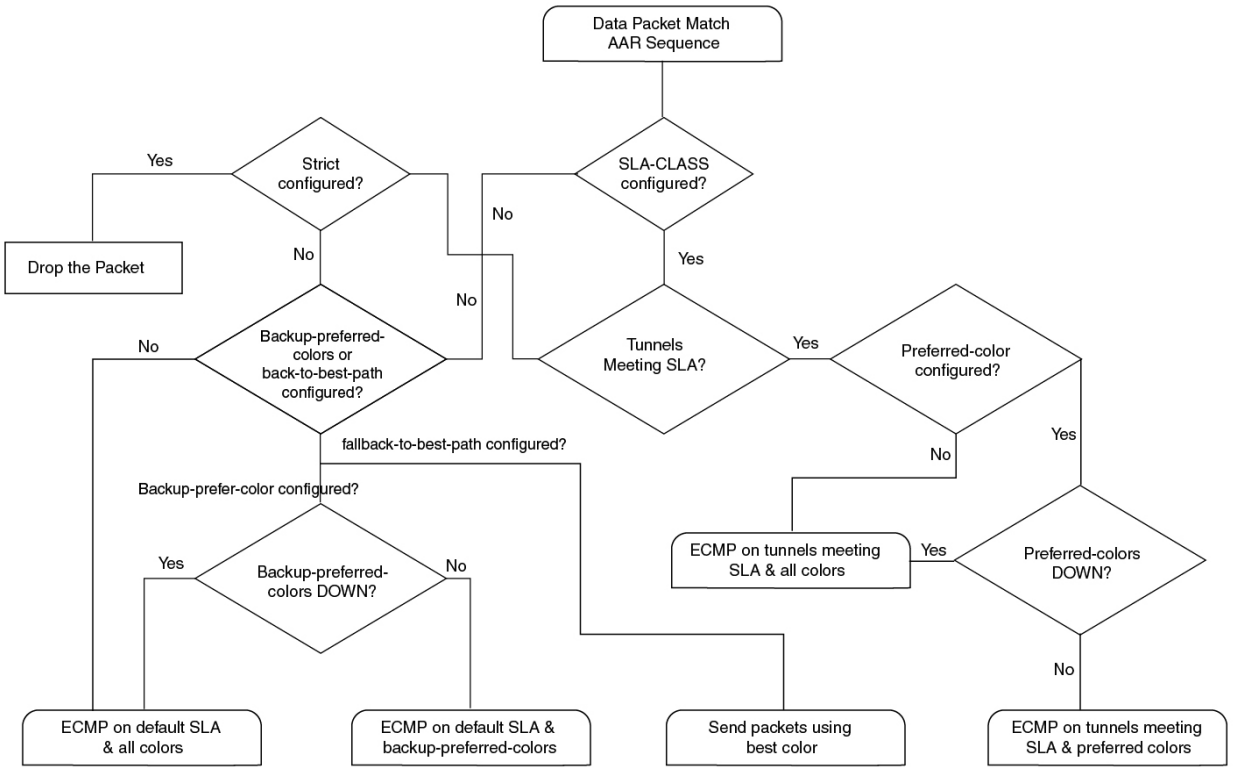
Feature Name	Release Information	Description
Best of the Worst (BOW) Tunnel Selection	Cisco vManage Release 20.5.1 Cisco SD-WAN Release 20.5.1	This feature introduces a new policy action **fallback-to-best-path** to pick the best path or color out of the available colors. When the data traffic does not meet any of the SLA class requirements, this feature allows you to select the best tunnel path criteria sequence using the **Fallback Best Tunnel** option under each SLA class to avoid packet loss.

Best Tunnel Path Overview

To avoid data packet loss and to configure the best application-aware routing tunnel selection when a SLA is not met, you can configure the following policy actions:

- **backup-preferred-color**
- **fallback-to-best-path**

Figure 13: Flow Chart for Application-Aware Routing Tunnel Selection



Recommendation for the Best Tunnel Path

- Configure the **fallback-to-best-path** policy action in Cisco SD-WAN Manager when configuring a SLA class.
- Configure the **backup-preferred-color** policy action in Cisco SD-WAN Manager when configuring traffic rules.

Configure Variance for Best Tunnel Path

Cisco SD-WAN Manager uses best of worst (BOW) to find a best tunnel when no tunnel meets any of the SLA class requirements.

Assume that the required latency is 100 ms to meet the SLA class requirements and tunnel T1 has 110 ms. Tunnel T2 has 111 ms and tunnel T3 has 112 ms.

As per the BOW logic, the best tunnel is T1. T2 and T3 are equally the best tunnels, with only a difference of a few ms.

You configure variance in Cisco SD-WAN Manager when configuring an SLA class. Variance accommodates small deviations as part of the best tunnel selection.

For more information, see Configure SLA Class.

Example: Without Variance Configured

At time t1: T1 has 100 ms, T2 has 101 ms, and T3 has 102 ms

At time t2: T1 has 101 ms, T2 has 100 ms, and T3 has 102 ms

At time t3: T1 has 101 ms, T2 has 112 ms, and T3 has 100 ms

At time t1, the best tunnel changes from T1 to T2, and for time t2, the best tunnel changes from T2 to T3. Because variance is not configured, this leads to data path reprogramming and changes to the data traffic paths.

Assume instead that you configure variance to dampen a small deviation in ms.

For example, you configure variance as 5 ms, which means that the best tunnel SLA = 100 ms. The range is from 100 ms to 105 ms.

Example: With Variance Configured

BOW(t1) = {T1, T2, T3}

BOW(t2) = {T1, T2, T3}

BOW(t3) = {T1, T2, T3}

With variance configured, there is no data path reprogramming required or changes to data traffic paths.

Verify Configuration of Variance for Best Tunnel Path

Example for Latency Variance

```
Device# show sdwan policy from-vsmart
from-vsmart sla-class video
latency                      100
jitter                       150
```

```
 fallback-best-tunnel         latency

Tunnel T1: Latency: 110 msec, Loss: 0%, Jitter: 200 msec
Tunnel T2: Latency: 115 msec, Loss: 0%, Jitter: 200 msec
Tunnel T3: Latency: 120 msec, Loss: 0%, Jitter: 200 msec
```

Without latency variance, the best tunnel is T1.

With latency variance configured as 10 ms, T1, T2, and T3 are the best tunnels.

The range is from 110 ms to 120 ms.

The best latency + variance is 110 ms + 10 ms.

Use the following formula to find the best tunnel selection for latency variance:

(best_latency, best_latency + latency_variance)

Example for Jitter Variance

```
Device# show sdwan policy from-vsmart
from-vsmart sla-class video
latency                      100
jitter                       150
 fallback-best-tunnel        jitter

Tunnel T1: Latency: 90 msec, Loss: 0%, Jitter: 160 msec
Tunnel T2: Latency: 80 msec, Loss: 0%, Jitter: 200 msec
Tunnel T3: Latency: 70 msec, Loss: 0%, Jitter: 152 msec
```

Without jitter variance, the best tunnel is T3.

With jitter variance configured as 10 ms, T1 and T3 are the best tunnels.

The range is from 152 ms to 162 ms.

The best jitter + variance is 152 ms + 10 ms.

Use the following formula to find the best tunnel selection for jitter variance:

(best_jitter, best_jitter + jitter_variance)

Example for Loss Variance

```
Device# show sdwan policy from-vsmart
from-vsmart sla-class video
latency                      100
jitter                       1
 fallback-best-tunnel        loss

Tunnel T1: Latency: 110 msec, Loss: 2%, Jitter: 200 msec
Tunnel T2: Latency: 115 msec, Loss: 3%, Jitter: 200 msec
Tunnel T3: Latency: 120 msec, Loss: 4%, Jitter: 200 msec
```

Without loss variance, the best tunnel is T1.

With loss variance configured as 1%, T1 and T2 are the best tunnels.

The range is from 2% to 3%.

The best loss + variance is 2%.

Use the following formula to find the best tunnel selection for loss variance:

(best_loss, best_loss + loss_variance)

Configure SLA Class

1. From the Cisco SD-WAN Manager menu, select **Configuration** > **Policies**. Centralized Policy is selected and displayed by default.
2. Click **Add Policy**.
3. In the create groups of interest page, from the left pane, click **SLA Class**, and then click **New SLA Class List**.
4. In the **SLA Class List Name** field, enter a name for SLA class list.
5. Define the SLA class parameters:
   a. In the **Loss** field, enter the maximum packet loss on the connection, a value from 0 through 100 percent.
   b. In the **Latency** field, enter the maximum packet latency on the connection, a value from 1 through 1,000 milliseconds.
   c. In the **Jitter** field, enter the maximum jitter on the connection, a value from 1 through 1,000 milliseconds.
   d. Choose the required app probe class from the **App Probe Class** drop-down list.
6. (Optional) Check the **Fallback Best Tunnel** check box to enable the best tunnel criteria.

   This optional field is available from Cisco SD-WAN Release 20.5.1 to pick the best path or color from the available colors when a SLA is not met. When this option is selected, you can choose the required criteria from the drop-down. The criteria are a combination of one or more of loss, latency, and jitter values.
7. Select the **Criteria** from the drop-down. The available criteria are:
   - None
   - Latency
   - Loss
   - Jitter
   - Latency, Loss
   - Latency, Jitter
   - Loss, Latency
   - Loss, Jitter
   - Jitter, Latency
   - Jitter, Loss
   - Latency, Loss, Jitter
   - Latency, Jitter, Loss
   - Loss, Latency, Jitter

- Loss, Jitter, Latency
- Jitter, Latency, Loss
- Jitter, Loss, Latency

8. (Optional) Enter the **Loss Variance (%)**, **Latency Variance (ms)**, and the **Jitter Variance (ms)** for the selected criteria.

   For more information, see Configure Variance for Best Tunnel Path.

9. Click **Add**.

Configure Traffic Rules

To configure an application-aware routing policy:

1. Click **Application Aware Routing**.
2. From the **Add Policy** drop-down list, choose **Create New**.
3. Click **Sequence Type**. A policy sequence containing the text string **App Route** is added in the left pane.
4. Double-click the **App Route** text string and enter a name for the policy sequence. You can copy, delete, or rename a policy sequence. The name you enter is displayed both in the **Sequence Type** list in the left pane and in the right pane.
5. In the right pane, click **Sequence Rule**. The **Match/Actions** dialog box opens, and **Match** is selected by default. The available policy match conditions are listed below the dialog box.
6. In the **Protocol** drop-down list, choose one of the following option:
   - **IPv4**
   - **IPv6**
   - **Both**

Note Depending on which protocol that you choose, the **Match** or **Actions** conditions may be different.

7. Click and choose one or more **Match** conditions. Set the values as described in the following table:

Table 31: Match Conditions

Match Condition	Procedure
None (match all packets)	Do not specify any match conditions.
Application/Application Family List	Click **Application/Application Family List** and choose an application list.

Cloud Saas Application List	Cisco SD-WAN Manager provides a list of several cloud applications that Cisco Catalyst SD-WAN Cloud OnRamp for SaaS can use to determine the best path selection for each SaaS application. For more information on Cisco Catalyst SD-WAN Cloud OnRamp for SaaS, see the *Cisco Catalyst SD-WAN Cloud OnRamp Configuration Guide, Cisco IOS XE Release 17.x*. **Note** **Cloud Saas Application List** displays as a match condition if you specify **IPv4** as the **Protocol** option. In the drop-down list, choose a SaaS application from the drop-down list.
DNS Application List	In the drop-down list, select an application family.
Destination Data Prefix	To match a list of destination prefixes, choose the list from the drop-down list. To match an individual destination prefix, type the prefix in the **Destination** dialog box.
Destination Region	You can use **Destination Region** in a Cisco Catalyst SD-WAN network using Cisco Catalyst SD-WAN Multi-Region Fabric. Choose one of the following options from the drop-down list: • **Primary**: Match traffic if the destination site is in the same primary region (also called access region) as the source. • **Secondary**: Match traffic if the destination site is not in the same primary region as the source but is within the same secondary region as the source. This traffic can reach the destination using a direct tunnel, as described for secondary regions. • **Other**: Match traffic if the destination site is not in the same primary region or secondary region as the source. This traffic requires a multi-hop path from the source to the destination. For more information on how to configure Multi-Region Fabric, see the *Cisco Catalyst SD-WAN Multi-Region Fabric (also Hierarchical SD-WAN) Configuration Guide.*
Destination Port	Enter the port number. Specify a single port number, a list of port numbers (with numbers separated by a space), or a range of port numbers (with the two numbers separated with a hyphen [-]).
Traffic To	When creating a data policy or an application-aware policy for a border router for Multi-Region Fabric, you can use match criteria to match traffic flowing to the access region, the core region, or a service VPN.
DNS (to enable split DNS)	In the drop-down list, choose **Request** to process DNS requests for the DNS applications, and choose **Response** to process DNS responses for the applications.
DSCP	Type the DSCP value, a number from 0 through 63.

PLP	Choose **Low** or **High**. To set the PLP to **High**, apply a policer that includes the **exceed remark** option.
Protocol	Type the internet protocol number, a number from 0 through 255.
ICMP Message	For **Protocol (IPv4)**, when you select a value as 1 in the **Protocol** field in the Match Conditions section, the **ICMP Message** field displays where you can select an ICMP message to apply to the data policy. For **Protocol (IPv6)**, when you select a value as 58 in the **Protocol** field in the **Match Conditions** section, the **ICMP Message** field displays where you can select an ICMP message to apply to the data policy. **Note** This field is available from Cisco IOS XE Release 17.4.1 or Cisco SD-WAN Release 20.4.1, and also Cisco vManage Release 20.4.1. When **Protocol** is selected as **Both**, the **ICMP Message or ICMPv6 Message** field displays.
Source Data Prefix	To match a list of source prefixes, choose the list from the drop-down list. To match an individual source prefix, enter the prefix in the **Source** field.
Source Port	Enter the port number. Specify a single port number, a list of port numbers (with numbers separated by a space), or a range of port numbers (with the two numbers separated with a hyphen [-]).

8. To select actions for the matched data traffic, click **Actions**. Set the values as described in the following table:

Table 32: Actions

Action	Procedure
Backup SLA Preferred Color	Set the policy action for a **Backup SLA Preferred Color** match condition. When no tunnel matches the SLA, direct the data traffic to a specific tunnel. Data traffic is sent out the configured tunnel if that tunnel interface is available. If that tunnel interface is not available, traffic is sent out to another available tunnel. You can specify one or more colors. The backup SLA preferred color is a loose matching condition, not a strict matching condition. Click **Backup SLA Preferred Color**. In the drop-down list, choose one or more colors.
Counter	Set the policy action for a **Counter** match condition. Click **Counter**. In the **Counter Name** field, enter the name of the file in which to store packet counters.
Log	You can place a sampled set of packets that match the SLA class rule into system logging **(syslog)** files. In addition to logging the packet headers, a **syslog** message is generated the first time a packet header is logged and then every five minutes thereafter, as long as the flow is active. Click **Log** to enable logging.

Action	Procedure

SLA Class List	Set the policy action for an **SLA Class List** match condition. For the SLA class, all matching data traffic is directed to a tunnel whose performance matches the SLA parameters defined in the class. The device first tries to send the traffic through a tunnel that matches the SLA. If a single tunnel matches the SLA, data traffic is sent through that tunnel. If two or more tunnels match, traffic is distributed among them. If no tunnel matches the SLA, data traffic is sent through one of the available tunnels.
	
	Click **SLA Class List**.
	
	In the **SLA Class** drop-down list, choose one or more SLA classes.
	
	Optionally, in the **Preferred Color** drop-down list, choose the color of the data plane tunnel or tunnels to prefer. Traffic is load-balanced across all the tunnels. If no tunnels match the SLA, data traffic is sent through any available tunnel. That is, color preference is a loose matching, not a strict matching.
	
	Optionally, when the **Preferred Color** is not selected, you can choose the preferered color group from the **Preferred Color Group** drop-down list. Select the preferred color group of the data plane tunnel or tunnels to prefer. You can configure up to three levels of priority based on the color or path preference. This field is available from Cisco IOS XE Catalyst SD-WAN Release 17.9.1a and Cisco vManage Release 20.9.1.
	
	Click **Strict/Drop** to perform strict matching of the SLA class. If no data plane tunnel is available that satisfies the SLA criteria, traffic is dropped.
	
	Click **Fallback to best path** to select the best available tunnel to avoid a packet drop.
	
	Note The **Fallback to best path** option is available from Cisco IOS XE Catalyst SD-WAN Release 17.5.1a and Cisco SD-WAN Release 20.5.1.
	
	You can select the **Fallback to best path** action only when the **Fallback Best Tunnel** option is enabled while defining a SLA class. If the **Fallback Best Tunnel** option is not enabled, then the following error message displays in Cisco SD-WAN Manager:
	
	`SLA Class selected, does not have Fallback Best Tunnel enabled.`
	`Please change the SLA class or change to Strict/Drop.`
	
	Click **Load Balance** to load balance traffic across all the tunnels.

Cloud SLA	**Cloud SLA** enables traffic to use the best path selection with Cisco Catalyst SD-WAN Cloud OnRamp for SaaS. Click **Cloud SLA**.

9. Click **Save Match and Actions**.
10. Create additional sequence rules as desired. Drag and drop to re-arrange them.
11. Click **Save Application Aware Routing Policy**.
12. Click **Next** to move to **Apply Policies to Sites and VPNs** in the wizard.

Default Action of Application-Aware Routing Policy

The default action of the policy defines how to handle packets that match none of the match conditions. For application-aware routing policy, if you do not configure a default action, all data packets are accepted and transmitted based on normal routing decisions, with no consideration of SLA.

To modify this behavior, include the **default-action sla-class** *sla-class-name* command in the policy, specifying the name of an SLA class you defined in the **policy sla-class** command.

When you apply an SLA class in a policy's default action, you cannot specify the **strict** option.

If no data plane tunnel satisfies the SLA class in the default action, the Cisco SD-WAN device selects one of the available tunnels by performing load-balancing across equal paths.

Expected behavior when data flow matches both AAR and data policies:

1. When data policy local TLOC action is configured, the **App-route preferred-color** and **backup-preferred-color** actions are ignored.
2. The **sla-class** and **sla-strict** actions are retained from the application routing configuration.
3. The data policy TLOC takes precedence.

When there is a **local-tloc-list** action that has multiple options, choose the local-TLOC that meets SLA.

- If no **local-tloc** meets SLA, then choose equal-cost multi-path routing (ECMP) for the traffic over the **local-tloc-list**.
- If none of the **local-tloc** is up, then choose a TLOC that is up.
- If none of the **local-tloc** is up and the DP is configured in restrict mode, then drop the traffic.

Configure Application Probe Class through Cisco Catalyst SD-WAN Manager

1. From the Cisco SD-WAN Manager menu, choose **Configuration** > **Policies**.
2. In **Centralized Policy**, click **Add Policy**. The **Create Groups of Interest** page appears.
3. Choose the list type **App Probe Class** from the left navigation panel to create your groups of interest.
4. Click **New App Probe Class**.
5. Enter the probe class name in the **Probe Class Name** field.
6. Choose the required forwarding class from the **Forwarding Class** drop-down list.

   If there are no forwarding classes, then create a class from the **Class Map** list page under the **Localized Policy Lists** in the **Custom Options** menu.

   To create a forwarding class:

   a. In the **Custom Options** drop-down, choose **Lists** from the Localized Policy options.
   b. In the Define Lists window, choose the list type **Class Map** from the left navigation panel.
   c. Click **New Class List** to create a new list.
   d. Enter **Class** and choose the **Queue** from the drop-down list.
   e. Click **Save**.

7. In the **Entries** pane, choose the appropriate color from the **Color** drop-down list and enter the **DSCP** value.

   Click + sign, to add more entries as required.

8. Click **Save**.

Add App-Probe-Class to an SLA Class

1. From the left pane, select **SLA Class**.
2. Click **New SLA Class List**.
3. In the **SLA Class List Name** field, enter a name for SLA class list.
4. Enter the required **Loss (%)**, **Latency (ms)**, and **Jitter (ms)**.
5. Choose the required app probe class from the **App Probe Class** drop-down list.
6. Click **Add**.

   The new SLA Class created with loss, latency, jitter, and app probe class is added to the table.

Configure Default DSCP on Cisco BFD Template

1. From the Cisco SD-WAN Manager menu, select **Configuration** > **Templates**.
2. Click **Feature Templates**.

Note In Cisco vManage Release 20.7.x and earlier releases, **Feature Templates** is titled **Feature**.

3. Click **Add Template**.
4. Select a device from the device list in the left pane.
5. In the right pane, select the BFD template listed under Basic Information.
6. Enter **Template Name** and **Description** in the respective fields.
7. In the **Basic Configuration** pane, enter **Multiplier** and **Poll Interval (milliseconds)**.
8. In the **Default DSCP value for BFD Packets** field, enter the required device specific value or choose the default value for DSCP.
9. (Optional) In the **Color** pane, choose the required color from the drop-down list.
10. Enter the required **Hello Interval (milliseconds)** and **Multiplier**.
11. Choose the **Path MTU Discovery** value.
12. Enter the **BFD Default DSCP value for tloc color**.
13. Click **Add**.

    The default DSCP and color values are configured on the BFD template.

Apply Policies to Sites and VPNs

In the last window of the policy configuration wizard, you associate the policy blocks that you created on the previous three windows with VPNs and with sites in the overlay network.

To apply a policy block to sites and VPNs in the overlay network:

1. From the Cisco SD-WAN Manager menu, choose **Configuration** > **Policies**. Centralized Policy is selected and displayed by default.
2. Click **Add Policy**. The Create Applications or Groups of Interest page is displayed.
3. Click **Next**. The Network Topology window opens, and in the Topology bar, Topology is selected by default.
4. Click **Next**. The Configure Traffic Rules window opens, and in the Application-Aware Routing bar, Application-Aware Routing is selected by default.
5. Click **Next**. The Apply Policies to Sites and VPNs window opens.
6. In the Policy Name field, enter a name for the policy. This field is mandatory and can contain only uppercase and lowercase letters, the digits 0 through 9, hyphens (–), and underscores (_). It cannot contain spaces or any other characters.
7. In the Policy Description field, enter a description of the policy. It can contain up to 2048 characters. This field is mandatory, and it can contain any characters and spaces.
8. From the Topology bar, select the type of policy block. The table then lists policies that you have created for that type of policy block.
9. Click **Add New Site List** and **VPN list**. Select one or more site lists and select one or more VPN lists. Click **Add**.
10. Click **Preview** to view the configured policy. The policy is displayed in CLI format.
11. Click **Save Policy**. The **Configuration** > **Policies** page appears, and the policies table includes the newly created policy.

For an application-aware route policy to take effect, you apply it to a list of sites in the overlay network:

```
vSmart(config)# apply-policy site-list list-name app-route-policy policy-name
```

When you apply the policy, you do not specify a direction (either inbound or outbound). Application-aware routing policy affects only the outbound traffic on the Cisco SD-WAN devices.

For all **app-route-policy** policies that you apply with **apply-policy** commands, the site IDs across all the site lists must be unique. That is, the site lists must not contain overlapping site IDs. An example of overlapping site IDs are those in the two site lists **site-list 1, site-id 1-100,** and **site-list 2 site-id 70-130**. Here, sites 70 through 100 are in both lists. If you were to apply these two site lists to two different **app-route-policy** policies, the attempt to commit the configuration on the Cisco Catalyst SD-WAN Controller would fail.

The same type of restriction also applies to the following types of policies:

- Centralized control policy (**control-policy**)
- Centralized data policy (**data-policy**)
- Centralized data policy used for cflowd flow monitoring (**data-policy** that includes a **cflowd** action and **apply-policy** that includes a **cflowd-template** command)

You can, however, have overlapping site IDs for site lists that you apply for different types of policy. For example, the sites lists for **app-route-policy** and **data-policy** policies can have overlapping site IDs. So for the two example site lists above, **site-list 1, site-id 1-100,** and **site-list 2 site-id 70-130**, you could apply one to a control policy and the other to a data policy.

As soon as you successfully activate the configuration on the Cisco Catalyst SD-WAN Controller by issuing a **commit** command, the controller pushes the application-aware routing policy to the Cisco SD-WAN devices at the specified sites.

To view the policy configured on the Cisco Catalyst SD-WAN Controller, use the **show running-config** command on the controller.

To view the policy that the Cisco Catalyst SD-WAN Controller has pushed to the device, issue the **show policy from-vsmart** command on the router.

To display flow information for the application-aware applications running on the device, issue the **show app dpi flows** command on the router.

How Application-Aware Routing Policy is Applied in Combination with Other Data Policies

If you configure a Cisco SD-WAN device with application-aware routing policy and with other policies, the policies are applied to data traffic sequentially.

On a Cisco SD-WAN device, you can configure the following types of data policy:

- Centralized data policy. You configure this policy on the Cisco Catalyst SD-WAN Controller, and the policy is passed to the device. You define the configuration with the **policy data-policy configuration** command, and you apply it with the **apply-policy site-list data-policy,** or **apply-policy site-list vpn-membership** command.
- Localized data policy, which is commonly called access lists. You configure access lists on the device with the **policy access-list** configuration command. You apply them, within a VPN, to an incoming interface with the **vpn interface access-list in** configuration command or to an outgoing interface with the **vpn interface access-list out** command.
- Application-aware routing policy. Application-aware routing policy affects only traffic that is flowing from the service side (the local/LAN side) to the tunnel (WAN) side of the Cisco SD-WAN device. You configure application-aware routing policy on the Cisco Catalyst SD-WAN Controller with the **policy app-route-policy** configuration command, and you apply it with the **apply-policy site-list app-route-policy** command. When you commit the configuration, the policy is passed to the appropriate devices. Then, matching data traffic on the device is processed in accordance with the configured SLA conditions. Any data traffic that is not dropped as a result of this policy is passed to the data policy for evaluation. If the data traffic does not match and if no default action is configured, transmit it without SLA consideration.

You can apply only one data policy and one application-aware routing policy to a single site in the overlay network. When you define and apply multiple site lists in a configuration, you must ensure that a single data policy or a single application-aware routing policy is not applied to more than one site. The CLI does not check for this circumstance, and the **validate** configuration command does not detect whether multiple policies of the same type are applied to a single site.

For data traffic flowing from the service side of the router to the WAN side of the router, policy evaluation of the traffic evaluation occurs in the following order:

1. Apply the input access list on the LAN interface. Any data traffic that is not dropped as a result of this access list is passed to the application-aware routing policy for evaluation.
2. Apply the application-aware routing policy. Any data traffic that is not dropped as a result of this policy is passed to the data policy for evaluation. If the data traffic does not match and if no default action is configured, transmit it without SLA consideration.

3. Apply the centralized data policy. Any data traffic that is not dropped as a result of the input access list is passed to the output access list for evaluation.
4. Apply the output access list on the WAN interface. Any data traffic that is not dropped as a result of the output access list is transmitted out the WAN interface.

For data traffic coming from the WAN through the router and into the service-side LAN, the policy evaluation of the traffic occurs in the following order:

1. Apply the input access list on the WAN interface. Any data traffic that is not dropped as a result of the input access list is passed to the data policy for evaluation.
2. Apply the data policy. Any data traffic that is not dropped as a result of the input access list is passed to the output access list for evaluation.
3. Apply the output access list on the LAN interface. Any data traffic that is not dropped as a result of the output access list is transmitted out the LAN interface, towards its destination at the local site.

As mentioned above, application-aware routing policy affects only traffic that is flowing from the service side (the local/LAN side) to the tunnel (WAN) side of the Cisco SD-WAN device, so data traffic inbound from the WAN is processed only by access lists and data policy.

Note When both application-aware routing and data policies are configured, if the data policy rules that contain actions such as redirect DNS, NextHop, secure internet gateway, NAT VPN, or service, the traffic which matches those rules will skip AAR policy even though the traffic also matches rules defined in the AAR policy. Data policy actions override AAR rules.

Activate an Application-Aware Routing Policy

To activate a policy:

1. From the Cisco SD-WAN Manager menu, choose **Configuration** > **Policies**. **Centralized Policy** is selected and displayed by default.
2. For the desired policy, click **...** and select **Activate**. The Activate Policy popup opens. It lists the IP addresses of the reachable Cisco SD-WAN Controllers to which the policy is to be applied.
3. Click **Activate**.

When you activate an application-aware routing policy, the policy is sent to all the connected Cisco SD-WAN Controllers.

Monitor Data Plane Tunnel Performance

The Bidirectional Forwarding Detection (BFD) protocol runs over all data plane tunnels between Cisco SD-WAN devices, monitoring the liveness, and network and path characteristics of the tunnels. Application-aware routing uses the information gathered by BFD to determine the transmission performance of the tunnels. Performance is reported in terms of packet latency and packet loss on the tunnel.

BFD sends Hello packets periodically to test the liveness of a data plane tunnel and to check for faults on the tunnel. These Hello packets provide a measurement of packet loss and packet latency on the tunnel. The Cisco

SD-WAN device records the packet loss and latency statistics over a sliding window of time. BFD keeps track of the six most recent sliding windows of statistics, placing each set of statistics in a separate bucket. If you configure an application-aware routing policy for the device, it is these statistics that the router uses to determine whether a data plane tunnel's performance matches the requirements of the policy's SLA.

The following parameters determine the size of the sliding window:

Parameters	Default	Configuration Command	Range
BFD Hello packet interval	1 second	**bfd color** *color* **hello-interval** *seconds*	1 through 65535 seconds
Polling interval for application-aware routing	10 minutes (600,000 milliseconds)	**bfd app-route poll-interval** *milliseconds*	1 through 4,294,967 ($2^{32} - 1$) milliseconds
Multiplier for application-aware routing	6	**bfd app-route multiplier** *number*	1 through 6

Let us use the default values for these parameters to explain how application-aware routing works:

- For each sliding window time period, application-aware routing sees 600 BFD Hello packets (BFD Hello interval x polling interval: 1 second x 600 seconds = 600 Hello packets). These packets provide measurements of packet loss and latency on the data plane tunnels.
- Application-aware routing retains the statistics for 1 hour (polling interval x multiplier: 10 minutes x 6 = 60 minutes).
- The statistics are placed in six separate buckets, indexed with the numbers 0 through 5. Bucket 0 contains the latest statistics, and bucket 5 the oldest. Every 10 minutes, the newest statistics are placed in bucket 0, the statistics in bucket 5 are discarded, and the remaining statistics move into the next bucket.
- Every 60 minutes (every hour), application-aware routing acts on the loss and latency statistics. It calculates the mean of the loss and latency of all the buckets in all the sliding windows and compares this value to the specified SLAs for the tunnel. If the calculated value satisfies the SLA, application-aware routing does nothing. If the value does not satisfy the SLA, application-aware routing calculates a new tunnel.
- Application-aware routing uses the values in all six buckets to calculate the mean loss and latency for a data tunnel. This is because the multiplier is 6. While application-aware always retains six buckets of data, the multiplier determines how many it actually uses to calculate the loss and latency. For example, if the multiplier is 3, buckets 0, 1, and 2 are used.

Because these default values take action only every hour, they work well for a stable network. To capture network failures more quickly so that application-aware routing can calculate new tunnels more often, adjust the values of these three parameters. For example, if you change just the polling interval to 1 minute (60,000 milliseconds), application-aware routing reviews the tunnel performance characteristics every minute, but it performs its loss and latency calculations based on only 60 Hello packets. It may take more than 1 minute for application-aware routing to reset the tunnel if it calculates that a new tunnel is needed.

To display the next-hop information for an IP packet that a device sends out a service side interface, use the **show policy service-path** command. To view the similar information for packets that the router sends out a WAN transport tunnel interface, use the **show policy tunnel-path** command.

Enable Application Visibility on Cisco SD-WAN Devices

You can enable application visibility directly on Cisco SD-WAN devices, without configuring application-aware routing policy so that you can monitor all the applications running in all VPNs in the LAN. To do this, configure application visibility on the router:

```
vEdge(config)#  policy app-visibility
```

To monitor the applications, use the **show app dpi applications** and **show app dpi supported-applications** commands on the device.

Dampen Data Plane Tunnels

Table 33: Feature History

Feature Name	Release Information	Description
Configure Dampening on Data Plane Tunnels	Cisco SD-WAN Release 20.5.1 Cisco vManage Release 20.5.1	This feature introduces a configurable delay (dampening) mechanism on data plane tunnels to minimize tunnel flapping on the WAN links. The dampening process removes a tunnel from the SLA class until it stops flapping and becomes stable.

Restrictions for Tunnel Dampening

- If there’s no other tunnel available on an SLA class, you can add a tunnel into the SLA class when it meets the SLA class criteria, instead of configuring tunnel dampening. Don't configure dampening on a tunnel when:
  - Rebooting a router
  - Configuring any one of the two transport locators (TLOCs)
  - TLOCs are flapping due to interface shutdown and no-shutdown events.
- Don't configure dampening on a tunnel when there’s only one TLOC present on a Cisco vEdge device or using the tunnel as the circuit of last resort on a Cisco vEdge device.

Information About Tunnel Dampening

In a network operation, you can optimize the path that an application data traffic takes through the network by directing it to WAN links. Some of the WAN links can be unstable and can cause data plane tunnel on the WAN links to flap. This data plane tunnel is the tunnel under detention. The tunnel flapping can change the tunnel quality such that the tunnel SLA classes keep changing. In this scenario, when tunnel quality degrades and it doesn't meet the desired SLA criteria, critical traffic such as, voice or video is redirected to a different tunnel. The traffic flow resumes from the tunnel under detection when the tunnel SLA normalizes.

The tunnel dampening feature allows you to configure a delay mechanism on a tunnel to suppress the effect of traffic flaps due to the tunnel not meeting the SLA criteria. You can dampen a tunnel and remove it from the SLA class to minimize notifications from the tunnel and prevent instability in the network.

Note The dampening of a tunnel suppresses the SLA-change notifications until the tunnel under detection stops flapping and it meets the SLA criteria consistently.

Functionalities of Tunnel Dampening

- The tunnel SLA of an application is calculated based on the values of global poll interval and multiplier.
- You can dampen a tunnel before adding the tunnel into the user-defined SLA class and not into the default SLA class.
- When a tunnel is part of an SLA class, there's no change in the behavior of removing it from the SLA class. The tunnel SLA is calculated at every poll interval based on the values of global multiplier. If the tunnel doesn't meet the SLA criteria, it's taken out from the SLA class.
- The tunnel dampening for an SLA class starts when a tunnel meets the SLA class criteria. If the tunnel doesn't meet the SLA criteria, you can reset the tunnel dampening and start it again when the tunnel meets the SLA class criteria.
- When you configure tunnel dampening on a BFD color, there's no impact on the existing tunnels for this color and they remain in their current SLA class. If the tunnel doesn't meet the SLA criteria, it's taken out from the SLA class, and dampened before adding it back.
- The tunnel dampening feature can be configured only by using the CLI Add-on Feature templates to enter the configuration applicable to your environment.

Default Class Behavior of Tunnel Dampening

By default, every tunnel is part of the default SLA class. When you dampen a tunnel and it isn't part of the SLA class, the traffic on the data plane tunnels between devices falls into default class (if there's no tunnel in the SLA class). The data traffic can then choose any of the available tunnels based on the ECMP method. However, you can use the best of the worst tunnel selection feature (introduced in Cisco SD-WAN Release 20.5.1) to choose the best tunnel from the available tunnels rather than simple ECMP. You can configure the best tunnel criteria on each SLA class because for different SLA classes, the criteria to choose the best tunnel can be different. For example, the best tunnel criteria can be the lowest latency tunnel for voice SLA and lowest loss tunnel for data SLA class.

Configure Tunnel Dampening Using the CLI

1. Configure the poll interval of BFD timers used by application-aware routing.

   ```
   Device# config
   Device(config)# bfd app-route poll-interval poll-interval
   ```

   By default, the poll interval is 10 minutes (600,000 milliseconds).

   The poll interval value specifies the interval at which SLA is calculated periodically for each tunnel.

2. Configure the total number of poll intervals (multiplier) for which the SLA data is maintained.

   The average of all the poll intervals determine the SLA of a tunnel.

```
Device(config)# bfd app-route multiplier multiplier
```

By default, the multiplier value is six.

3. Configure the BFD timers used on transport tunnels (color) for application-aware routing.

```
Device(config)# bfd app-route color color
```

4. Configure dampening on a tunnel.

```
Device(config-color-<color>)# sla-damp-multipler damp-multiplier
```

After you configure tunnel dampening, the dampening remains effective for a duration calculated as a product of damp multiplier and poll interval values.

5. Reset dampening on a tunnel for a specific color.

```
Device# request sla-dampening-reset color color
```

The following is a sample configuration for removing a tunnel from an SLA class within three minutes and dampen it for a duration of one hour (3,600,000 milliseconds).

```
Device# config
Device(config)# bfd app-route poll-interval 60000
Device(config)# bfd app-route multiplier 3
Device(config)# bfd app-route color public-internet
Device(config-color-public-internet)# sla-damp-multiplier 60
!
```

The following is a sample command to reset dampening.

```
Device# request sla-dampening-reset color public-internet
```

Verify Tunnel Dampening

The following is a sample output from the **show app-route stats** command.

```
Device# show app-route stats

app-route statistics 192.168.0.1 192.168.101.2 ipsec 12346 12386
 remote-system-ip 172.16.248.101
 local-color      public-internet
 remote-color     public-internet
 mean-loss
 mean-latency     15
 sla-class-index  0, 1
 Dampening-sla-class-index 2,3
 Dampening-multiplier-left 10,20

TOTAL          AVERAGE  AVERAGE  TX DATA  RX DATA
INDEX  PACKETS  LOSS  LATENCY  JITTER   PKTS     PKTS
------------------------------------------------------------
0      600      0     16       21       0        0
1      600      0     14       18       0        0
2      599      0     17       20       0        0
3      599      0     14       18       0        0
4      600      0     15       19       0        0
5      599      0     15       19       0        0
...
```

In this command output:

Command Output	Purpose
Dampening-sla-class-index	Displays the SLA indexes that are dampened.
Dampening-multiplier-left	Represents the damp-multiplier left for both the SLA indexes.

Configure Application-Aware Routing Using CLIs

Following are the high-level steps for configuring an application-aware routing policy:

1. Create a list of overlay network sties to which the application-aware routing policy is to be applied (in the **apply-policy** command):

   ```
   vSmart(config)# policy
   vSmart(config-policy)# lists site-list list-name
   vSmart(config-site-list)# site-id site-id
   ```

   The list can contain as many site IDs as necessary. Include one **site-id** command for each site ID. For contiguous site IDs, you can specify a range of numbers separated with a dash (–). Create additional site lists, as needed.

2. Create SLA classes and traffic characteristics to apply to matching application data traffic:

   ```
   vSmart(config)# policy sla-class sla-class-name
   vSmart(config-sla-class)# jitter milliseconds
   vSmart(config-sla-class)# latency milliseconds
   vSmart(config-sla-class)# loss percentage
   vSmart(config-sla-class)# app-probe-class app-probe-class
   vSmart(config-sla-class)# fallback-best-tunnelcriterialatencylossjitter
   ```

3. Create lists of applications, IP prefixes, and VPNs to use in identifying application traffic of interest (in the **match** section of the policy definition):

   ```
   vSmart(config)# policy lists
   vSmart(config-lists)# app-list list-name
   vSmart(config-app-list)# (app application-name | app-family family-name)

   vSmart(config-lists)# prefix-list list-name
   vSmart(config-prefix-list)# ip-prefix prefix/length

   vSmart(config-lists)# vpn-list  list-name
   vSmart(config-vpn-list)# vpn vpn-id
   ```

4. If you are configuring a logging action, configure how often to log packets to syslog files:

   ```
   vEdge(config)# policy log-frequency number
   ```

5. Create an application-aware routing policy instance and associate it with a list of VPNs:

   ```
   vSmart(config)# policy app-route-policy policy-name
   vSmart(config-app-route-policy)# vpn-list list-name
   ```

6. Within the policy, create one or more numbered sequence of match–action pairs, where the match parameters define the data traffic and applications of interest and the action parameters specify the SLA class to apply if a match occurs.

   a. Create a sequence:

      ```
      vSmart(config-app-route-policy)# sequence number
      ```

b. Define match parameters for data packets:

```
vSmart(config-sequence)# match  parameters
```

c. Define the action to take if a match occurs:

```
vSmart(config-sequence)# action sla-class sla-class-name [strict]
vSmart(config-sequence)# action sla-class sla-class-name [strict] preferred-color
colors
vSmart(config-sequence)# <userinput>action backup-sla-preferred-color</userinput>
<varname>colors</varname>
```

The first two **action** options direct matching data traffic to a tunnel interface that meets the SLA characteristics in the specified SLA class:

- **sla-class** *sla-class-name*—When you specify an SLA class with no additional parameters, data traffic that matches the SLA is forwarded as long as one tunnel interface is available. The software first tries to send the traffic through a tunnel that matches the SLA. If a single tunnel matches the SLA, data traffic is sent through that tunnel. If two or more tunnels match, traffic is distributed among them. If no tunnel matches the SLA, data traffic is sent through one of the available tunnels.
- **sla-class** *sla-class-name* **preferred-color** *color*—To set a specific tunnel to use when data traffic matches an SLA class, include the **preferred-color** option, specifying the color of the preferred tunnel. If more than one tunnel matches the SLA, traffic is sent to the preferred tunnel. If a tunnel of the preferred color is not available, traffic is sent through any tunnel that matches the SLA class. If no tunnel matches the SLA, data traffic is sent through any available tunnel. In this sense, color preference is considered to be a loose matching, not a strict matching, because data traffic is always forwarded, whether a tunnel of the preferred color is available or not.
- **sla-class** *sla-class-name* **preferred-color** *colors*—To set multiple tunnels to use when data traffic matches an SLA class, include the **preferred-color** option, specifying two or more tunnel colors. Traffic is load-balanced across all tunnels.

If no tunnel matches the SLA, data traffic is sent through any available tunnel. In this sense, color preference is considered to be a loose matching, not a strict matching, because data traffic is always forwarded, whether a tunnel of the preferred color is available or not. When no tunnel matches the SLA, you can choose how to handle the data traffic:

- **strict**—Drop the data traffic.
- **backup-sla-preferred-color** *colors*—Direct the data traffic to a specific tunnel. Data traffic is sent out the configured tunnel if that tunnel interface is available; if that tunnel is unavailable, traffic is sent out another available tunnel. You can specify one or more colors. As with the **preferred-color** option, the backup SLA preferred color is loose matching. In a single **action** configuration, you cannot include both the **strict** and **backup-sla-preferred-color** options.

d. Count the packets or bytes that match the policy:

```
vSmart(config-sequence)# action count counter-name
```

e. Place a sampled set of packets that match the SLA class rule into syslog files:

```
vSmart(config-sequence)# action log
```

f. The match–action pairs within a policy are evaluated in numerical order, based on the sequence number, starting with the lowest number. If a match occurs, the corresponding action is taken and policy evaluation stops.

7. If a packet does not match any of the conditions in one of the sequences, a default action is taken. For application-aware routing policy, the default action is to accept nonmatching traffic and forward it with no consideration of SLA. You can configure the default action so that SLA parameters are applied to nonmatching packets:

   ```
   vSmart(config-policy-name)# default-action sla-class sla-class-name
   ```

8. Apply the policy to a site list:

   ```
   vSmart(config)# apply-policy site-list list-name app-route-policy policy-name
   ```

Configure Application Probe Class Using CLI

Configure app-probe-class, real-time-video and map them with the SLA class as shown in the following example:

```
Device(config)# app-probe-class real-time-video
Device(config)# forwarding-class videofc
Device(config)# color mpls dscp 34
Device(config)# color biz-internet dscp 40
Device(config)# color lte dscp 0


Device(config)# sla-class streamsla
Device(config)# latency 20
Device(config)# loss 10
Device(config)# app-probe-class real-time-video
```

Configure the default value for DSCP using BFD template as shown:

```
Device(config)# bfd default-dscp 50
Device(config)# bfd color mpls 15
```

Application-Aware Routing Policy Configuration Example

This topic shows a straightforward example of configuring application-aware routing policy. This example defines a policy that applies to ICMP traffic, directing it to links with latency of 50 milliseconds or less when such links are available.

You configure application-aware routing policy on a Cisco Catalyst SD-WAN Controller. The configuration consists of the following high-level components:

- Definition of the application (or applications)
- Definition of App Probe Class (Optional)
- Definition of SLA parameters
- Definition of sites, prefixes, and VPNs
- Application-aware routing policy itself
- Specification of overlay network sites to which the policy is applied

The order in which you configure these components is immaterial from the point of view of the CLI. However, from an architectural design point of view, a logical order is to first define all the parameters that are invoked in the application-aware routing policy itself or that are used to apply the policy to various sites in the overlay network. Then, you specify the application-aware routing policy itself and the network sites to which you want to apply the policy.

Here is the procedure for configuring this application-aware routing policy on a Cisco Catalyst SD-WAN Controller:

1. Define the SLA parameters to apply to matching ICMP traffic. In our example, we want to direct ICMP traffic to links that have a latency of 50 milliseconds or less:

```
vSmart# config
vSmart(config)# policy sla-class test_sla_class latency 50
vSmart(config-sla-class-test_sla_class)#
```

2. Define the site and VPN lists to which we want to apply the application-aware routing policy:

```
vSmart(config-sla-class-test_sla_class)# exit
vSmart(config-sla-class-test_sla_class)# lists vpn-list vpn_1_list vpn 1
vSmart(config-vpn-list-vpn_1_list)# exit
vSmart(config-lists)# site-list site_500 site-id 500
vSmart(config-site-list-site_500)#
```

3. Configure the application-aware routing policy. Note that in this example, we apply the policy to the application in two different ways: In sequences 1, 2, and 3, we specify the protocol number (protocol 1 is ICMP, protocol 6 is TCP, and protocol 17 is UDP).

```
vSmart(config-site-list-site_500)# exit
vSmart(config-lists)# exit
vSmart(config-policy)# app-route-policy test_app_route_policy
vSmart(config-app-route-policy-test_app_route_policy)# vpn-list vpn_1_list
vSmart(config-vpn-list-vpn_1_list)# sequence 1 match protocol 6
vSmart(config-match)# exit
vSmart(config-sequence-1)# action sla-class test_sla_class strict
vSmart(config-sequence-1)# exit
vSmart(config-vpn-list-vpn_1_list)# sequence 2 match protocol 17
vSmart(config-match)# exit
vSmart(config-sequence-2)# action sla-class test_sla_class
vSmart(config-sequence-2)# exit
vSmart(config-vpn-list-vpn_1_list)# sequence 3 match protocol 1
vSmart(config-match)# exit
vSmart(config-sequence-3)# action sla-class test_sla_class strict
vSmart(config-sequence-3)# exit
vSmart(config-sequence-4)#
```

4. Apply the policy to the desired sites in the Cisco SD-WAN overlay network:

```
vSmart(config-sequence-4)# top
vSmart(config)# apply-policy site-list site_500 app-route-policy test_app_route_policy
```

5. Display the configuration changes:

```
vSmart(config-site-list-site_500)# top
vSmart(config)# show config
```

6. Validate that the configuration contains no errors:

```
vSmart(config)# validate
Validation complete
```

7. Activate the configuration:

```
vSmart(config)# commit
Commit complete.
```

8. Exit from configuration mode:

```
vSmart(config)# exit
vSmart#
```

Putting all the pieces of the configuration together gives this configuration:

```
vSmart# show running-config policy
policy
 sla-class test_sla_class
  latency 50
 !
 app-route-policy test_app_route_policy
  vpn-list vpn_1_list
   sequence 1
    match
     protocol 6
    !
    action sla-class test_sla_class strict
   !
   sequence 2
    match
     protocol 17
    !
    action sla-class test_sla_class
   !
   sequence 3
    match
     protocol 1
    !
    action sla-class test_sla_class strict
   !
  !
 !
 lists
  vpn-list vpn_1_list
   vpn 1
  !
  site-list site_500
   site-id 500
  !
  site-list site_600
   site-id 600
  !
 !
!
apply-policy
 site-list site_500
  app-route-policy test_app_route_policy
 !
!
```

The following example defines the multicast protocol:

```
policy
!
 sla-class SLA_BEST_EFFORT
  jitter 900
 !
 sla-class SLA_BUSINESS_CRITICAL
  loss    1
```

```
 latency 250
 jitter  300
!
sla-class SLA_BUSINESS_DATA
 loss    3
 latency 400
 jitter  500
!
sla-class SLA_REALTIME
 loss    2
 latency 300
 jitter  60
!
app-route-policy policy_multicast
 vpn-list multicast-vpn-list
  sequence 10
   match
    source-ip      10.0.0.0/8
    destination-ip 10.255.255.254/8
   !
   action
    count mc-counter-10
    sla-class SLA_BUSINESS_CRITICAL
   !
  !
  sequence 15
   match
    source-ip      172.16.0.0/12
    destination-ip 172.31.255.254/12
   !
   action
    count mc-counter-15
    sla-class SLA_BEST_EFFORT
   !
  !
  sequence 20
   match
    destination-ip 192.168.0.1
   !
   action
    count mc-counter-20
    sla-class SLA_BUSINESS_CRITICAL
   !
  !
  sequence 25
   match
    protocol       17
   !
   action
    count mc-counter-25
    sla-class SLA_REALTIME
   !
  !
  sequence 30
   match
    source-ip      192.168.0.0/16
    destination-ip 192.168.255.254
    protocol       17
   !
   action
    count mc-counter-30
    sla-class SLA_BUSINESS_DATA preferred-color lte
   !
  !
```

```
   default-action sla-class SLA_BEST_EFFORT
   !
   sequence 35
    match
     source-ip      10.0.0.0/8
     destination-ip 10.255.255.254/8
     protocol       17
    !
    action
     count mc-counter-35
     sla-class SLA_BUSINESS_DATA preferred-color lte
     backup-sla-preferred-color 3g
    !
   !
 lists
  vpn-list multicast-vpn-list
   vpn 1
   vpn 60
   vpn 4001-4010
   vpn 65501-65510
  !
  site-list multicast-site-list
   site-id 1100
   site-id 500
   site-id 600
  !
 !
!
apply-policy
 site-list multicast-site-list
  app-route-policy policy_multicast
 !
!
```

Ranking Color Prreference Example

```
app-route-policy SAMPLE _AAR
 vpn-list ONE
  sequence 10
   match
    dscp 46
   !
   action
    sla VOICE_SLA strict preferred-color-group GROUP2_COLORS
   !
  !
  sequence 20
   match
    dscp 34
   !
   action
    sla VOICE_SLA preferred-color-group GROUP1_COLORS
   !
  !
  sequence 30
   match
    dscp 28
   !
   action
    sla VOICE_SLA preferred-color-group GROUP3_COLORS
   !
  !
 !
policy lists
```

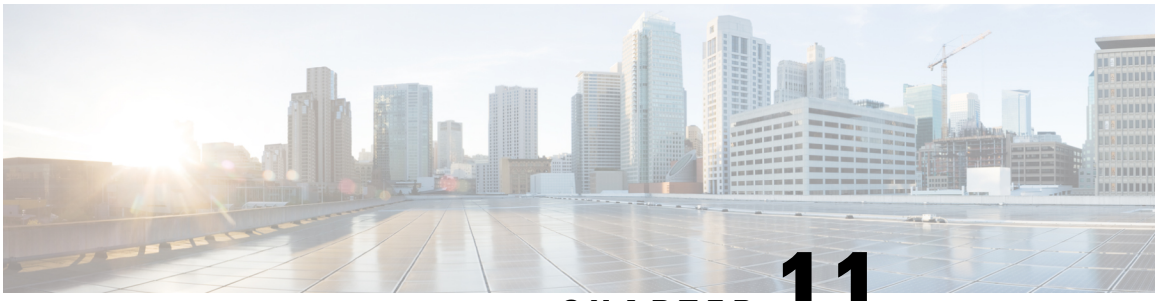
```
 preferred-color-group GROUP1_COLORS
  primary-preference
   color-preference biz-internet
   path-preference direct-tunnel
  !
  secondary-preference
   color-preference mpls
   path-preference multi-hop-path
  !
  tertiary-preference
   color-preference lte
  !
 !
 preferred-color-group GROUP2_COLORS
  primary-preference
   color-preference mpls
  !
  secondary-preference
   color-preference biz-internet
  !
 !
 preferred-color-group GROUP3_COLORS
  primary-preference
   color-preference mpls biz-internet lte
  !
```

Note You can configure path-preference option only if you enable the Multi-Region Fabric option in Cisco SD-WAN Manager.

AAR Policy for IPv6 Applications Example

```
policy
  sla-class Default
   jitter 100
   latency 300
   loss 25
  !
 app-route-policy _VPN1_AAR-Policy-for-IPv6-Traffic
  vpn-list VPN1
    sequence 1
     match
      app-list Msft-0365
     !
     action
      sla-class Default  preferred-color public-internet
     !
    !
 !
 lists
  app-list Msft-0365
   app ms-office-web-apps
  !
  site-list SITE-100
   site-id 100
  !
  vpn-list VPN1
   vpn 1
  !
 !
```

```
!
apply-policy
 site-list SITE-100
  app-route-policy _VPN1_AAR-Policy-for-IPv6-Traffic
 !
!
```



CHAPTER 11

Traffic Flow Monitoring

To achieve simplification and consistency, the Cisco SD-WAN solution has been rebranded as Cisco Catalyst SD-WAN. In addition, from Cisco IOS XE SD-WAN Release 17.12.1a and Cisco Catalyst SD-WAN Release 20.12.1, the following component changes are applicable: **Cisco vManage** to **Cisco Catalyst SD-WAN Manager**, **Cisco vAnalytics** to **Cisco Catalyst SD-WAN Analytics**, **Cisco vBond** to **Cisco Catalyst SD-WAN Validator**, **Cisco vSmart** to **Cisco Catalyst SD-WAN Controller**, and **Cisco Controllers** to **Cisco Catalyst SD-WAN Control Components**. See the latest Release Notes for a comprehensive list of all the component brand name changes. While we transition to the new names, some inconsistencies might be present in the documentation set because of a phased approach to the user interface updates of the software product.

Traffic Flow Monitoring

Information About Traffic Flow Monitoring

The following sections describe traffic flow monitoring.

Traffic Flow Monitoring with Cflowd Overview

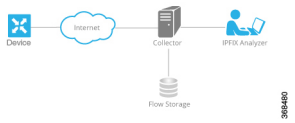
Cflowd is a flow analysis tool, used for analyzing Flexible NetFlow (FNF) traffic data. It monitors traffic flowing through Cisco vEdge devices in the overlay network and exports flow information to a collector, where it can be processed by an IP Flow Information Export (IPFIX) analyzer. For a traffic flow, Cflowd periodically sends template reports to flow collector. These reports contain information about the flows and the data is extracted from the payload of these reports.

You can create a Cflowd template that defines the location of Cflowd collectors, how often sets of sampled flows are sent to the collectors, and how often the template is sent to the collectors (on Cisco SD-WAN Controllers and on Cisco SD-WAN Manager). You can configure a maximum of four Cflowd collectors per Cisco vEdge device. To have a Cflowd template take effect, apply it with the appropriate data policy.

You must configure at least one Cflowd template, but it need not contain any parameters. With no parameters, the data flow cache on the nodes is managed using default settings, and no flow export occurs.

Cflowd traffic flow monitoring is equivalent to FNF.

The Cflowd software implements Cflowd version 10, as specified in *RFC 7011* and *RFC 7012*. Cflowd version 10 is also called the IP Flow Information Export (IPFIX) protocol.



Cflowd performs 1:1 sampling. Information about all flows is aggregated in the Cflowd records; flows are not sampled. Cisco vEdge devices do not cache any of the records that are exported to a collector.

Note NetFlow on Secure Internet Gateway (SIG) tunnels is not supported on Cisco vEdge devices.

Cflowd and SNMP Comparison

Cflowd monitors service side traffic. Cflowd mainly monitors traffic from LAN to WAN, WAN to LAN, LAN to LAN and DIA. If you use Cflowd and SNMP to monitor traffic of LAN interface (input or output), then packets and bytes should be similar. The difference of bytes in SNMP starts from L2 header, but Cflowd starts from L3 header. However, if we use Cflowd and SNMP to monitor traffic of WAN interface (input or output), then packets or bytes are unlikely to be the same. All the traffic of WAN interfaces is not service side traffic. For example, Cflowd does not monitor BFD traffic, but SNMP does. The packets or bytes of Cflowd and SNMP traffic are not the same.

Components of Cflowd

In the overlay network, you configure cflowd using a centralized data policy. As part of the policy, you specify the location of the collector.

By default, flow information is sent to the collector every 60 seconds. You can modify this and other timers related to how often cflowd templates are refreshed and how often a traffic flow times out.

You can configure many cflowd policies, but in one single cflowd policy, you can configure at most four external collectors. When you configure a new data policy that changes which flows are sampled, the software allows the old flows to expire gracefully rather than deleting them all at once.

The Cisco vEdge device exports template records and data records to a collector. The template record is used by the collector to parse the data record information that is exported to it.

Note Option templates are not supported on Cisco vEdge devices.

The source IP address for the packet containing the IPFIX records is selected from the collector that is closer to the interfaces in the VPN. The flow records are exported through TCP or UDP connections for Cisco devices. Anonymization of records and TLS encryption are not performed, because it is assumed that the collector and the IPFIX analyzer are both located within the data center, traffic traveling within the data center is assumed to be safe.

Cflowd can track GRE, ICMP, IPsec, SCTP, TCP, and UDP flows.

IPFIX Information Elements for Cisco vEdge Devices

The Cisco Catalyst SD-WAN cflowd software exports the following IPFIX information elements to the cflowd collector. These information elements are a subset of those defined in *RFC 7012* and maintained by IANA. The elements are exported in the order listed. You cannot modify the information elements that are exported, nor can you change the order in which they appear.

Information Element	Element ID	Description	Data Type	Data Type Semantics	Units or Range
ipClassOfService	5	Value of type of service (TOS) field in the IPv4 packet header.	unsigned8 (1 byte)	identifier	—
ipNextHopIPv4Address	15	IPv4 address of the next IPv4 hop.	IPv4Address (4 bytes)	default	—
minimumIpTotalLength	25	Length of the smallest packet observed for this flow. The packet length includes the IP headers and the IP payload.	unsigned64 (8 bytes)	—	Octets
maximumIpTotalLength	26	Length of the largest packet observed for this flow. The packet length includes the IP headers and the IP payload.	unsigned64 (8 bytes)	—	Octets
icmpTypeCodeIPv4	32	Type and Code of the IPv4 ICMP message. The combination of both values is reported as (ICMP type * 256) + ICMP code.	unsigned16 (2 bytes)	identifier	—
octetTotalCount	85	Total number of octets in incoming packets for this flow at the observation point since initialization or re-initialization of the metering process for the observation point. The count includes the IP headers and the IP payload.	unsigned64 (8 bytes)	totalCounter	Octets
packetTotalCount	86	Total number of incoming packets for this flow at the observation point since initialization or re-initialization of the metering process for the observation point.	unsigned64 (8 bytes)	totalCounter	Packets
flowStartSeconds	150	Absolute timestamp of the first packet of this flow.	dateTime-Seconds (4 bytes)	—	—
flowEndSeconds	151	Absolute timestamp of the last packet of this flow.	dateTime-Seconds (4 bytes)	—	—
ipPrecedence	196	Value of IP precedence. This value is encoded in the first 3 bits of the IPv4 TOS field.	unsigned8 (1 byte)	flags	0 through 7
paddingOctets	210	Value of this Information Element is always a sequence of 0x00 values.	octetArray	default	—

Information About Configuring a Maximum FNF Record Rate for Aggregated Data

Minimum supported releases: Cisco IOS XE Catalyst SD-WAN Release 17.14.1a, Cisco Catalyst SD-WAN Control Components Release 20.14.1

Raw and Aggregated Traffic Flow Data

When traffic flow visibility is enabled (see Configure Global Flow Visibility), devices in the network send raw and aggregated traffic flow data to Cisco SD-WAN Manager.

To aggregate flow data, routers use 4-tuples of flow data (containing VPN ID, application name, ingress interface of the flow, and egress interface of the flow) as a key for consolidating the raw data of multiple flows. The router consolidates each flow for which the 4-tuple is identical into a single aggregated FNF record.

Cisco SD-WAN Manager uses the aggregated data to provide a high-level view of network traffic flow information. The aggregated data shows the network applications that are producing traffic, but is less granular than the full traffic flow data. It does not provide source and destination addresses, or source and destination ports for traffic flows.

For a detailed view of traffic flows, use functions such as On Demand Troubleshooting. For information about On Demand Troubleshooting, see On-Demand Troubleshooting.

Maximum FNF Record Rate

You can configure a maximum rate (records per minute) of aggregated traffic data FNF records that a device can send to reduce the performance demands (CPU and memory) on the device. This may be helpful when there is a large number of applications producing network traffic. For information about configuring this, see Configure the Maximum FNF Record Rate for Aggregated Data, Using CLI Commands, on page 187.

Restrictions for Traffic Flow Monitoring

The following sections describe notes, limitations, and restrictions related to traffic flow monitoring.

Restrictions for Enabling Collect Loopback in Flow Telemetry When Using Loopbacks as TLOCs

- Supports configuration only through the Cisco Catalyst SD-WAN Controller CLI or Cisco SD-WAN Manager CLI-template. Feature template is not supported for this release.
- Collect loopback in FNF VPN0 interfaces is not supported.
- Collect loopback in the Decidated Internet Acccess (DIA) scenario, is not supported.
- Multi-tenant scenario is not supported.

Configure Traffic Flow Monitoring

The following sections provide information about configuring traffic flow monitoring.

Configure Cflowd Traffic Flow Monitoring

This topic provides general procedures for configuring Cflowd traffic flow monitoring. You configure Cflowd traffic flow monitoring using the basic components of centralized data policy. Cflowd template and Cflowd in data policy are independent of each other.

To configure policy for Cflowd traffic flow monitoring, use the Cisco SD-WAN Manager policy configuration wizard. The wizard consists of four sequential screens that guide you through the process of creating and editing policy components:

1. Create Applications or Groups of Interest—Create lists that group together related items and that you call in the match or action components of a policy.
2. Configure Topology—Create the network structure to which the policy applies.
3. Configure Traffic Rules—Create the match and action conditions of a policy.
4. Apply Policies to Sites and VPNs—Associate policy with sites and VPNs in the overlay network.

In the first three policy configuration wizard screens, you are creating policy components or blocks. In the last screen, you are applying policy blocks to sites and VPNs in the overlay network.

For the Cflowd policy to take effect, you must activate the policy.

Step 1: Start the Policy Configuration Wizard

To start the policy configuration wizard:

1. In the Cisco SD-WAN Manager NMS, select the **Configure** > **Policies** screen. When you first open this screen, the **Centralized Policy** tab is selected by default.
2. Click **Add Policy**.

The policy configuration wizard opens, and the Create Applications or Groups of Interest screen is displayed.

Step 2: Create Applications or Groups of Interest

To create lists of applications or groups to use in Cflowd policy:

1. Create new lists as described in the following table:
   - Prefix
     a. In the left bar, click **Prefix**.
     b. Click **New Prefix List**.
     c. Enter a name for the list.
     d. In the Add Prefix field, enter one or more data prefixes separated by commas.
     e. Click **Add**.
   - Site
     a. In the left bar, click **Site**.
     b. Click **New Site List**.

c. Enter a name for the list.

d. In the Add Site field, enter one or more site IDs separated by commas.

e. Click **Add**.

- VPN

a. In the left bar, click **VPN**.

b. Click **New VPN List**.

c. Enter a name for the list.

d. In the Add VPN field, enter one or more VPN IDs separated by commas.

e. Click **Add**.

2. Click **Next** to Configure Topology in the wizard. When you first open this screen, the Topology tab is selected by default.

Step 3: Configure the Network Topology

To configure the network topology:

In the Topology tab, create a network topology as described:

1. Hub and Spoke - Policy for a topology with one or more central hub sites and with spokes connected to a hub

a. In the Add Topology drop-down, select **Hub and Spoke**.

b. Enter a name for the hub-and-spoke policy.

c. Enter a description for the policy.

d. In the VPN List field, select the VPN list for the policy.

e. In the left pane, click **Add Hub and Spoke**. A hub-and-spoke policy component containing the text string My Hub-and-Spoke is added in the left pane.

f. Double-click the **My Hub-and-Spoke** text string, and enter a name for the policy component.

g. In the right pane, add hub sites to the network topology:

1. Click **Add Hub Sites**.

2. In the **Site List** field, select a site list for the policy component.

3. Click **Add**.

4. Repeat Steps 7a, 7b, and 7c to add more hub sites to the policy component.

h. In the right pane, add spoke sites to the network topology:

1. Click **Add Spoke Sites**.

2. In the **Site List** field, select a site list for the policy component.

3. Click **Add**.

4. Repeat Steps 8a, 8b, and 8c to add more spoke sites to the policy component.

i. Repeat Steps 5 through 8 to add more components to the hub-and-spoke policy.

j. Click **Save Hub and Spoke Policy**.

2. Mesh - Partial-mesh or full-mesh region

a. In the Add Topology drop-down, select **Mesh**.

b. Enter a name for the mesh region policy component.

c. Enter a description for the mesh region policy component.

d. In the VPN List field, select the VPN list for the policy.

e. Click **New Mesh Region**.

f. In the Mesh Region Name field, enter a name for the individual mesh region.

g. In the Site List field, select one or more sites to include in the mesh region.

h. Repeat Steps 5 through 7 to add more mesh regions to the policy.

i. Click **Save Mesh Region**.

To use an existing topology:

1. In the Add Topology drop-down, click **Import Existing Topology**. The Import Existing Topology popup displays.
2. Select the type of topology.
3. In the Policy drop-down, select the name of the topology.
4. Click **Import**.

Click **Next** to move to Configure Traffic Rules in the wizard. When you first open this screen, the Application-Aware Routing tab is selected by default.

Step 4: Configure Traffic Rules

To configure traffic rules for Cflowd policy:

1. In the Application-Aware Routing bar, select the **Cflowd** tab.
2. Click the **Add Policy** drop-down.
3. Select **Create New**. The Add Cflowd Policy popup opens.
4. Configure timer parameters for the Cflowd template:

a. In the Active Flow Timeout field, specify how long to collect a set of flows on which traffic is actively flowing, a value from 30 through 3,600 seconds. The default is 600 seconds (10 minutes).

b. In the Inactive Flow Timeout field, specify how long to wait to send a set of sampled flows to a collector for a flow on which no traffic is flowing, a value from 1 through 3,600 seconds. The default is 60 seconds (1 minute).

c. In the Flow Refresh Interval field, specify how often to send the Cflowd template record fields to the collector, a value from 60 through 86,400 seconds (1 minute through 1 day). The default is 90 seconds.

d. In the Sampling Interval field, specify how many packets to wait before creating a new flow, a value from 1 through 65,536 seconds. While you can configure any integer value, the software rounds the value down to the nearest power of 2.

5. Click **Add New Collector**, and configure the location of the Cflowd collector. You can configure up to four collectors.

   a. In the VPN ID field, enter the number of the VPN in which the collector is located.

   b. In the IP Address field, enter the IP address of the collector.

   c. In the Port Number field, enter the collector port number. The default port is 4739.

   d. In the Transport Protocol drop-down, select the transport type to use to reach the collector, either TCP or UDP.

   e. In the Source Interface field, enter the name of the interface to use to send flows to the collector. It can be either a Gigabit Ethernet, a 10-Gigabit Ethernet interface (**ge**), or a loopback interface (**loopback** *number*).

6. Click **Save Cflowd Policy**.

7. Click **Next** to move to Apply Policies to Sites and VPNs in the wizard.

Step 5: Apply Policies to Sites and VPNs

To apply a policy block to sites and VPNs in the overlay network:

1. If you are already in the policy configuration wizard, skip to Step 6. Otherwise, in the Cisco SD-WAN Manager NMS, select the **Configure** > **Policies** screen. When you first open this screen, the Centralized Policy tab is selected by default.

2. Click **Add Policy**. The policy configuration wizard opens, and the Create Applications or Groups of Interest screen is displayed

3. Click **Next**. The Network Topology screen opens, and in the Topology bar, the Topology tab is selected by default.

4. Click **Next**. The Configure Traffic Rules screen opens, and in the Application-Aware Routing bar, the Application-Aware Routing tab is selected by default.

5. Click **Next**. The Apply Policies to Sites and VPNs screen opens.

6. In the Policy Name field, enter a name for the policy. This field is mandatory and can contain only uppercase and lowercase letters, the digits 0 through 9, hyphens (–), and underscores (_). It cannot contain spaces or any other characters.

7. In the Policy Description field, enter a description of the policy. It can contain up to 2048 characters. This field is mandatory, and it can contain any characters and spaces.

8. From the Topology bar, select the type of policy block. The table then lists policies that you have created for that type of policy block.

9. Click **Add New Site List**. Select one or more site lists, and click **Add**.

10. Click **Preview** to view the configured policy. The policy is displayed in CLI format.
11. Click **Save Policy**. The **Configuration** > **Policies** screen opens, and the policies table includes the newly created policy.

Step 6: Activate a Centralized Policy

Activating a Cflowd policy sends that policy to all connected Cisco Catalyst SD-WAN Controllers. To activate a Cflowd policy:

1. In the Cisco SD-WAN Manager NMS, select the **Configure** > **Policies** screen. When you first open this screen, the Centralized Policy tab is selected by default.
2. Select a policy.
3. Click the **More Actions** icon to the right of the row, and click **Activate**. The Activate Policy popup opens. It lists the IP addresses of the reachable Cisco Catalyst SD-WAN Controllers to which the policy is to be applied.
4. Click **Activate**.

Configure Cflowd Traffic Flow Monitoring Using the CLI

Following are the high-level steps for configuring a Cflowd centralized data policy to perform traffic monitoring and to export traffic flows to a collector:

1. Create a list of overlay network sites to which the Cflowd centralized data policy is to be applied (in the **apply-policy** command).

   ```
   vSmart(config)# policy
   vSmart(config-policy)# lists site-list list-name
   vSmart(config-lists-list-name)# site-id site-id
   ```

   The list can contain as many site IDs as necessary. Include one **site-id** command for each site ID. For contiguous site IDs, you can specify a range of numbers separated with a dash (–). Create additional site lists, as needed.

2. Create a list of VPN for which the Cflowd centralized data policy is to be configured (in the **policy data-policy** command).

   ```
   vSmart(config)# policy lists
   vSmart(config-lists)# vpn-list list-name
   vSmart(config-lists-list-name)# vpn vpn-id
   ```

3. Create lists of IP prefixes, as needed.

   ```
   vSmart(config)# policy lists
   vSmart(config-lists)# prefix-list list-name
   vSmart(config-lists-list-name)# ip-prefix prefix/length
   ```

4. Configure a Cflowd template, and optionally, configure template parameters, including the location of the Cflowd collector, the flow export timers, and the flow sampling interval.

   ```
   vSmart(config)# policy cflowd-template template-name
   vSmart(config-cflowd-template-template-name)# collector vpn vpn-id address ip-address
    port port-number transport-type (transport_tcp | transport_udp) source-interface
   interface-name
   vSmart(config-cflowd-template-template-name)# flow-active-timeout seconds
   vSmart(config-cflowd-template-template-name)# flow-inactive-timeout seconds
   ```

```
vSmart(config-cflowd-template-template-name)# protocol ipv4/ipv6/both
vSmart(config-cflowd-template-template-name)# template-refresh seconds
```

You must configure a Cflowd template, but it need not contain any parameters. With no parameters, the data flow cache on router is managed using default settings, and no flow export occurs. You can configure one Cflowd template per router, and it can export to a maximum of four collectors.

By default, an actively flowing data set is exported to the collector every 60 seconds (1 minute), a data set for a flow on which no traffic is flowing is sent every 10 seconds, and the Cflowd template record fields (the three timer values) are sent to the collector every 600 seconds.

Also by default, a new flow is created immediately after an existing flow has ended. If you modify the configuration of the template record fields, the changes take effect only on flows that are created after the configuration change has been propagated to the router. Because an existing flow continues indefinitely, to have configuration changes take effect, clear the flow with the **clear app cflowd flows** command.

Note On Cisco IOS XE Catalyst SD-WAN devices, a flow-active-timeout is fixed as 60 seconds. If a flow-inactive-timeout is fixed as 10 seconds. The **flow-active-timeout** and **flow-inactive-timeout** value that is configured on Cisco SD-WAN Controller or Cisco SD-WAN Manager do not take effect on Cisco IOS XE Catalyst SD-WAN devices.

5. If you configure a logging action, configure how often to log packets to the syslog files.

   ```
   vEdge(config)# policy log-frequency number
   ```

6. Create a data policy instance and associate it with a list of VPNs.

   ```
   vSmart(config)# policy data-policy policy-name
   vSmart(config-data-policy-policy-name)# vpn-list list-name
   ```

7. Create a sequence to contain a single match–action pair.

   ```
   vSmart(config-vpn-list-list-name)# sequence number
   vSmart(config-sequence-number)#
   ```

   The match–action pairs are evaluated in order, by sequence number, starting with the lowest numbered pair and ending when the route matches the conditions in one of the pairs. If no match occurs, the default action is taken.

8. Define match parameters for the data packets.

   ```
   vSmart(config-sequence-number)# match  parameters
   ```

9. Enable Cflowd action.

   ```
   vSmart(config-sequence-number)# action cflowd
   ```

10. In the action, count or log data packets.

    ```
    vSmart(config-sequence-number)# action count counter-name
    vSmart(config-sequence-number)# action log
    ```

11. Create additional numbered sequences of match–action pairs within the data policy, as needed.

12. If a route does not match any of the conditions in one of the sequences, it is rejected by default. If you want nonmatching prefixes to be accepted, configure the default action for the policy.

    ```
    vSmart(config-policy-name)# default-action accept
    ```

13. Apply the policy and the Cflowd template to one or more sites in the overlay network.

```
vSmart(config)# apply-policy site-list list-name data-policy policy-name
vSmart(config)# apply-policy site-list list-name cflowd-template template-name
```

Configuration Examples for Flexible NetFlow Export of BFD Metrics

Minimum releases: Cisco IOS XE Catalyst SD-WAN Release 17.10.1a and Cisco Catalyst SD-WAN Control Components Release 20.10.1

The following example shows a centralized policy configuration with export of BFD metrics enabled:

```
Device# show sdwan policy from-vsmart
from-vsmart cflowd-template fnf
 flow-active-timeout    600
 flow-inactive-timeout  60
 template-refresh       600
 flow-sampling-interval 1
 protocol               ipv4
 customized-ipv4-record-fields
  no collect-tos
  no collect-dscp-output
 collector vpn 0 address 10.0.100.1 port 4739 transport transport_udp
  bfd-metrics-export
   export-interval 600
```

The following example shows FNF BFD telemetry data with average jitter, average latency, and loss metrics:

```
{ 'Data_Template': 'Data_Flow',
  'ObservationDomainId': 6,
  'Version': 10,
  'arrive_time': 1658807309.2496994,
  'dfs_tfs_length': 200,
  'export_dfs_tfs_templates_list_dict': { 'FlowSequence': 3354,
                                          'Flowset_id': '258',
                                          'Flowset_length': 200,
                                          'Length': 286,
                                          'ObservationDomainId': 6,
                                          'TimeStamp': 1658807269,
                                          'Version': 10,
                                          'flow': [ { 'bfd_avg_jitter': 1000,
                                                      'bfd_avg_latency': 1000,
                                                      'bfd_loss': 15,
                                                      'bfd_pfr_update_ts': 1658806692155,
                                                      'bfd_rx_cnt': 0,
                                                      'bfd_tx_cnt': 0,
                                                      'ipDiffServCodePoint': 48,
                                                     'tloc_table_overlay_session_id': 10},

                                                      …
                                                  ]},
 'flow_length': 4,
  'flow_time': 1658807269,
  'flowset_id': '258',
  'header': { 'FlowSequence': 3354,
              'Length': 286,
              'ObservationDomainId': 6,
              'TimeStamp': 1658807269,
              'Version': 10},
  'host': '10.0.100.15',
  'ipfix_length': 286,
  'packet_number': 2,
  'template_id': '258'}
```

Apply and Enable Cflowd Policy

For a centralized data policy to take effect, you must apply it to a list of sites in the overlay network:

```
vSmart(config)# apply-policy site-list list-name data-policy policy-name
```

To activate the Cflowd template, associate it with the data policy:

```
vSmart(config)# apply-policy cflowd-template template-name
```

For all **data-policy** policies that you apply with **apply-policy** commands, the site IDs across all the site lists must be unique. That is, the site lists must not contain overlapping site IDs. An example of overlapping site IDs are those in the two site lists **site-list 1 site-id 1-100** and **site-list 2 site-id 70-130**. Here, sites 70 through 100 are in both lists. If you apply these two site lists to two different **data-policy** policies, the attempt to commit the configuration on the Cisco Catalyst SD-WAN Controller would fail.

The same type of restriction also applies to the following types of policies:

- Application-aware routing policy (**app-route-policy**)
- Centralized control policy (**control-policy**)
- Centralized data policy (**data-policy**)

You can, however, have overlapping site IDs for site lists that you apply for different types of policy. For example, the sites lists for **control-policy** and **data-policy** policies can have overlapping site IDs. So for the two example site lists above, **site-list 1 site-id 1-100** and **site-list 2 site-id 70-130**, you could apply one to a control policy and the other to a data policy.

After you successfully activate the configuration by issuing a **commit** command, the Cisco Catalyst SD-WAN Controller pushes the data policy to the Cisco vEdge devices located in the specified sites. To view the policy as configured on the Cisco Catalyst SD-WAN Controller, use the **show running-config** command in the Cisco Catalyst SD-WAN Controller. To view the policy that has been pushed to the device, use the **show policy from-vsmart** command on the device.

To display the centralized data policy as configured on the Cisco Catalyst SD-WAN Controller, use the **show running-config** command:

```
vSmart# show running-config policy
vSmart# show running-config apply-policy
```

Enable Cflowd Visibility on Cisco vEdge devices

You can enable Cflowd visibility directly on Cisco vEdge devices, without configuring a data policy, so that you can perform traffic-flow monitoring on traffic coming to the router from all VPNs in the LAN. To do this, configure Cflowd visiblity on the device:

```
Device(config)# policy flow-visibility
```

To monitor the applications, use the **show app cflowd flows** and **show app cflowd statistics** commands on the device.

Cflowd Traffic Flow Monitoring Configuration Examples

This topic shows a complete example of configuring traffic flow monitoring.

Configuration Steps

Enable Cflowd traffic monitoring with a centralized data policy, so all configuration is done on a Cisco Catalyst SD-WAN Controller. The following example procedure monitors all TCP traffic, sending it to a single collector:

1. Create a Cflowd template to define the location of the collector and to modify Cflowd timers.

```
vsmart(config)# policy cflowd-template test-cflowd-template
vsmart(config-cflowd-template-test-cflowd-template)# collector vpn 1 address 172.16.155.15
 port 13322 transport transport_udp
vsmart(config-cflowd-template-test-cflowd-template)# flow-inactive-timeout 60
vsmart(config-cflowd-template-test-cflowd-template)# template-refresh 90
```

2. Create a list of VPNs whose traffic you want to monitor.

```
vsmart(config)# policy lists vpn-list vpn_1 vpn 1
```

3. Create a list of sites to apply the data policy to.

```
vsmart(config)# policy lists site-list cflowd-sites site-id 400,500,600
```

4. Configure the data policy.

```
vsmart(config)# policy data-policy test-cflowd-policy
vsmart(config-data-policy-test-cflowd-policy)# vpn-list vpn_1
vsmart(config-vpn-list-vpn_1)# sequence 1
vsmart(config-sequence-1)# match protocol 6
vsmart(config-match)# exit
vsmart(config-sequence-1)# action accept cflowd
vsmart(config-action)# exit
vsmart(config-sequence-1)# exit
vsmart(config-vpn-list-vpn_1)# default-action accept
```

5. Apply the policy and the Cflowd template to sites in the overlay network.

```
vsmart(config)# apply-policy site-list cflowd-sites data-policy test-cflowd-policy
Device(config-site-list-cflowd-sites)# cflowd-template test-cflowd-template
```

6. Activate the data policy.

```
vsmart(config-site-list-cflowd-sites)# validate
Validation complete
vsmart(config-site-list-cflowd-sites)# commit
Commit complete.
vsmart(config-site-list-cflowd-sites)# exit configuration-mode
```

Example Configuration

Here is a complete example of a Cflowd configuration:

```
vsmart(config)# show configuration
apply-policy
 site-list cflowd-sites
  data-policy     test-cflowd-policy
  cflowd-template test-cflowd-template
 !
!
policy
 data-policy test-cflowd-policy
  vpn-list vpn_1
   sequence 1
    match
     protocol 6
```

```
    !
    action accept
     cflowd
    !
   !
   default-action accept
  !
 !
 cflowd-template test-cflowd-template
  flow-inactive-timeout 60
  template-refresh      90
  collector vpn 1 address 192.168.0.1 protocol ipv4 port 13322 transport transport_udp
 !
 lists
  vpn-list vpn_1
   vpn 1
  !
  site-list cflowd-sites
   site-id 400,500,600
  !
 !
!
```

Verify Cflowd Configuration

To verify the Cflowd configuration after activating it on the Cisco Catalyst SD-WAN Controller, use the **show running-config policy** and **show running-config apply-policy** commands.

Starting from Cisco IOS XE Catalyst SD-WAN Release 17.12.1a, the cflowd commands have been enhanced for both IPv4 and IPv6 flow records.

```
Device# show flow record sdwan_flow_record-xxx
```

IPv4 flow record:

```
flow record sdwan_flow_record-1666223692122679:
  Description:        flow and application visibility records
  No. of users:       1
  Total field space:  102 bytes
  Fields:
    match ipv4 protocol
    match ipv4 source address
    match ipv4 destination address
    match transport source-port
    match transport destination-port
    match routing vrf service
    collect ipv4 dscp
    collect transport tcp flags
    collect interface input
    collect interface output
    collect counter bytes long
    collect counter packets long
    collect timestamp absolute first
    collect timestamp absolute last
    collect application name
    collect flow end-reason
    collect connection initiator
    collect overlay session id input
    collect overlay session id output
    collect connection id long
    collect drop cause id
    collect counter bytes sdwan dropped long
    collect sdwan sla-not-met
    collect sdwan preferred-color-not-met
```

```
    collect sdwan qos-queue-id
collect counter packets sdwan dropped long
```

IPv6 flow format:

```
flow record sdwan_flow_record_ipv6-1667963213662363:
  Description:        flow and application visibility records
  No. of users:       1
  Total field space:  125 bytes
  Fields:
    match ipv6 protocol
    match ipv6 source address
    match ipv6 destination address
    match transport source-port
    match transport destination-port
    match routing vrf service
    collect ipv6 dscp
    collect transport tcp flags
    collect interface input
    collect interface output
    collect counter bytes long
    collect counter packets long
    collect timestamp absolute first
    collect timestamp absolute last
    collect application name
    collect flow end-reason
    collect connection initiator
    collect overlay session id input
    collect overlay session id output
    collect connection id long
    collect drop cause id
    collect counter bytes sdwan dropped long
    collect sdwan sla-not-met
    collect sdwan preferred-color-not-met
    collect sdwan qos-queue-id
    collect counter packets sdwan dropped long
```

```
Device# show flow monitor sdwan_flow_monitor cache
Cache type: Normal (Platform cache)
Cache size: 128000
Current entries: 4
High Watermark: 5
Flows added: 6
Flows aged: 2
- Inactive timeout ( 10 secs) 2
IPV4 SOURCE ADDRESS: 10.20.24.110
IPV4 DESTINATION ADDRESS: 10.20.25.110
TRNS SOURCE PORT: 40254
TRNS DESTINATION PORT: 443
IP VPN ID: 1
IP PROTOCOL: 6
tcp flags: 0x02
interface input: Gi5
interface output: Gi1
counter bytes long: 3966871
counter packets long: 52886
timestamp abs first: 02:07:45.739
timestamp abs last: 02:08:01.840
flow end reason: Not determined
connection initiator: Initiator
interface overlay session id input: 0
interface overlay session id output: 4
connection connection id long: 0xD8F051F000203A22
```

Check the Flows

On the Cisco vEdge devices affected by the Cflowd data policy, various commands let you check the status of the Cflowd flows.

To display information about the flows themselves.

```
vEdge# show app cflowd flows
                                                          TCP                                                                                  TIME
                                   SRC    DEST         IP     CNTRL  ICMP            EGRESS  INGRESS  TOTAL  TOTAL  MIN  MAX                               TO
VPN  SRC IP       DEST IP          PORT   PORT   DSCP  PROTO  BITS   OPCODE  NHOP IP  INTF    INTF     PKTS   BYTES  LEN  LEN  START TIME                   EXPIRE
------------------------------------------------------------------------------------------------------------------------------------------------------------------
1    10.20.24.15  172.16.155.15  46772  13322  0     6      2      0       0.0.0.0  0       0        1      78     78   78   Wed Nov 19 12:31:45 2014  3
1    10.20.24.15  172.16.155.15  46773  13322  0     6      2      0       0.0.0.0  0       0        1      78     78   78   Wed Nov 19 12:31:50 2014  8
1    10.20.24.15  172.16.155.15  46774  13322  0     6      2      0       0.0.0.0  0       0        1      78     78   78   Wed Nov 19 12:31:55 2014  13
1    10.20.24.15  172.16.155.15  46775  13322  0     6      2      0       0.0.0.0  0       0        1      78     78   78   Wed Nov 19 12:32:00 2014  18
1    10.20.24.15  172.16.155.15  46776  13322  0     6      2      0       0.0.0.0  0       0        1      78     78   78   Wed Nov 19 12:32:05 2014  23
1    10.20.24.15  172.16.155.15  46777  13322  0     6      2      0       0.0.0.0  0       0        1      78     78   78   Wed Nov 19 12:32:10 2014  28
1    10.20.24.15  172.16.155.15  46778  13322  0     6      2      0       0.0.0.0  0       0        1      78     78   78   Wed Nov 19 12:32:15 2014  33
1    10.20.24.15  172.16.155.15  46779  13322  0     6      2      0       0.0.0.0  0       0        1      78     78   78   Wed Nov 19 12:32:19 2014  38
1    10.20.24.15  172.16.155.15  46780  13322  0     6      2      0       0.0.0.0  0       0        1      78     78   78   Wed Nov 19 12:32:25 2014  43
1    10.20.24.15  172.16.155.15  46781  13322  0     6      2      0       0.0.0.0  0       0        1      78     78   78   Wed Nov 19 12:32:30 2014  48
1    10.20.24.15  172.16.155.15  46782  13322  0     6      2      0       0.0.0.0  0       0        1      78     78   78   Wed Nov 19 12:32:35 2014  53
1    10.20.24.15  172.16.155.15  46783  13322  0     6      2      0       0.0.0.0  0       0        1      78     78   78   Wed Nov 19 12:32:40 2014  58
```

To quickly get a count of the number of flows.

```
vEdge# show app cflowd flow-count
VPN  count
------------
1    12
```

To display flow statistics.

```
vEdge# show app cflowd statistics

      data_packets            :      0
      template_packets        :      0
      total-packets           :      0
      flow-refresh            :      123
      flow-ageout             :      117
      flow-end-detected       :      0
      flow-end-forced         :      0
```

The following commands show information about the Cflowd collectors and the Cflowd template information that is sent to the collector.

```
vEdge# show app cflowd collector

VPN  COLLECTOR IP   COLLECTOR  CONNECTION             IPFIX    CONNECTION  TEMPLATE  DATA
ID   ADDRESS        PORT       STATE       PROTOCOL   VERSION  RETRY       PACKETS   PACKETS
-----------------------------------------------------------------------------------------
1    172.16.155.15  13322      false       TCP        10       133         0         0

vEdge# show app cflowd template
app cflowd template name test-cflowd-template
app cflowd template flow-active-timeout 30
app cflowd template flow-inactive-timeout 60
app cflowd template template-refresh 90
```

FNF IPv6 Configuration Example for IPv6 traffic

The following example shows the centralized policy configuration with Cflowd for IPv6 traffic:

```
policy
 data-policy _vpn_1_accept_cflowd_vpn_1
  vpn-list vpn_1
   sequence 102
    match
     source-ipv6      2001:DB8:0:/32
     destination-ipv6 2001:DB8:1:/32
    !
```

```
    action accept
     count  cflowd_ipv6_1187157291
     cflowd
    !
   !
   default-action accept
  !
 !
 cflowd-template cflowd_server
  flow-active-timeout   60
  flow-inactive-timeout 30
  protocol              ipv6
 !
 lists
  vpn-list vpn_1
   vpn 1
  site-list vedge1
   site-id 500
  !

apply-policy
 site-list vedge1
  data-policy _vpn_1_accept_cflowd_vpn_1 all
  cflowd-template cflowd_server
```

FNF Export Spread Configuration Example

The following example shows the configuration for export spreading:

```
Device# show sdwan policy from-vsmart
from-vsmart cflowd-template cflowd
 flow-active-timeout    600
 flow-inactive-timeout  60
 template-refresh       60
 flow-sampling-interval 1
 protocol               ipv4
 customized-ipv4-record-fields
  no collect-tos
  no collect-dscp-output
 collector vpn 0 address 10.0.100.1 port 4739 transport transport_udp
  export-spread
   app-tables   20
   tloc-tables  10
   other-tables 5
```

Configure the Maximum FNF Record Rate for Aggregated Data, Using CLI Commands

Minimum supported releases: Cisco IOS XE Catalyst SD-WAN Release 17.14.1a, Cisco Catalyst SD-WAN Control Components Release 20.14.1

Before You Begin

For more information about using CLI templates, see CLI Add-On Feature Templates and CLI Templates. By default, CLI templates execute commands in global configuration mode.

Configure the Maximum FNF Record Rate

Configure the maximum rate (FNF records per minute) for a device to send aggregated traffic data to Cisco SD-WAN Manager.

policy app-agg-node *max-records-per-minute*

Example

The following configures a device to send a maximum of 1000 FNF records per minute of aggregated traffic data.

```
policy app-agg-node 1000
```

Example

The following restores a device to the default value of sending a maximum of 10000 FNF records per minute of aggregated traffic data.

```
no policy app-agg-node
```

Verify Traffic Flow Monitoring

The following sections provide information about verifying traffic flow monitoring.

Verify Collect Loopback

You can verify the ingress and egress interface output using the following command.

show sdwan app-fwd cflowd flows

The following is a sample output from the **show sdwan app-fwd cflowd flows** using the **flows** keyword.

```
Device#show sdwan app-fwd cflowd flows
app-fwd cflowd flows vpn 1 src-ip 10.10.15.12 dest-ip 10.20.15.12 src-port 0 dest-port 0
dscp 0 ip-proto 1
 tcp-cntrl-bits           24
 icmp-opcode              0
 total-pkts               5
 total-bytes              500
 start-time               "Tue Jun 27 09:21:09 2023"
 egress-intf-name         Loopback1
 ingress-intf-name        GigabitEthernet5
 application              ping
 family                   network-service
 drop-cause               "No Drop"
 drop-octets              0
 drop-packets             0
 sla-not-met              0
 color-not-met            0
 queue-id                 2
 initiator                2
 tos                      0
 dscp-output              0
 sampler-id               0
 fec-d-pkts               0
 fec-r-pkts               0
 pkt-dup-d-pkts-orig      0
 pkt-dup-d-pkts-dup       0
```

```
 pkt-dup-r-pkts           0
 pkt-cxp-d-pkts           0
 category                 0
 service-area             0
 cxp-path-type            0
 region-id                0
 ssl-read-bytes           0
 ssl-written-bytes        0
 ssl-en-read-bytes        0
 ssl-en-written-bytes     0
 ssl-de-read-bytes        0
 ssl-de-written-bytes     0
 ssl-service-type         0
 ssl-traffic-type         0
 ssl-policy-action        0
 appqoe-action            0
 appqoe-sn-ip             0.0.0.0
 appqoe-pass-reason       0
 appqoe-dre-input-bytes   0
 appqoe-dre-input-packets 0
 appqoe-flags             0
```

You can verify the ingress and egress interface output using the following command.

show sdwan app-fwd cflowd table

The following is a sample output from the **show sdwan app-fwd cflowd table** using the **table** keyword.

```
show sdwan app-fwd cflowd flows table
PKT    PKT    PKT   PKT                                                                  SSL
 SSL                                                                                APPQOE  APPQOE

                                                                                                TCP

                                                                 SLA  COLOR
                    FEC   FEC   DUP D  DUP D  DUP   CXP                          CXP
SSL    SSL      EN     SSL EN   DE     SSL DE   SSL      SSL      SSL
 APPQOE  DRE     DRE
                                                              SRC    DEST          IP     CNTRL
ICMP    TOTAL  TOTAL                                 EGRESS INTF       INGRESS INTF
                                                DROP    DROP     NOT  NOT    QUEUE
   DSCP    SAMPLER  D     R     PKTS   PKTS   R     D               SERVICE  PATH  REGION
 READ   WRITTEN  READ   WRITTEN  READ   WRITTEN  SERVICE  TRAFFIC  POLICY  APPQOE  APPQOE
  PASS    INPUT   INPUT    APPQOE
VPN  SRC IP                  DEST IP                  PORT   PORT   DSCP  PROTO  BITS
OPCODE  PKTS   BYTES   START TIME                NAME              NAME
APPLICATION  FAMILY             DROP CAUSE   OCTETS  PACKETS  MET  MET    ID     INITIATOR
  TOS  OUTPUT  ID       PKTS  PKTS  ORIG   DUP    PKTS  PKTS  CATEGORY  AREA     TYPE  ID
     BYTES  BYTES    BYTES  BYTES    BYTES  BYTES    TYPE     TYPE     ACTION  ACTION  SN
IP    REASON  BYTES   PACKETS  FLAGS
----------------
1    10.10.15.11             10.20.20.10             0      0      0     1      24     0
       5      500     Tue Jun 27 09:21:06 2023  Loopback1         GigabitEthernet5  ping
       network-service     No Drop      0       0        0     0      2      2          0
  0       0        0     0      0      0      0     0     0         0         0     0
0      0        0      0        0      0        0        0        0       0       0.0.0.0
 0       0       0        0
0    10.0.5.5                10.0.15.10                58048  22     4     6      24
0       41     1752    Tue Jun 27 09:21:06 2023  internal0/0/rp:0  GigabitEthernet9  unknown
      network-service     No Drop      0       0        0    0      2      0          0
 0       0        0     0     0      0      0     0     0         0         0     0       0
      0        0      0        0      0        0        0        0       0       0.0.0.0
0       0       0        0
1    10.10.15.11             10.20.20.10             0      2048   0     1      24
2048    5      500     Tue Jun 27 09:21:06 2023  GigabitEthernet5  Loopback1         ping
```

```
        network-service      No Drop       0       0        0    0      2      2          0
   0       0        0     0      0      0      0     0      0          0        0      0
 0      0        0      0         0      0        0        0        0       0       0.0.0.0
  0       0       0         0
1    10.10.15.11              10.5.10.15               0      2048   0     1      31
2048    20     960     Tue Jun 27 09:21:06 2023  Null              GigabitEthernet5  ping
        network-service      Ipv4NoRoute  960     20       0    0      2      2          0
   0       0        0     0      0      0      0     0      0          0        0      0
 0      0        0      0         0      0        0        0        0       0       0.0.0.0
  0       0       0         0
1    10.10.15.11              10.20.20.10              50920  4739   0     17     31     0
       473    524768  Tue Jun 27 09:21:06 2023  GigabitEthernet5  internal0/0/rp:0  ipfix
       network-management  No Drop      0       0        0    0      2      1          0
  0       0        0     0      0      0      0     0      0          0        0      0
0      0        0      0         0      0        0        0        0       0       0.0.0.0
 0       0       0         0
0    10.0.5.10                10.0.5.10                 22     58048  48    6      24
0       39     3020    Tue Jun 27 09:21:05 2023  GigabitEthernet9  internal0/0/rp:0  ssh
        terminal             No Drop      0       0        0    0      2      2          0
   0       0        0     0      0      0      0     0      0          0        0      0
 0      0        0      0         0      0        0        0        0       0       0.0.0.0
  0       0       0         0
1    10.10.15.11              10.20.20.10              0      771    48    1      31
771     8      4192    Tue Jun 27 09:21:05 2023  internal0/0/rp:0  GigabitEthernet5  icmp
        network-service      No Drop      0       0        0    0      2      2          0
   0       0        0     0      0      0      0     0      0          0        0      0
 0      0        0      0         0      0        0        0        0       0       0.0.0.0
  0       0       0         0
1    fe40::6044:ff:feb7:c2db  ff01::1:ff00:10          0      34560  0     58     0
34560   6     432     Tue Jun 27 09:20:41 2023  internal0/0/rp:0  GigabitEthernet5  ipv6-icmp
    network-service      No Drop      0       0        0    0      0      2          0    0
       0        0     0      0      0      0     0      0          0        0     0       0
    0        0      0        0      0        0        0        0       0       0.0.0.0  0
      0       0        0
1    10:20:20::10             fe40::6024:ff:feb6:c1db  0      34816  56    58     0
34816   4     288     Tue Jun 27 09:20:41 2023  GigabitEthernet5  internal0/0/rp:0  ipv6-icmp
    network-service      No Drop      0       0        0    0      2      2          0    0
       0        0     0      0      0      0     0      0          0        0     0       0
    0        0      0        0      0        0        0        0       0       0.0.0.0  0
      0       0        0
```

Verify Interface Binding on the Device

You can verify the interface binding on the device using the following command.

show sdwan control local-properties wan-interface-list

The following is a sample output from the **show sdwan control local-properties wan-interface-list** using the **wan-interface-list** keyword.

The command displays:

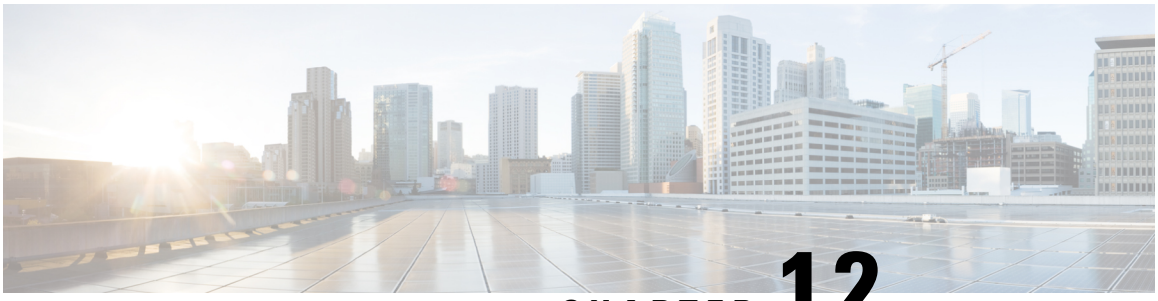
- The physical interface bound to the loopback WAN interface in bind mode.
- Unbind for loopback WAN interface in unbind mode.
- N/A for any other cases.

```
Device#show sdwan control local-properties wan-interface-list
 NAT TYPE: E -- indicates End-point independent mapping
           A -- indicates Address-port dependent mapping
           N -- indicates Not learned
```

```
          Note: Requires minimum two vbonds to learn the NAT type

                              PUBLIC          PUBLIC PRIVATE         PRIVATE
                PRIVATE
MAX    RESTRICT/              LAST         SPI TIME    NAT  VM          BIND
INTERFACE                    IPv4           PORT   IPv4            IPv6
                PORT    VS/VM COLOR            STATE
CNTRL CONTROL/     LR/LB  CONNECTION   REMAINING   TYPE CON REG     INTERFACE

                                                                    STUN
                  PRF IDs
-------------------------------------------------------------------------------------------------------------------
GigabitEthernet1              10.0.10.10      12346  10.0.10.10      ::
                   12346    2/1  lte             up     2      no/yes/no   No/No  0:20:20:27
   0:01:14:20  N    5  Default N/A
GigabitEthernet4              10.0.10.10      12346  10.0.10.10      ::
                   12346    2/0  blue            up     2      no/yes/no   No/No  0:20:20:27
   0:01:14:20  N    5  Default N/A
Loopback1                     1.1.1.1         12366  1.1.1.1         ::
                   12366    2/0  custom1         up     2      no/yes/no   No/No  0:20:20:27
   0:01:14:20  N    5  Default GigabitEthernet1
Loopback2                     2.2.2.2         12406  2.2.2.2         ::
                   12406    2/0  custom2         up     2      no/yes/no   No/No  0:20:20:27
   0:01:14:20  N    5  Default Unbind
```



CHAPTER 12

Forward Error Correction

Note To achieve simplification and consistency, the Cisco SD-WAN solution has been rebranded as Cisco Catalyst SD-WAN. In addition, from Cisco IOS XE SD-WAN Release 17.12.1a and Cisco Catalyst SD-WAN Release 20.12.1, the following component changes are applicable: **Cisco vManage** to **Cisco Catalyst SD-WAN Manager**, **Cisco vAnalytics** to **Cisco Catalyst SD-WAN Analytics**, **Cisco vBond** to **Cisco Catalyst SD-WAN Validator**, **Cisco vSmart** to **Cisco Catalyst SD-WAN Controller**, and **Cisco Controllers** to **Cisco Catalyst SD-WAN Control Components**. See the latest Release Notes for a comprehensive list of all the component brand name changes. While we transition to the new names, some inconsistencies might be present in the documentation set because of a phased approach to the user interface updates of the software product.

Table 34: Feature History

Feature Name	Release Information	Description
Forward Error Correction	Cisco SD-WAN Release 19.1.x	Feature introduced. FEC is a mechanism to recover lost packets on a link by sending extra "parity" packets for every group of 4 packets.

Forward Error Correction (FEC) is a mechanism to recover lost packets on a link by sending extra "parity" packets for every group of 4 packets. As long as the receiver receives a subset of packets in the group (at-least N-1) and the parity packet, up to a single lost packet in the group can be recovered. FEC is supported on Cisco vEdge 1000, 2000, and 5000 routers.

Supported Devices for Forward Error Correction

The forward error correction is supported on all the Cisco IOS XE Catalyst SD-WAN devices.

Configure Forward Error Correction for a Policy

Step 1 From the Cisco SD-WAN Manager menu, choose **Configuration** > **Policies**.

Step 2 Click **Centralized Policy** and then click **Add Policy**.

Step 3 Click **Next**.

Step 4 Click **Next** again and then click **Configure Traffic Rules**.

Step 5 Click **Traffic Data**, and from the **Add Policy** drop-down list, choose **Create New**.

Step 6 Click **Sequence Type**.

Step 7 From the **Add Data Policy** pop-up menu, choose **QoS**.

Step 8 Click **Sequence Rule**.

Step 9 In the **Applications/Application Family List**, choose one or more applications or lists.

Step 10 Click **Accept**.

Step 11 Click **Actions** and click **Loss Correction**.

Step 12 In the **Actions** area, choose one of the following:

- **FEC Adaptive**: Only send FEC information when the loss detected by the system exceeds the packet loss threshold.
- **FEC Always**: Always send FEC information with every transmission.
- **Packet Duplication** check box: Duplicates packets through secondary links to reduce packet loss if one link goes down.

Step 13 Click **Save Match and Actions**.

Step 14 Click **Save Data Policy**.

Step 15 Click **Next** and take these actions to create a centralized policy:

a) Enter a **Name** and a **Description**.
b) Select **Traffic Data Policy**.
c) Choose VPNs and a site list for the policy.
d) Save the policy.

Monitor Forward Error Correction Tunnel Information

Step 1 From the Cisco SD-WAN Manager menu, choose **Monitor** > **Devices**.

Cisco vManage Release 20.6.x and earlier: From the Cisco SD-WAN Manager menu, choose **Monitor** > **Network**.

Step 2 Choose a device group.

Step 3 In the left panel, click **Tunnel**, which displays under WAN.

The WAN tunnel information includes the following:

- A graph that shows the total tunnel loss for the selected tunnels.

- A table that provides the following information for each tunnel endpoint:
  - Name of the tunnel endpoint
  - Communications protocol that the endpoint uses
  - State of the endpoint
  - Jitter, in ms, on the endpoint
  - Packet loss percentage for the endpoint
  - Latency, in ms, on the endpoint
  - Total bytes transmitted from the endpoint
  - Total bytes received by the endpoint
  - Application usage link

Monitor Forward Error Application Family Information

Step 1 From the Cisco SD-WAN Manager menu, choose **Monitor** > **Devices**.

Cisco Catalyst SD-WAN Control Components Release 20.6.1 and earlier: From the Cisco SD-WAN Manager menu, choose **Monitor** > **Network**.

Step 2 Choose a device group.

Step 3 In the left panel, click **SAIE Applications**, which displays under **Applications**.

Note In Cisco Catalyst SD-WAN Control Components Release 20.7.1 and earlier releases, **SAIE Applications** is called **DPI Applications**.

The FEC Recovery Rate application information includes the following:

- A graph for which you can choose the following perspective:
  - Application Usage—Usage of various types of traffic for the selected application families, in KB.

- A table that provides the following for each application family:
  - Name of the application family.
  - Packet Delivery Performance for the application family.

Note If you need to see the packet delivery performance for the selected application family, ensure that packet duplication is enabled. Packet delivery performance is calculated based on the formula as displayed in the Cisco SD-WAN Manager tooltip for the **Packet Delivery Performance** column.

- Traffic usage, in KB, MB, or GB for the selected application family.

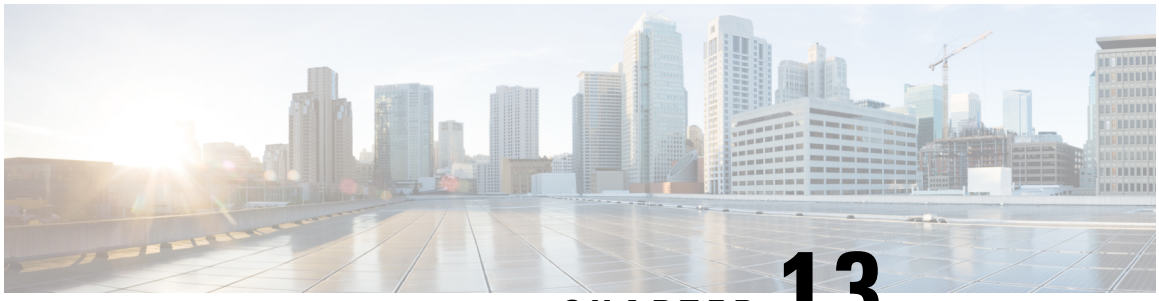
Monitor Forward Error Correction Status Using the CLI

Use the **show sdwan tunnel statistics fec** command to verify the FEC status on a Cisco IOS XE Catalyst SD-WAN device:

```
Device# show sdwan tunnel statistics fec
tunnel stats ipsec 80.80.10.19 80.80.10.25 12346 12366
fec-rx-data-pkts     0
fec-rx-parity-pkts   0
fec-tx-data-pkts     0
fec-tx-parity-pkts   0
fec-reconstruct-pkts 0
fec-capable          true
fec-dynamic          false
tunnel stats ipsec 80.80.10.19 80.80.10.50 12346 12346
fec-rx-data-pkts     122314
fec-rx-parity-pkts   30578
fec-tx-data-pkts     125868
fec-tx-parity-pkts   31467
fec-reconstruct-pkts 3
fec-capable          true
fec-dynamic          false
```

The following table describes the FEC counters related to the output shown in the **show sdwan tunnel statistics fec** command:

Name of Counter	Description
fec-rx-data-pkts	Displays the number of data packets received by the device.
fec-rx-parity-pkts	Displays the number of parity packets received by the device.
fec-tx-data-pkts	Displays the number of data packets sent by the device.
fec-tx-parity-pkts	Displays the number of parity packets sent by the device.
fec-reconstruct-pkts	Displays the number of received packets reconstructed by the device.



CHAPTER 13

Packet Duplication for Noisy Channels

Note To achieve simplification and consistency, the Cisco SD-WAN solution has been rebranded as Cisco Catalyst SD-WAN. In addition, from Cisco IOS XE SD-WAN Release 17.12.1a and Cisco Catalyst SD-WAN Release 20.12.1, the following component changes are applicable: **Cisco vManage** to **Cisco Catalyst SD-WAN Manager**, **Cisco vAnalytics** to **Cisco Catalyst SD-WAN Analytics**, **Cisco vBond** to **Cisco Catalyst SD-WAN Validator**, **Cisco vSmart** to **Cisco Catalyst SD-WAN Controller**, and **Cisco Controllers** to **Cisco Catalyst SD-WAN Control Components**. See the latest Release Notes for a comprehensive list of all the component brand name changes. While we transition to the new names, some inconsistencies might be present in the documentation set because of a phased approach to the user interface updates of the software product.

Table 35: Feature History

Feature Name	Release Information	Description
Packet Duplication for Noisy Channels	Cisco Catalyst SD-WAN Release 19.2.1	This feature helps mitigate packet loss over noisy channels, thereby maintaining high application QoE for voice and video.

Information about Packet Duplication

Cisco vEdge devices use packet duplication to overcome packet loss.

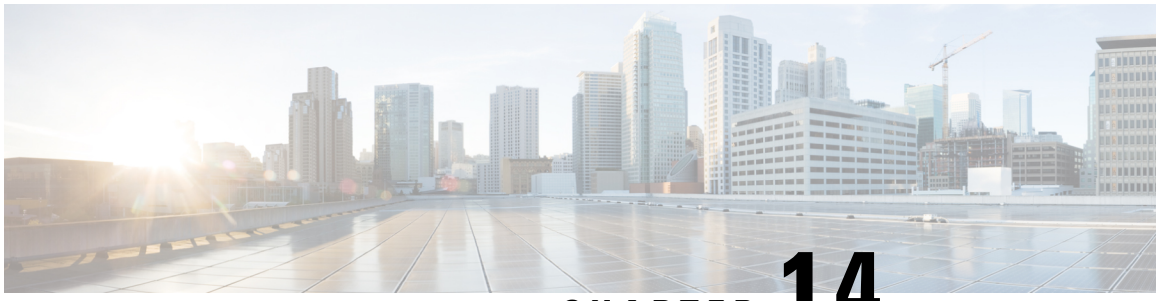
Packet duplication sends copies of packets on alternate available paths to reach Cisco vEdge devices. If one of the packets is lost, a copy of the packet is forwarded to the server. Receiving Cisco vEdge devices discard copies of the packet and forward one packet to the server.

Packet duplication is suitable for edges with multiple access links. Once packet duplication is configured and pushed to your device, you can see the tunnel packet duplication statistics.

Packet duplication cannot work in conjunction with local or remote TLOC in the policy. Data policy or AAR is not configured when specifying the packet duplicated tunnel.

Configure Packet Duplication

1. Select **Configuration** > **Policies**
2. Select **Centralized Policy** at the top of the page and then click **Add Policy**.
3. Click **Next** twice to select Configure Traffic Rules.
4. Select **Traffic Data**, and from the Add Policy drop-down, click **Create New**.
5. Click **Sequence Type** in the left pane.
6. From the Add Data Policy pop-up, select **QoS**.
7. Click **Sequence Rule**.
8. In the **Applications/Application Family List/Data Prefix**, Select one or more applications or lists.
9. Click **Actions** and select **Loss Correction**.
10. In the Actions area, select the **Pack Duplication** option to enable the packet duplication feature.
    - **FEC Adaptive**—Only send Forward Error Correction (FEC) information when the system detects a packet loss.
    - **FEC Always**—Always send FEC information with every transmission.
    - **None**—Use when no loss protection is needed.
    - **Packet Duplication**—Enable when packets need to be duplicated and sent on the next available links to reduce packet loss.
11. Click **Save Match and Actions**.
12. Click **Save Data Policy**.
13. Click **Next** and take these actions to create a Centralized Policy:
    - Enter a Name and a Description.
    - Select **Traffic Data Policy**.
    - Choose **VPNs/site list** for the policy.
    - Save the policy.



CHAPTER 14

Elephant Flow Throttling

The following sections provide information on configuring to throttle Elephant Flow (EF) traffic.

Information About Elephant Flow

Starting from Cisco SD-WAN Release 20.9.1, you can configure to throttle the Elephant Flow (EF) traffic on vEdge2k devices. The traffic flow from both the directions is considered to be the same flow, and is not dependent on the direction. Any flow above the configured rate-threshold in KPPS (Kilo Packets Per Second) is considered as an elephant flow.

You can configure to throttle the elephant flow traffic, when the following conditions occur:

- If the application's performance (voice/video calls/MS Teams) is reduced due to EF.
- If latency has increased due to EF.

Enable EF throttling to:

- Track packet rate for each flow.
- Identify Elephant flows based on the packets per second for the flows.
- Drop packets of the elephant flow, if the CPU utilization and queue threshold exceed the default or configured threshold value.

When a CPU is fully loaded and the input rate exceeds the processing rate, the packet queue builds up, as a result, there is an increase in latency. As the queued up packets become head of line, they block other packets, causing significant latency in other flows that other CPUs must process. As a part of this feature, we identify such elephant flows and drop the packets without processing them, freeing up space for other packets to be processed faster. When this feature is enabled, the elephant flow packets are dropped (whenever there is CPU overload or packet queue builds up) to keep the rest of the flows from being congested.

Restrictions for Elephant Flow Throttling

- Only vEdge2k devices are supported.
- If configured, packet loss is expected for the elephant flows.
- In Cisco SD-WAN Release 20.6.x and later releases, NAT is disabled globally on the router when you execute unpinning of the flows.

Configure Elephant Flow Throttling Using a CLI Template

For more information about using CLI templates, see CLI Templates.

Note By default, CLI templates execute commands in global config mode.

This section provides example CLI configurations to configure elephant flow throttling using a CLI template for Cisco vEdge2k devices:

1. Configure elephant flow in policy mode:

```
vEdge2k# config terminal
vEdge2k(config)# policy
vEdge2k(config-policy)# elephant-flow
vEdge2k(policy-elephant-flow)#
```

2. Enable elephant flow throttling configurations for Cisco vEdge2k:

```
vEdge2k(policy-elephant-flow)# enable
```

3. Specify a rate in Kilo Packets Per Second (KPPS) above which flow is considered as elephant flow:

```
vEdge2k(policy-elephant-flow)# rate-threshold value
```

4. Specify elephant flow queue depth threshold to drop packets:

```
vEdge2k(policy-elephant-flow)# queue-depth threshold-value
```

5. Specify the maximum allowed queue depth to start dropping packets of all flows:

```
vEdge2k(policy-elephant-flow)# max-queue-depth depth
```

6. Define scope for eflow direction.

```
vEdge2k(policy-elephant-flow)# custom-eflow
```

   a. Specify list of sequences. A maximum of eight custom-eflow sequences can be configured. If custom-eflow sequences are not configured, any flow which has more packet rate than elephant-flow-rate-threshold is considered as elephant flow.

```
vEdge2k(policy-custom-eflow)# sequence sequence-value
```

b. Specify match criteria. Even if a single custom-eflow sequence is configured, only flows matching the custom-eflow sequences will be considered as elephant flow. Configure atleast one custom-eflow sequence to consider the matching flow as elephant flow. In the custom-eflow sequences, the match conditions can contain any combination of client-ip/server-ip/protocol. Protocol can be UDP or TCP. Client-IP, and Server-IP can be the required client/server subnet.

```
vEdge2k(config-sequence-sequence-value)# match
[client-ip IPv4 prefix (IP/length)][server-ip IPv4 prefix (IP/length)]
[protocol TCP | UDP]
```

Here's the complete configuration example for elephant flow:

```
config terminal
policy
!
 elephant-flow
!
  enable
  max-queue-depth 25000
  queue-depth 200
  rate-threshold 20
  custom-eflow
!
   sequence 1
!
    match
!
      protocol TCP
      client-ip 10.2.3.0/24
      server-ip 10.2.4.0/24
```

Verify Elephant Flow Throttling Configurations Using the CLI

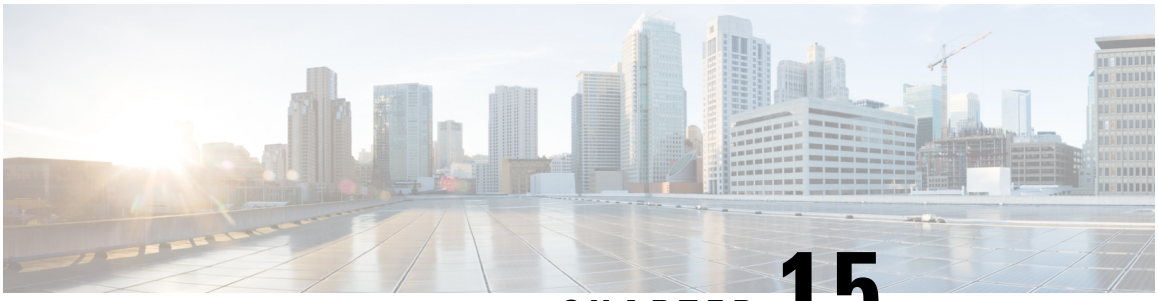
The following is a sample output from the **show policy ef-stats** command:

```
vEdge2k# show policy ef-stats
                          ADD
      ADD    DEL    CUR   SUPER                                                    CUR
CORE  SUPER  SUPER  SUPER BLOCK   ADD   DEL   CUR   SCAN     EF   CUSTOM  HASH     CPU
NUM   BLOCK  BLOCK  BLOCK FAILED  FLOW  FLOW  FLOW  COUNTER  NUM  MATCH   COLLISION USAGE
-------------------------------------------------------------------------------------------
2     1      0      1     0       0     0     0     20523    0    0       0         00.04
3     1      0      1     0       1     0     1     20523    0    0       0         00.01
4     1      0      1     0       0     0     0     20523    0    0       0         00.00
5     1      0      1     0       0     0     0     20523    0    0       0         00.01
6     1      0      1     0       0     0     0     20523    0    0       0         00.01
7     1      0      1     0       0     0     0     20523    0    0       0         00.01
8     1      0      1     0       0     0     0     20523    0    0       0         00.02
9     1      0      1     0       1     0     1     20523    0    0       0         00.02
10    1      0      1     0       0     0     0     20523    0    0       0         00.01
11    1      0      1     0       0     0     0     20523    0    0       0         00.01
12    1      0      1     0       0     0     0     20523    0    0       0         00.00
13    1      0      1     0       1     0     1     20523    0    0       0         00.01
14    1      0      1     0       0     0     0     20523    0    0       0         00.01
```

```
15    1       0       1       0       0       0       0       20523   0       0       0           00.01
16    1       0       1       0       0       0       0       20523   0       0       0           00.02
17    1       0       1       0       0       0       0       20523   0       0       0           00.00
18    1       0       1       0       0       0       0       20523   0       0       0           00.01
19    1       0       1       0       0       0       0       20523   0       0       0           00.01
20    1       0       1       0       0       0       0       20523   0       0       0           00.01
```

The following is a sample output from the **show running-config policy elephant-flow** command:

```
vEdge2k# show running-config policy elephant-flow
policy
 elephant-flow
  enable
  max-queue-depth 25000
  queue-depth 200
  rate-threshold 20
  custom-eflow
   sequence 1
    match
     protocol TCP
     client-ip 1.2.3.0/24
     server-ip 1.2.4.0/24
    !
   !
  !
 !
!
```



CHAPTER 15

Service Chaining

Note To achieve simplification and consistency, the Cisco SD-WAN solution has been rebranded as Cisco Catalyst SD-WAN. In addition, from Cisco IOS XE SD-WAN Release 17.12.1a and Cisco Catalyst SD-WAN Release 20.12.1, the following component changes are applicable: **Cisco vManage** to **Cisco Catalyst SD-WAN Manager**, **Cisco vAnalytics** to **Cisco Catalyst SD-WAN Analytics**, **Cisco vBond** to **Cisco Catalyst SD-WAN Validator**, **Cisco vSmart** to **Cisco Catalyst SD-WAN Controller**, and **Cisco Controllers** to **Cisco Catalyst SD-WAN Control Components**. See the latest Release Notes for a comprehensive list of all the component brand name changes. While we transition to the new names, some inconsistencies might be present in the documentation set because of a phased approach to the user interface updates of the software product.

Note Beginning with Cisco IOS XE Catalyst SD-WAN Release 17.13.1a and Cisco Catalyst SD-WAN Manager Release 20.13.1, see Service Insertion for information about service chaining.

Services in the Network

Services such as firewall, load balancer, and intrusion detection and prevention (IDP) are often run within a virtualized environment, and they may physically be centralized in one location or in several locations for redundancy. Services may be internal, cloud based, or external subscriptions. Networks must be able to reroute traffic from any location in the network through such services.

Customers want the ability to internally spawn or externally subscribe to new services on demand—for capacity, redundancy, or simply to select best-of-breed technologies. For example, if a firewall site exceeds its capacity, a customer can spawn a new firewall service at a new location. Supporting this new firewall would require the configuration of policy-based, weighted load distribution to multiple firewalls.

Following are some of the reasons to reroute a traffic flow through a service or chain of services:

- Traffic flow from a less secure region of a network must pass through a service, such as a firewall, or through a chain of services to ensure that it has not been tampered with.
- For a network that consists of multiple VPNs, each representing a function or an organization, traffic between VPNs must traverse through a service, such as a firewall, or through a chain of services. For example, in a campus, interdepartmental traffic might go through a firewall, while intradepartmental traffic might be routed directly.
- Certain traffic flows must traverse a service, such as a load balancer.

Today, the only way to reroute traffic flow is by provisioning every routing node—from the source to the service node to the systems beyond the service node—with a policy route. This is done either by having an operator manually configure each node or by using a provisioning tool that performs the configuration for each node on behalf of the operator. Either way, the process is operationally complex to provision, maintain, and troubleshoot.

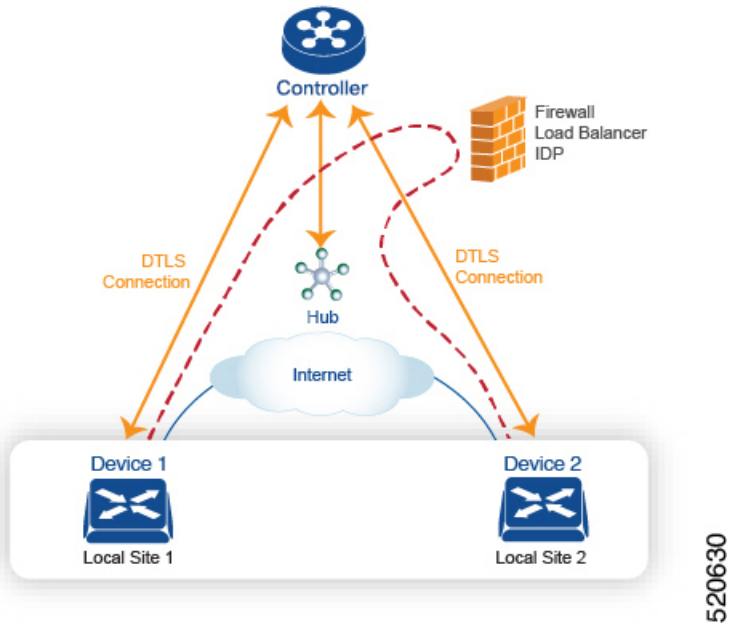
Provisioning Services in the Cisco Catalyst SD-WAN Overlay Network

In the Cisco Catalyst SD-WAN solution, the network operator can enable and orchestrate all service chaining from a central controller, that is, from the Cisco SD-WAN Controller. No configuration or provisioning is required on any of the devices.

The general flow of service chaining in a Cisco Catalyst SD-WAN network is as follows:

- Devices advertise the services available in their branch or campus—such as firewall, IDS, and IDP—to the Cisco SD-WAN Controllers in their domain. Multiple devices can advertise the same services.
- Devices also advertise their OMP routes and TLOCs to the Cisco SD-WAN Controllers.
- For traffic that requires services, the policy on the Cisco SD-WAN Controller changes the next hop for the OMP routes to the service landing point. In this way, the traffic is first processed by the service before being routed to its final destination.

The following figure illustrates how service chaining works in the Cisco Catalyst SD-WAN solution. The network shown has a centralized hub router that is connected to two branches, each with a device. The standard network design implements a control policy such that all traffic from branch site 1 to branch site 2 travels through the hub router. Sitting behind the hub router is a firewall device. So now, assume we want all traffic from site 1 to site 2 to first be processed by the firewall. Traffic from the device at site 1 still flows to the hub router, but instead of sending it directly to site 2, the hub router redirects the traffic to the firewall device. When the firewall completes its processing, it returns all cleared traffic to the hub, which then passes it along to the device at site 2.



Service Route SAFI

The hub and local branch devices advertise the services available in their networks to the Cisco SD-WAN Controllers in its domain using service routes, which are sent by way of OMP using the service route Subsequent Address Family Identifier (SAFI) bits of the OMP NLRI. The Cisco SD-WAN Controllers maintain the service routes in their RIB, and they do not propagate these routes to the devices.

Each service route SAFI has the following attributes:

- VPN ID (vpn-id)—Identifies the VPN that the service belongs to.
- Service ID (svc-id)—Identifies the service being advertised by the service node. The Cisco Catalyst SD-WAN software has the following predefined services:
  - FW, for firewall (maps to svc-id 1)
  - IDS, for Intrusion Detection Systems (maps to svc-id 2)
  - IDP, for Identity Providers (maps to svc-id 3)
  - netsvc1, netsvc2, netsvc3, and netsvc4, which are reserved for custom services (they map to svc-id 4, 5, 6, and 7, respectively)
- Label—For traffic that must traverse a service, the Cisco SD-WAN Controller replaces the label in the OMP route with the service label in order to direct the traffic to that service.
- Originator ID (originator-id)—The IP address of the service node that is advertising the service.
- TLOC—The transport location address of the device that is “hosting” the service.
- Path ID (path-id)—An identifier of the OMP path.

Service Chaining Policy

To route traffic through a service, you provision either a control policy or a data policy on the Cisco SD-WAN Controller. You use a control policy if the match criteria are based on a destination prefix or any of its attributes. You use a data policy if the match criteria include the source address, source port, DSCP value, or destination port of the packet or traffic flow. You can provision the policy directly using the CLI, or it can be pushed from Cisco SD-WAN Manager.

The Cisco SD-WAN Controller maintains OMP routes, TLOC routes, and service routes in its route table. A given OMP route carries a TLOC and the label associated with it. On a Cisco SD-WAN Controller, a policy can be applied that changes the TLOC and its associated label to be that of a service.

Tracking the Health of the Service Chain

Beginning with Cisco SD-WAN Release 20.3.1, Cisco Catalyst SD-WAN periodically probes devices providing network services to test whether they are operational. Tracking the availability of devices in the service chain helps to prevent a null route, which can occur if a policy routes traffic to a service device which is not available. By default, Cisco Catalyst SD-WAN writes the tracking results to a service log, but this can be disabled.

Limitations

- Service insertion over tunnel interface is not supported on Cisco IOS XE Catalyst SD-WAN devices.
- Control policy based service-chain action on locally hosted service-chain is not supported.

- Configuring service-chain and AppQoE on the same device is not supported irrespective of the data-policy or control-policy based actions.
- Tracker for service insertion over tunnel interface is not supported on Cisco vEdge devices.

Configure Service Chaining

Here is the workflow for configuring service chaining for a device managed by Cisco Catalyst SD-WAN:

1. Service devices are accessed through a specific VRF. In the VPN template that corresponds to the VRF for a service device, configure service chaining, specifying the service type and device addresses. By default, the tracking feature adds each service device status update to the service log. You can disable this in the VPN template.
2. Attach the VPN template to the device template for the device managed by Cisco Catalyst SD-WAN.
3. Apply the device template to the device.

Configure Service Chaining Using Cisco SD-WAN Manager

To configure service chaining for a device.

1. In Cisco SD-WAN Manager, create a VPN template.
2. Click**Service**.
3. In the **Service** section, click **New Service** and configure the following:
   - **Service Type**: Select the type of service that the service device is providing.
   - **IP Address**: IP Address is the only working option.
   - **IPv4 Address**: Enter between one and four addresses for the device.
   - **Tracking**: Determines whether the periodic health updates of the service device are recorded in the system log. Default: On

Note Maximum number of services: 8

4. Click **Add**. The service appears in the table of configured services.

CLI Equivalent for Cisco IOS XE Catalyst SD-WAN Devices

The following table shows how configuration of service chaining by CLI corresponds to configuration in Cisco SD-WAN Manager. CLI configuration differs between Cisco IOS XE Catalyst SD-WAN devices and Cisco vEdge devices. The CLI example below is for a Cisco IOS XE Catalyst SD-WAN device.

CLI (Cisco IOS XE Catalyst SD-WAN device)	Cisco SD-WAN Manager
`service firewall vrf 10`	In Cisco SD-WAN Manager, configure service insertion in the VPN template for a specific VRF—VRF 10 in this example. Select the service type from the drop-down —firewall in this example.
`no track-enable` **Note** Default: enabled	When adding a service in the VPN template **Service**, select **On** or **Off** for **Tracking**.
`ipv4 address 10.0.2.1 10.0.2.2`	In the VRF template **Service**, enter one or more IP addresses for the service device providing a specific service.

CLI Example

```
sdwan
  service firewall vrf 10
  ipv4 address 10.0.2.1 10.0.2.2
commit
```

CLI Equivalent for Cisco vEdge Devices

The following table shows how configuration of service chaining by CLI corresponds to configuration in Cisco SD-WAN Manager. CLI configuration differs between Cisco IOS XE Catalyst SD-WAN devices and Cisco vEdge devices. The CLI example below is for a Cisco vEdge device.

CLI (Cisco vEdge device)	Cisco SD-WAN Manager
`vpn 10`	In Cisco SD-WAN Manager, configure service insertion in the VPN template—VPN 10 in this example. Select the service type from the drop-down—firewall in this example.
`service FW address 10.0.2.1`	Select the service type from the drop-down—firewall in this example. Provide one or more addresses for the service device.
`no track-enable` **Note** Default: enabled	When adding a service in the VPN template **Service**, select **On** or **Off** for **Tracking**.

CLI Example

```
vpn 10
  service FW address 10.0.2.1
commit
```

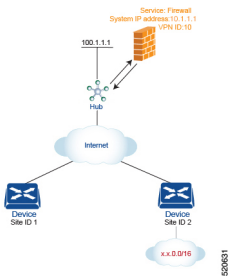
Service Chaining Configuration Examples

Service chaining control policies direct data traffic to service devices that can be located in various places in the network before the traffic is delivered to its destination. For service chaining to work, you configure a centralized control policy on the Cisco SD-WAN Controller, and you configure the service devices themselves

on the device collocated in the same site as the device. To ensure that the services are advertised to the Cisco SD-WAN Controller, the IP address of the service device must resolve locally.

This topic provides examples of configuring service chaining.

Route Intersite Traffic through a Service



A simple example is to route data traffic traveling from one site to another through a service. In this example, we route all traffic traveling from the device at Site 1 to the device at Site 2 through a firewall service that sits behind a hub (whose system IP address is 100.1.1.1). To keep things simple, all devices are in the same VPN.

For this scenario, you configure the following:

- On the hub router, you configure the IP address of the firewall device.
- On the Cisco SD-WAN Controller, you configure a control policy that redirects traffic destined from Site 1 to Site 2 through the firewall service.
- On the Cisco SD-WAN Controller, you apply the control policy to Site 1.

Here is the configuration procedure:

1. On the hub router, provision the firewall service, specifying the IP address of the firewall device. With this configuration, OMP on the hub router advertises one service route to the Cisco SD-WAN Controller. The service route contains a number of properties that identify the location of the firewall, including the TLOC of the hub router and a service label of svc-id-1, which identifies the service type as a firewall. (As mentioned above, before advertising the route, the device ensures that the firewall's IP address can be resolved locally.)

   ```
   vpn 10
     service FW address 10.1.1.1
   ```

2. On the Cisco SD-WAN Controller, configure a control policy that redirects data traffic traveling from Site 1 to Site 2 through the firewall. Then, also on the Cisco SD-WAN Controller, apply this policy to Site 1.

   ```
   policy
     lists
       site-list firewall-sites
         site-id 1
     control-policy firewall-service
       sequence 10
         match route
           site-id 2
         action accept
           set service FW vpn 10
       default-action accept
   ```

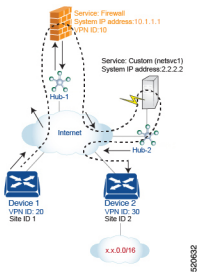
```
apply-policy
  site-list firewall-sites control-policy firewall-service out
```

This policy configuration does the following:

- Create a site list called **firewall-sites** that is referenced in the **apply-policy** command and that enumerates all the sites that this policy applies to. If you later want to scale this policy so that all traffic destined to Site 2 from other sites should also first pass through the firewall, all you need to do is add the additional site IDs to the **firewall-sites** site list. You do not need to change anything in the **control-policy firewall-service** portion of the configuration.
- Define a control policy named **firewall-service**. This policy has one sequence element and the following conditions:
  - Match routes destined for Site 2.
  - If a match occurs, accept the route and redirect it to the firewall service provided by the Hub router, which is located in VPN 10.
  - Accept all nonmatching traffic. That is, accept all traffic not destined for Site 2.
- Apply the policy to the sites listed in **firewall-list**, that is, to Site 1. The Cisco SD-WAN Validator applies the policy in the outbound direction, that is, on routes that it redistributes to Site 1. In these routes:
  - The TLOC is changed from Site 2's TLOC to the hub router's TLOC. This is the TLOC that the Cisco SD-WAN Controller learned from the service route received from the hub router. It is because of the change of TLOC that traffic destined for Site 2 is directed to the hub router
  - The label is changed to svc-id-1, which identifies the firewall service. This label causes the hub router to direct the traffic to the firewall device.

When the hub router receives the traffic, it forwards it to the address 10.1.1.1, which is the system IP address of the firewall. After the firewall has finished processing the traffic, the firewall returns the traffic to the hub router, and this router then forwards it to its final destination, which is Site 2.

Route Inter-VPN Traffic through a Service Chain with One Service per Node



A service chain allows traffic to pass through two or more services before reaching its destination. The example here routes traffic from one VPN to another through services located in a third VPN. The services are located behind different hub routers. Specifically, we want all traffic from device-1 in VPN 20 and that is destined for prefix x.x.0.0/16 in VPN 30 on device-2 to go first through the firewall behind Hub-1 and then through the custom service netsvc1 behind Hub-2 before being sent to its final destination.

For this policy to work:

- VPN 10, VPN 20, and VPN 30 must be connected by an extranet, such as the Internet
- VPN 10 must import routes from VPN 20 and VPN 30. Routes can be selectively imported if necessary.
- VPN 20 must import routes from VPN 30. Routes can be selectively imported if necessary.
- VPN 30 must import routes from VPN 20. Routes can be selectively imported if necessary.

For this scenario, you configure four things:

- You configure the IP address of the firewall device on the Hub-1 router.
- You configure the IP address of the custom service device on the Hub-2 router.
- On the Cisco SD-WAN Controller, you configure a control policy that redirects traffic destined from Site 1 to Site 2 through the firewall device.
- On the Cisco SD-WAN Controller, you configure a second control policy that redirects traffic to the custom service device.

Here is the configuration procedure:

1. Configure the firewall service on Hub-1. With this configuration, OMP on the Hub-1 router advertises a service route to the Cisco SD-WAN Controller. The service route contains a number of properties that identify the location of the firewall, including the TLOC of the hub router and a service label of svc-id-1, which identifies the service type as a firewall.

   ```
   vpn 10
     service fw address 10.1.1.1
   ```

2. Configure the custom service netsvc1 on Hub-2. With this configuration, OMP on the Hub-2 router advertises a service route to the Cisco SD-WAN Controller. The service route contains the TLOC of the Hub-2 and a service label of svc-id-4, which identifies the custom service.

   ```
   vpn 10
     service netsvc1 address 2.2.2.2
   ```

3. Create a control policy on the Cisco SD-WAN Controller for first service in the chain—the firewall—and apply it to Site 1, which is the location of the device-1 router:

   ```
   policy
     lists
       site-list firewall-custom-service-sites
         site-id 1
     control-policy firewall-service
       sequence 10
         match route
           vpn 30
           site-id 2
         action accept
           set service FW
       default-action accept
   apply-policy
     site-list firewall-custom-service-sites control-policy firewall-service out
   ```

   This policy configuration does the following:

   - Create a site list called **firewall-custom-service-sites** that is referenced in the **apply-policy** command and that enumerates all the sites that this policy applies to.
   - Define a control policy named **firewall-service** that has one sequence element and the following conditions:

- Match routes destined for both VPN 30 and Site 2.
- If a match occurs, accept the route and redirect it to a firewall service.
- If a match does not occur, accept the traffic.

- Apply the policy to the sites in the **firewall-custom-service-sites** site list, that is, to Site 1. The Cisco SD-WAN Controller applies this policy in the outbound direction, that is, on routes that it redistributes to Site 1. In these routes:
  - The TLOC is changed from Site 2's TLOC to the Hub-1 router's TLOC. This is the TLOC that the Cisco SD-WAN Controller learned from the service route received from the hub. It is because of the change of TLOC that traffic destined for Site 2 is directed to the Hub-1 router.
  - The label is changed to svc-id-1, which identifies the firewall service. This label causes the Hub-1 router to direct the traffic to the firewall device.

When the Hub-1 router receives the traffic, it forwards it to the address 10.1.1.1, which is the system IP address of the firewall. After the firewall completes processing the traffic, it returns the traffic to the Hub-1 router, which, because of the policy defined in the next step, forwards it to the Hub-2 router.

4. Create a control policy on the Cisco SD-WAN Controller for the second service in the chain, which is the custom service, and apply it to the site of the Hub-1 router:

```
policy
  site-list custom-service
    site-id 3
  control-policy netsvc1-service
    sequence 10
      match route
        vpn 30
        site-id 2
      action accept
        set service netsvc1
    default-action accept
apply-policy
  site-list custom-service control-policy netsvc1-service out
```

This policy configuration does the following:

- Create a site list called **custom-service** that is referenced in the **apply-policy** command and that enumerates all the sites that this policy applies to.
- Define a control policy named **netsvc1-service** that has one sequence element and the following conditions:
  - Match routes destined for both VPN 30 and Site 2.
  - If a match occurs, accept the route and redirect it to the custom service.
  - If a match does not occur, accept the traffic.

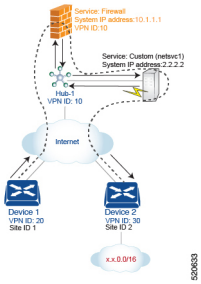
- Apply the policy to the sites in the **custom-service** list, that is, to Site 3. The Cisco SD-WAN Controller applies this policy in the outbound direction, that is, on routes that it redistributes to Site 3. In these routes:
  - The TLOC is changed from Site 2's TLOC to the Hub-2 router's TLOC. This is the TLOC that the Cisco SD-WAN Controller learned from the service route received from the Hub-2 router.

It is because of the change of TLOC that traffic destined for Site 2 is directed to the Hub-2 router.

- The label is changed to svc-id-4, which identifies the custom service. This label causes the Hub-2 to direct the traffic to the device that is hosting the custom service

When the Hub-2 routers receives the traffic, it forwards it to the address 2.2.2.2, which is the system IP address of the device hosting the custom service. After the traffic has been processed, it is returned to the Hub-2 router, which then forwards it to its final destination, Site 2.

Route Inter-VPN Traffic through a Service Chain with Multiple Services per Node



If a service chain has more than one service that is connected to the same node, that is, both services are behind the same device, you use a combination of control policy and data policy to create the desired service chain. The example here is similar to the one in the previous section, but instead has a firewall and a custom service (netsvc-1) behind a single hub router. Here, we want all data traffic from device-1 in VPN 20 destined for prefix x.x.0.0/16 on device-2 in VPN 30 to first go through the firewall at Hub-1, then through the custom service netsvc1, also at Hub-1, and then to its final destination.

For this policy to work:

- VPN 10, VPN 20, and VPN 30 must be connected by an extranet, such as the Internet.
- VPN 10 must import routes from VPN 20 and VPN 30. Routes can be selectively imported if necessary.
- VPN 20 must import routes from VPN 30. Routes can be selectively imported if necessary.
- VPN 30 must import routes from VPN 20. Routes can be selectively imported if necessary.

For this scenario, you configure the following:

- On the hub router, you configure the firewall and custom services.
- On the Cisco SD-WAN Controller, you configure a control policy that redirects data traffic from Site 1 that is destined to Site 2 through the firewall.
- On the Cisco SD-WAN Controller, you configure a data policy that redirects data traffic to the custom service.

Here is the configuration procedure:

1. On the hub router, configure the firewall and custom services:

```
vpn 10
  service FW address 10.1.1.1
  service netsvc1 address 2.2.2.2
```

With this configuration, OMP on the hub router advertises two service routes to the Cisco SD-WAN Controller, one for the firewall and the second for the custom service netsvc1. Both service routes contain the TLOC of the Hub-1 router and a service label that identifies the type of service. For the firewall service, the label is svc-id-1, and for the custom service, the label is svc-id-4.

2. On the Cisco SD-WAN Controller, configure a control policy controller to reroute traffic destined for VPN 30 (at Site 2) to firewall service that is connected to Hub-1 (at Site 3), and apply this policy to Site 1:

```
policy
  lists
    site-list device-1
      site-id 1
  control-policy firewall-service
    sequence 10
      match route
        vpn 30
      action accept
        set service FW
apply-policy
  site-list device-1 control-policy firewall-service out
```

3. On the Cisco SD-WAN Controller, configure a data policy that redirects, or chains, the data traffic received from the firewall device to the custom service netsvc1. Then apply this policy to Hub-1. This data policy routes packets headed for destinations in the network x.x.0.0/16 to the IP address 2.2.2.2, which is the system IP address of the device hosting the custom service.

```
policy
  lists
    site-list device-2
      site-id 2
    site-list Hub-1
      site-id 3
    prefix-list svc-chain
      ip-prefix x.x.0.0/16
    vpn-list vpn-10
      vpn 10
  data-policy netsvc1-policy
    vpn-list vpn-10
      sequence 1
        match
          ip-destination x.x.0.0/16
        action accept
          set next-hop 2.2.2.2
apply-policy
  site-list Hub-1 data-policy netsvc1-policy from-service
```

Active or Backup Scenario with Service Chaining

When using **set service** action to configure active or backup control policy with **set service** action for service chaining, if total number of available paths (summary of active and standby paths) is more than configured **send-path-limit**, do not set preference directly to routes. Ensure to use **set tloc-list** action to set preferences together with **set service** action. Otherwise, you may see cases where either only active or only backup paths are advertised to a particular spoke router.

For example, in the Cisco SD-WAN Controller OMP table, there are eight active and backup paths. Based on the best-path calculation, the paths are sorted in the following order:

backup1, backup2, backup3, backup4, active1, active2, active3, active4

When **send-path-limit 4** is configured, if you apply the first policy, only the four backup paths are sent. If you apply the second policy, two active and two backup paths are sent.

Example of policy susceptible for failures if **send-path-limit** is lower than total number of active and backup paths:

```
control-policy SET_SERVICE_ACTIVE-BACKUP
  sequence 10
   match route
    prefix-list _AnyIpv4PrefixList
    site-list   HUBS_PRIMARY
    tloc-list   INTERNET_TLOCS
   !
   action accept
    set
     preference 200
     service FW vpn 10
    !
   !
  !
  sequence 20
   match route
    prefix-list _AnyIpv4PrefixList
    site-list   HUBS_SECONDARY
    tloc-list   INTERNET_TLOCS
   !
   action accept
    set
     preference 100
     service FW vpn 10
    !
   !
  !
  default-action accept
!
!
```

Example of the same policy but fixed according to recommendations:

```
policy
lists
  tloc-list HUBS_PRIMARY_INTERNET_TLOCS
   tloc 10.0.0.0 color biz-internet encap ipsec preference 200
   tloc 10.0.0.1 color biz-internet encap ipsec preference 200
  !
  tloc-list HUBS_SECONDARY_INTERNET_TLOCS
   tloc 10.255.255.254 color biz-internet encap ipsec preference 100
   tloc 10.255.255.255 color biz-internet encap ipsec preference 100
  !
!
control-policy SET_SERVICE_ACTIVE-BACKUP_FIXED
  sequence 10
   match route
    prefix-list _AnyIpv4PrefixList
    site-list   HUBS_PRIMARY
    tloc-list   INTERNET_TLOCS
   !
   action accept
    set
     service FW vpn 10 tloc-list HUBS_PRIMARY_INTERNET_TLOCS
    !
   !
  !
  sequence 20
   match route
```

```
    prefix-list _AnyIpv4PrefixList
    site-list   HUBS_SECONDARY
    tloc-list   INTERNET_TLOCS
   !
   action accept
    set
     service FW vpn 10 tloc-list HUBS_SECONDARY_INTERNET_TLOCS
    !
   !
  !
  default-action accept
!
!
```

Monitor Service Chaining

You can monitor different aspects of service chaining on hub and spoke devices.

Note Configuring a service device to operate as part of the service chain is called service insertion.

- On a hub device, view the configured services.
  - From the Cisco SD-WAN Manager menu:

    View the configured services on the **Real Time** monitoring page (**Monitor** > **Devices** > *hub-device* > **Real Time**). For **Device Options**, select **OMP Services**.

    Cisco vManage Release 20.6.x and earlier: View the configured services on the **Real Time** monitoring page (**Monitor** > **Network** > *hub-device* > **Real Time**). For **Device Options**, select **OMP Services**.

- On a spoke device, view the details of the service chain path.
  - **Using Cisco SD-WAN Manager**:

    View the service chain path on the **Traceroute** page (**Monitor** > **Devices** > *spoke-device* > **Troubleshooting** > **Connectivity** > **Trace Route**). Enter the destination IP, VPN, and source interface for the desired path.

    Cisco vManage Release 20.6.x and earlier: View the service chain path on the **Traceroute** page (**Monitor** > **Network** > *spoke-device* > **Troubleshooting** > **Connectivity** > **Trace Route**). Enter the destination IP, VPN, and source interface for the desired path.

  - **Using the CLI**:

    Use the **traceroute** command. For information, see the Cisco Catalyst SD-WAN Command Reference.

Example: View a Service Chain Path Between Two Spoke Devices

The following example shows how to view the path between two spokes before and after adding a service chain between them, using Cisco SD-WAN Manager or the CLI.

For clarity, the example presents a scenario of two spoke devices, a hub device, and a service device providing a firewall service, and shows how to configure the firewall service chain.

Here are the details for each device in the scenario:

Device	Address
Hub, through interface ge0/4	10.20.24.15
Spoke 1	10.0.3.1
Spoke 2	10.0.4.1
Service device (firewall service)	10.20.24.17

Configuration of the three devices:

```
Hub
====
vm5# show running-config  vpn 1
vpn 1
 name ospf_and_bgp_configs
 service FW
  address 10.20.24.17
 exit
 router
  ospf
   router-id 10.100.0.1
   timers spf 200 1000 10000
   redistribute static
   redistribute omp
   area 0
    interface ge0/4
    exit
   exit
  !
 !
 interface ge0/4
  ip address 10.20.24.15/24
  no shutdown
 !
 interface ge0/5
  ip address 10.30.24.15/24
  no shutdown
 !
!


Spoke 1
=======
vpn 1
 name ospf_and_bgp_configs
 interface ge0/1
  ip address 10.0.3.1/24
  no shutdown
 !
!


Spoke2
======
vpn 1
 interface ge0/1
  ip address 10.0.4.1/24
  no shutdown
```

```
 !
!
```

1. **Without Service Insertion**:

   At this point, no service insertion policy has been configured, so executing **traceroute** on Spoke 1 to display the path details to Spoke 2 (10.0.4.1) shows a simple path to Spoke 2:

   → Spoke 2 (10.0.4.1)

   ```
   vm4# traceroute vpn 1 10.0.4.1
   Traceroute  10.0.4.1 in VPN 1
   traceroute to 10.0.4.1 (10.0.4.1), 30 hops max, 60 byte packets
    1  10.0.4.1 (10.0.4.1)  7.447 ms  8.097 ms  8.127 ms
   ```

   Similarly, viewing the Traceroute page in Cisco SD-WAN Manager shows a simple path from Spoke 1 to Spoke 2.

2. **With Service Insertion**:

   The following Cisco SD-WAN Controller policy configures service insertion for a firewall service, using the firewall service device described above.

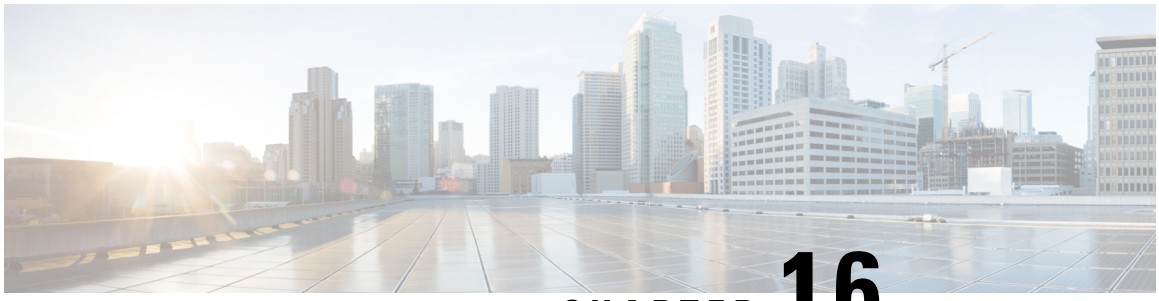
   ```
   vm9# show running-config policy
   policy
    lists
     site-list firewall-sites
      site-id 400
     !
    !
    control-policy firewall-services
     sequence 10
      match route
       site-id 600
      !
      action accept
       set
        service FW vpn 1
       !
      !
     !
     default-action accept
    !
   !
   vm9# show running-config apply-policy
   apply-policy
    site-list firewall-sites
     control-policy firewall-services out
    !
   !
   ```

   After configuring the service insertion, executing **traceroute** on Spoke 1 (10.0.3.1) to display the path details to Spoke 2 (10.0.4.1) shows this path:

   → Hub (10.20.24.15) → Firewall service device (10.20.24.17) → Hub (10.20.24.15) → Spoke 2 (10.0.4.1)

   ```
   Traceroute -m 15 -w 1 -s 10.0.3.1 10.0.4.1 in VPN 1
   traceroute to 10.0.4.1 (10.0.4.1), 15 hops max, 60 byte packets
   1 10.20.24.15 (10.20.24.15) 2.187 ms 2.175 ms 2.240 ms
   2 10.20.24.17 (10.20.24.17) 2.244 ms 2.868 ms 2.873 ms
   3 10.20.24.15 (10.20.24.15) 2.959 ms 4.910 ms 4.996 ms
   4 10.0.4.1 (10.0.4.1) 5.045 ms 5.213 ms 5.247 ms
   ```

Similarly, viewing the **Traceroute** page in Cisco SD-WAN Manager shows each step of the path from Spoke 1 to Spoke 2, through the hub and firewall service device.



CHAPTER 16

Cisco vEdge Device as a NAT Device

To achieve simplification and consistency, the Cisco SD-WAN solution has been rebranded as Cisco Catalyst SD-WAN. In addition, from Cisco IOS XE SD-WAN Release 17.12.1a and Cisco Catalyst SD-WAN Release 20.12.1, the following component changes are applicable: **Cisco vManage** to **Cisco Catalyst SD-WAN Manager**, **Cisco vAnalytics** to **Cisco Catalyst SD-WAN Analytics**, **Cisco vBond** to **Cisco Catalyst SD-WAN Validator**, **Cisco vSmart** to **Cisco Catalyst SD-WAN Controller**, and **Cisco Controllers** to **Cisco Catalyst SD-WAN Control Components**. See the latest Release Notes for a comprehensive list of all the component brand name changes. While we transition to the new names, some inconsistencies might be present in the documentation set because of a phased approach to the user interface updates of the software product.

Cisco vEdge device can act as a NAT device, both on the transport side and on the service side of the router. On the transport side, the NAT functionality allows traffic from a local site to flow directly to the Internet rather than being backhauled to a colo facility that provides NAT services for Internet access. The NAT function is performed as the traffic enters the overlay tunnel to the WAN transport. On the service side, NAT functionality allows traffic from the local site to traverse the NAT before entering the overlay tunnel.

Table 36: Feature History

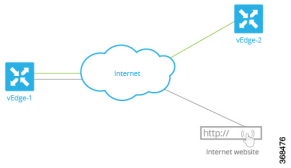
Release	Description
Cisco SD-WAN 19.1	Feature introduced. Cisco vEdge device can act as a NAT device, both on the transport side and on the service side of the router. On the transport side, the NAT functionality allows traffic from a local site to flow directly to the Internet rather than being backhauled to a colo facility that provides NAT services for Internet access.

Cisco vEdge Device as a NAT Device on the Transport Side

To provide users at a local site with direct, secure access to Internet resources, such as websites, you can configure the Cisco vEdge device to function as a Network Address Translation (NAT) device, performing

both address and port translation (NAPT). Enabling NAT allows traffic exiting from a Cisco vEdge device to pass directly to the Internet rather than being backhauled to a colocation facility that provides NAT services for Internet access. Using NAT in this way on a Cisco vEdge device can eliminate traffic "tromboning" and allows for efficient routes, that have shorter distances, between users at the local site and the network-based applications that they use.

The figure below shows the router acting as a NAT device. The vEdge splits its traffic into two flows, which you can think of as two separate tunnels. One traffic flow, shown in green, remains within the overlay network and travels between the two routers in the usual fashion, on the secure IPsec tunnels that form the overlay network. The second traffic stream, shown in grey, is redirected through the Cisco vEdge device's NAT device and then out of the overlay network to a public network.



The NAT functionality on a Cisco vEdge device operates in a standard end-point independent fashion. The NAT software performs both address and port translation (NAPT). It establishes a translation entry between a private address and port pair inside the overlay network and a public address and port outside the overlay network. Once this translation entry is created, the NAT software allows incoming connections from an external host to be established with that private address and port only if that private address and port already established a connection to the external host. That is, an external host can reply to traffic from the private address and port; it cannot initiate a connection.

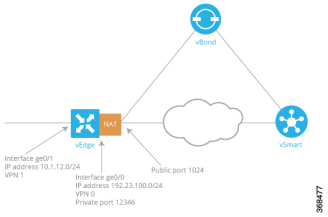
Cisco vEdge devices provide Application Layer Gateway (ALG) FTP support with Network Address Translation – Direct Internet Access (NAT-DIA) and Service NAT. Starting from Cisco SD-WAN Release 20.4.1 Service NAT support is extended to the FTP server. Service NAT was already supported for FTP ALG on the client side for Cisco SD-WAN Release 18.4.x and later releases.

From Cisco SD-WAN Release 20.4.1 the following scenarios are supported:

1. Depending on the location of the FTP server and FTP client:
   - FTP client in private network and FTP server on the internet.
   - FTP server in private network and FP client on the internet.
2. Depending on the type of connectivity present between the FTP client and FTP server:
   - FTP client and server are connected through NAT DIA.
   - FTP client and server are connected through overlay using service side NAT.

Transport-Side NAT Operation

The following figure explains how the NAT functionality on the Cisco vEdge device splits traffic into two flows (or two tunnels), so that some of it remains within the overlay network and some goes directly to the Internet or other public network.



In this figure, the Cisco vEdge device has two interfaces:

- Interface ge0/1 faces the local site and is in VPN 1. Its IP address is 10.1.12.0/24.
- Interface ge0/0 faces the transport cloud and is in VPN 0 (the transport VPN). Its IP address is 192.23.100.0/24, and it uses the default OMP port number, 12346, for overlay network tunnels.

To configure the Cisco vEdge device to act as a NAT device so that some traffic from the router can go directly to a public network, you do three things:

- Enable NAT in the transport VPN (VPN 0) on the WAN-transport–facing interface, which here is ge0/0. All traffic exiting from the Cisco vEdge device, going either to other overlay network sites or to a public network, passes through this interface.
- To direct data traffic from other VPNs to exit from the Cisco vEdge device directly to a public network, enable NAT in those VPNs or ensure that those VPNs have a route to VPN 0.
- On the vCisco Catalyst SD-WAN Controller, create a centralized data policy the redirects the desired data traffic from the non-transport VPN to VPN 0, and then apply that data policy to the non-transport VPN. In this case, we apply the policy to VPN 1.

Once NAT is enabled on the Cisco vEdge device, data traffic affected by the centralized data policy (here, the data traffic from VPN 1) is split into two flows:

- Traffic destined for another Cisco vEdge device in the overlay network remains in VPN 1, and it travels directly through the IPsec data plane tunnel from the source Cisco vEdge device to the destination Cisco vEdge device. This traffic never passes through VPN 0, and therefore it is never touched by NAT.
- Traffic destined for the public network passes from VPN 1 to VPN 0, where it is NATed. During the NAT processing, the source IP address is changed from 10.1.12.0/24 to that of ge0/0, 179.23.100.0/24, and the source port is changed to 1024.

When NAT is enabled, all traffic that passes through VPN 0 is NATed. This includes both the data traffic from VPN 1 that is destined for a public network, and all control traffic, including the traffic required to establish and maintain DTLS control plane tunnels between the Cisco vEdge device and the Cisco Catalyst SD-WAN Controller and between the router and the Cisco Catalyst SD-WAN Validator.

The Cisco Catalyst SD-WAN Validator learns both the public and private addresses of the Cisco vEdge device, and it advertises both address to the Cisco Catalyst SD-WAN Controller. In turn, the Cisco Catalyst SD-WAN Controller advertises both addresses to all the devices in its domain. Each Cisco vEdge device then decides whether to use the public or the private address to communicate with another Cisco vEdge device as follows:

- If the Cisco vEdge device is located at the same site as the other router (that is, if they are both configured with the same overlay network site ID), it communicates using the private address. Because both routers have the same site ID, they are behind the same NAT, and so their communication channels are already secure.

- If the Cisco vEdge device route is at a different site, it communicates with the other router using the public address. Then, the NAT functionality on the Cisco vEdge device translates the public address to the proper private address.

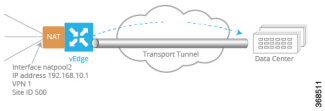
If a Cisco Catalyst SD-WAN Controller connected to a corporate NAT and a NAT-enabled Cisco vEdge device are located at the same physical overlay network site, you must configure them with different Cisco Catalyst SD-WAN site identifiers in order for them to be able to communicate. Similarly, if more than one NAT-enabled Cisco vEdge device is located at the same physical overlay network site, each one must be configured with a different site identifier.

Cisco vEdge Device as a Service-Side NAT Device

On a Cisco vEdge device, you can configure NAT on the service side of the router so that data traffic traverses the NAT before entering the overlay tunnel that is located in the transport VPN. The service-side NAT performs NAT to mask the IP address of data traffic it receives. You can configure both dynamic NAT and 1:1 static NAT on the Cisco vEdge device.

Service-Side NAT Operation

The following figure explains how the Cisco vEdge device provides NAT services on the service side:



In this figure, the Cisco vEdge device has one NAT interface in VPN 1. This interface pools all service-side traffic destined for the NAT interface. The interface name is natpool2, and its IP address is 192.168.10.1. This IP address is the address each packet's IP address is translated to.

To configure the service-side NAT operation on the Cisco vEdge device so that traffic traverses the NAT in VPN 1 before being placed on the transport tunnel towards its destination, you do two things:

- Create a NAT pool interface in VPN 1, the service-side VPN. Here, the NAT pool number is 2.
- To direct data traffic from prefixes within VPN 1 to the service-side NAT, create a centralized data policy on the vSmart controller. In the match condition, specify the prefixes to be NATed. In the action condition, set the desired NAT pool, here, natpool 2. Then apply the data policy to the desired site (here, site 500), and apply it to traffic coming from the service side.

When service-side NAT is enabled, all matching prefixes in VPN 1 are directed to the natpool2 interface. This traffic is NATed, with the NAT swapping out the service-side IP address and replacing it with its NAT pool IP address. The packet then gets forwarded to its destination, here the data center.

Configure Local Internet Exit

To configure a Cisco vEdge device to be an Internet exit point, you enable NAT within a VPN on the Cisco vEdge device, and then you configure a centralized data policy on a Cisco vSmart controller. This policy splits the traffic within the VPN so that some of it is directed towards remote sites within the VPN, and hence remains within the overlay network, and other traffic is directed to the Internet or other destinations outside

the overlay network. It is also possible to configure a Cisco vEdge device to forward data traffic directly to the Internet, by specifying the destination IP prefix.

NAT Configuration Considerations

When configuring a Cisco vEdge device to act as a NAT device, keep the following considerations in mind:

- For a Cisco vEdge device that is acting as a vBond orchestrator, do not enable NAT operation on the interface that is tied to the vBond orchestrator's IP address. If you do so, the orchestrator is placed into a private address space behind the NAT. For the overlay network to function properly, the vBond orchestrator must be in a public address space. You can, however, enable NAT operation on other Cisco vEdge device interfaces.
- When you enable NAT on a Cisco vEdge device, the router NATs all traffic that is sent out through VPN 0. That is, both data traffic and control traffic are NATed.
- The NAT operation on outgoing traffic is performed in VPN 0, which is always only a transport VPN. The router's connection to the Internet is in VPN 0. Performing the NAT operation in VPN 0 avoids the IPsec tunnels that carry data traffic within the overlay network.
- If you configure NAT on multiple interfaces in VPN 0, ECMP is performed among the interfaces.
- When you use NAT—either by configuring it on an interface or by setting it as an action in a centralized data policy—no route lookup is performed. Instead, traffic is forwarded to one of the available NAT default gateways.
- The Cisco vEdge device NAT implementation uses end-point–independent NAT. If your network contains other NAT devices that interact with the Cisco vEdge device NAT, these devices must either perform end-point–independent NAT, or they must be configured with policy rules so that they do not change the port numbers for Cisco Catalyst SD-WAN overlay network destinations.
- When a Cisco vEdge device has two or more NAT interfaces, and hence two or more DIA connections to the internet, by default, data traffic is forwarding on the NAT interfaces using ECMP. To direct data traffic to a specific DIA interface, configure a centralized data policy on the Cisco vSmart controller that sets two actions—**nat** and **local-tloc** color. In the **local-tloc** color action, specify the color of the TLOC that connects to the desired DIA connection.
- Interface IP has to be lesser than NAT range start IP. It is required for IP address of the NAT interface to be lower than the IP addresses used for the IP NAT pool range and static NAT translations. When this requirement is not met, the error is displayed and the configuration will be rejected. When NAT interface IP is higher than the static NAT mapping IP entry, error "Source address is not in the range of the interface IP prefix" displays. The address assigned to the interface IP is in the same subnet as the static mapping IP.

  For example, interface IP is 192.168.1.100/24, and the natpool has a range of 192.168.1.10 to 192.168.1.30 with a static mapping of the translated address 192.168.1.10, configuration will be rejected, error will displayed.

  If the interface IP is lower than the natpool range and static mapping, it allows to commit the configuration with no issues, configuration will be accepted.

  For example, interface IP is 192.168.1.1, and the natpool has a range of 192.168.1.10 to 192.168.1.30 with a static mapping of the translated address 192.168.1.10, configuration will be accepted as the interface IP is lower than the natpool range and static mapping, it allows to commit the configuration with no issues.

Direct Traffic to Exit to the Internet Using Data Policy

To use a centralized data policy to direct traffic from a Cisco vEdge device directly to the Internet, you enable NAT functionality in the WAN VPN or VPNs, and then you create and apply a centralized data policy.

Enable NAT Functionality in the WAN VPN

The first step in setting up Internet exit on a Cisco vEdge device is to configure the router to act as a NAT device. You do this by enabling NAT functionality in VPNs that have interfaces that connect to a WAN transport network. By default, VPN 0 always connects to the WAN transport. Other VPNs in your network might also connect to WANs.

To configure a Cisco vEdge device to act as a NAT device:

1. Enable NAT in the desired VPN:

   ```
   vEdge(config)# vpn vpn-id interface interface-name nat
   ```

2. By default, NAT mappings from the Cisco Catalyst SD-WAN overlay network side of the NAT to the external side of the NAT remain active, and NAT mapping timers are refreshed regularly to keep the mapping operational. To also refresh NAT mappings of packets coming from the external side of the NAT into the overlay network, change the refresh behavior:

   ```
   vEdge(config-nat)# refresh bi-directional
   ```

3. NAT sessions time out after a period of non-use. By default, TCP sessions time out after 60 minutes, and UDP sessions time out after 20 minutes. To change these times:

   ```
   vEdge(config-nat)# tcp-timeout minutes
   vEdge(config-nat)# udp-timeout minutes
   ```

   The times can be from 1 to 65535 minutes.

   The following NAT session timers are fixed, and you cannot modify them:

   - TCP session timeout if no SYN-ACK response is received—5 seconds
   - TCP session timeout if three-way handshaking is not established—10 seconds
   - TCP session timeout after receiving a FIN/RST packet—30 seconds
   - ICMP timeout—6 seconds
   - Other IP timeout—60 seconds

4. By default, the Cisco vEdge device does not receive inbound ICMP error messages. However, NAT uses ICMP to relay error messages across a NAT. To have the router receive the NAT ICMP messages:

   ```
   vEdge(config-nat)# no block-icmp-error
   ```

   In case of a DDoS attack, you might want to return to the default, to again prevent the Cisco vEdge device from receiving inbound ICMP error messages.

Create a Data Policy to Direct Traffic to the Internet Exit

To direct data traffic from a Cisco vEdge device to an Internet exit point, you split the destination of the traffic within a VPN, sending some to remote sites in the VPN and directing the traffic that is destined to the Internet (or other destinations outside the overlay network) to exit directly from the local Cisco vEdge device to the external destination.

To split the traffic, configure a centralized data policy on a Cisco vSmart controller:

1. Configure the source prefix of the data traffic:

```
vSmart(config)# policy data-policy policy-name
vSmart(data-policy)# vpn-list list-name
vSmart(vpn-list)# sequence number
vSmart(sequence)# match source-ip ip-prefix
```

2. Configure the destination of the data traffic, either by IP prefix or by port number:

```
vSmart(sequence)# match destination-ip ip-prefix
vSmart(sequence)# match destination-port port-number
```

3. Direct matching data traffic to the NAT functionality. You can optionally configure a packet counter.

```
vSmart(sequence)# action accept
vSmart(accept)# count counter-name
vSmart(accept)# nat use-vpn 0
```

4. Configure additional sequences, as needed, for other source prefixes and destination prefixes or ports, and for other VPNs.

5. Change the default data policy accept default action from reject to accept. With this configuration, all non-matching data traffic is forwarded to service-side VPNs at remote sites instead of being dropped.

```
vSmart(vpn-list)# default-action accept
```

6. Apply the data policy to particular sites in the overlay network:

```
vSmart(config)# apply-policy site-list list-name data-policy policy-name from-service
```

Direct Traffic To Exit to the Internet Based Only on IP Prefix

You can direct lcal data traffic to exit to the internet based only on the destination IP prefix. To configure this, in the service VPN, forward traffic that is destined towards an internet location to VPN 0, which is the WAN transport VPN:

```
vEdge(config)# vpn vpn-id
vEdge(config-vpn)# ip route prefix vpn 0
```

In the **vpn** command, specify the VPN ID of the service-side VPN from which you are sending the traffic. In the **ip route** command, *prefix* is the IPv4 prefix of the remote destination.The **vpn 0** option configures the software to perform the route lookup in VPN 0 rather than in the service-side VPN. This is done because the service-side VPN cannot resolve the route.

For the traffic redirection to work, in VPN 0, you must enable NAT on the interface associated with the configured prefix:

```
vEdge(config)# vpn 0 interface interface-name nat
```

Here, the interface is the one to use to reach the destination prefix.

The following snippet illustrates the two parts of the configuration:

```
vEdge# show running-config vpn 1
vpn 1
 ...
 ip route 10.1.17.15/32 vpn 0
!
vEdge# show running-config vpn 0
vpn 0
 ...
 interface ge0/1
```

```
  ...
  nat
  !
  no shutdown
 !
!
```

To verify that the redirection is working properly, look at the output of the **show ip routes** command:

```
vEdge# show ip routes
Codes Proto-sub-type:
  IA -> ospf-inter-area,
  E1 -> ospf-external1, E2 -> ospf-external2,
  N1 -> ospf-nssa-external1, N2 -> ospf-nssa-external2,
  e -> bgp-external, i -> bgp-internal
Codes Status flags:
  F -> fib, S -> selected, I -> inactive,
  B -> blackhole, R -> recursive

                                  PROTOCOL  NEXTHOP  NEXTHOP          NEXTHOP
VPN  PREFIX              PROTOCOL   SUB TYPE  IF NAME  ADDR            VPN      TLOC IP          COLOR            ENCAP  STATUS
-------------------------------------------------------------------------------------------------------------------------------
0    0.0.0.0/0           static     -         ge0/0    10.1.15.13      -        -                -                -      F,S
0    10.0.20.0/24        connected  -         ge0/3    -               -        -                -                -      F,S
0    10.0.100.0/24       connected  -         ge0/7    -               -        -                -                -      F,S
0    10.1.15.0/24        connected  -         ge0/0    -               -        -                -                -      F,S
0    10.1.17.0/24        connected  -         ge0/1    -               -        -                -                -      F,S
0    57.0.1.0/24         connected  -         ge0/6    -               -        -                -                -      F,S
0    172.16.255.15/32    connected  -         system   -               -        -                -                -      F,S
1    10.1.17.15/32       nat        -         ge0/1    -               0        -                -                -      F,S
1    10.20.24.0/24       ospf       -         ge0/4    -               -        -                -                -      -
1    10.20.24.0/24       connected  -         ge0/4    -               -        -                -                -      F,S
1    10.20.25.0/24       omp        -         -        -               -        172.16.255.16    lte              ipsec  F,S
1    56.0.1.0/24         connected  -         ge0/5    -               -        -                -                -      F,S
1    60.0.1.0/24         omp        -         -        -               -        172.16.255.16    lte              ipsec  F,S
1    61.0.1.0/24         omp        -         -        -               -        172.16.255.16    lte              ipsec  F,S
512  10.0.1.0/24         connected  -         eth0     -               -        -                -                -      F,S
```

In VPN 1, the prefix 10.1.17.15/32 is associated with the protocol "nat", which reflects the configuration of the **ip route** command in VPN 1. For this prefix, the next-hop interface is **ge0/1**, and the next-hop VPN is VPN 0. This prefix is installed into the route table only if the resolving next hop is over an interface on which NAT is enabled.

The prefix that you configure in the **ip route** represents a route in the specified VPN (the service VPN whose ID you enter in the first command above). To direct traffic to that prefix, you can redistribute it into BGP or OSPF:

```
vEdge(config-vpn)# bgp address-family  address-family  redistribute nat
vEdge(config-vpn)# ospf redistribute nat
```

Track Transport Interface Status

When you enable NAT on a transport interface to allow the local router to forward traffic directly to the internet rather than first forwarding the traffic to a data center router connected to the internet, the router directs data traffic according to the centralized data policy that is applied to that interface, forwarding some traffic directly to the internet (or other external network) and other traffic to other VPNs in the overlay network, including the data center. If the internet or external network becomes unavailable, for example, due to a brownout, the router has no way to learn of this disruption, and it continues to forward traffic based on the policy rules. The result is that traffic that is being forwarded to the internet is silently dropped.

To prevent the internet-bound traffic from being dropped, you can configure the router to track the status of the transport interface and to redirect the traffic to the non-NATed tunnel on the transport interface when the local internet is unavailable. With tracking enabled, the router periodically probes the path to the internet to determine whether it is up. When it detects that the path is down, the router withdraws the NAT route to the internet destination, and reroutes the traffic to the non-NATed tunnel on the interface so that another router in the overlay network can forward the traffic to the internet. The local router continues to periodically check the status of the path to the interface. When it detects that the path is again functioning, the router reinstalls the NAT route to the internet.

To track the transport interface status, you create a global interface tracker, and then you apply it to the transport interface on which NAT is enabled.

To create a transport interface tracker:

```
vEdge(config)# system
vEdge(config-system)#  tracker tracker-name
vEdge(config-tracker)# endpoint-dns-name dns-name
vEdge(config-tracker)# endpoint-ip ip-address
vEdge(config-tracker)# interval seconds
vEdge(config-tracker)# multiplier number
vEdge(config-tracker)# threshold milliseconds
```

The tracker name can be up to 128 lowercase characters.

At a minimum, you must specify the IP address or DNS name of a destination on the internet. This is the destination to which the router sends probes to determine the status of the transport interface. You can configure either one IP address or one DNS name.

By default, a status probe is sent every minute (60 seconds). To modify this value, change the time in the **interval** command to a value from 10 through 600 seconds.

By default, the router waits 300 milliseconds to receive a response from the internet destination. To modify the time to wait for a response, change the time in the **threshold** command to a value from 100 through 1000 milliseconds.

By default, after sending three probes and receiving no responses, the router declares that transport interface is down. To modify the number of retries, change the number in the **multiplier** command to a value from 1 through 10.

You can configure up to eight interface trackers.

To apply a tracker to a transport interface:

```
vEdge(config)# vpn 0
vEdge(vpn)# interface interface-name
vEdge(interface)# tracker tracker-name
```

You can apply only one tracker to an interface.

Configure Service-Side NAT

You can configure both dynamic NAT and 1:1 static NAT on the service side of a router. To do so, you create a NAT pool interface within a service VPN on the router, and then you configure a centralized data policy on the Cisco vSmart controller. This policy directs data traffic with the desired prefixes to the service-side NAT. Finally, you configure either dynamic NAT or static NAT on the desired NAT pool interfaces.

Create a NAT Pool Interface

On the router, you create a NAT pool interface. This interface NATs data traffic that is directed to it and then forwards the traffic towards its destination.

To create a NAT pool interface:

1. In the desired VPN, create the NAT pool interface:

   ```
   vEdge(config-vpn)# interface natpool number
   ```

   The pool can have a number from 1 through 31. You refer to this NAT pool number in the action portion of the centralized data policy that you configure to direct data traffic to the pool. You can configure a maximum of 31 NAT pool interfaces in a VPN.

2. Configure the NAT pool interface's IP address:

   ```
   vEdge(config-natpool)#  ip address prefix/show ip routes length
   ```

   The length of the IP address determines the number of addresses that the router can NAT at the same time. For each NAT pool interface, you can configure a maximum of 250 IP addresses.

3. Enable the interface:

   ```
   vEdge(config-natpool)# no shutdown
   ```

On a NAT pool interface, you can configure only these two commands (**ip address** and **shutdown**/**no shutdown**) and the **nat** command, discussed below. You cannot configure any of the other interface commands.

Here is an example of configuring the NAT pool interface:

```
vEdge# show running-config vpn 1
vpn 1
 interface ge0/4
  ip address 10.20.24.15/24
  no shutdown
 !
 interface ge0/5
  ip address 56.0.1.15/24
  no shutdown
 !
 interface natpool2
  ip address 192.179.10.1/32
  nat
  !
  no shutdown
 !
!
```

To display information about the NAT pool interface, use the **show interface** command:

```
vEdge# show interface vpn 1

                                IF      IF                                                                  TCP
                                ADMIN   OPER    ENCAP  PORT                                 SPEED          MSS                 RX       TX
VPN  INTERFACE  IP ADDRESS      STATUS  STATUS  TYPE   TYPE     MTU   HWADDR                MBPS   DUPLEX  ADJUST  UPTIME      PACKETS  PACKETS
-------------------------------------------------------------------------------------------------------------------------------------------------
1    ge0/4      10.20.24.15/24  Up      Up      null   service  1500  00:0c:29:7d:1e:26     10     full    1420    0:01:24:06  566      565
1    ge0/5      56.0.1.15/24    Up      Up      null   service  1500  00:0c:29:7d:1e:30     10     full    1420    0:01:24:06  26       4
1    natpool2   192.179.10.1/32 Up      Up      null   service  1500  00:00:00:00:00:00     10     full    1420    0:00:40:57  0        0
```

Create a Data Policy To Direct Data Traffic to a Service-Side NAT

To direct data traffic from the service side of the router to the NAT, you create a centralized data policy on the Cisco vSmart controller. In the match condition of the policy, you identify the data traffic that you want to direct to the NAT. One way to do this is to match on the IP prefixes of the data traffic. In the action condition of the policy, you direct the matching traffic to one of the number NAT pools. Finally, you apply the policy to the service side at the desired overlay network sites.

To create a data policy to direct data traffic to a service-side NAT:

1. Configure the lists required for the data policy. You must configure a list of VPN and sites. If you are matching on data prefixes, configure a data prefix list.

   ```
   vSmart(config-policy-lists)# vpn-list list-name
   vSmart(config-policy-vpn-list)# vpn vpn-id
   vSmart(config-policy-lists)# site-list list-name
   vSmart(config-policy-site-list)# site-id site-id
   vSmart(config-policy-lists)# data-prefix-list list-name
   vSmart(config-policy-data-prefix-list)# ip-prefix prefix/length
   ```

2. Configure a data policy:

```
vSmart(config-policy)# data-policy policy-name
vSmart(config-data-policy)# vpn-list list-name
vSmart(config-vpn-list)# sequence number
```

3. Configure the desired match conditions:

```
vSmart(config-sequence)# match condition
```

4. In the action, associate matching data traffic with the desired NAT pool:

```
vSmart(config-sequence)# action accept
vSmart(config-sequence)# nat pool number
```

5. Configure the desired default action for the data policy:

```
vSmart(config-vpn-list)# default-action (accept | reject)
```

6. Apply the policy to the desired sites in the overlay network:

```
vSmart(config)# apply-policy site-list list-name data-policy policy-name from-service
```

Here is an example of configuring the centralized data policy:

```
vSmart# show running-config policy
policy
 data-policy service-side-nat-policy
  vpn-list vpn-1
   sequence 10
    match
     source-data-prefix-list prefixes-to-nat
    !
    action accept
     nat pool 2
    !
   !
   default-action accept
  !
 !
 lists
  vpn-list vpn-1
   vpn 1
  !
  data-prefix-list prefixes-to-nat
   ip-prefix 56.0.1.0/24
  !
  site-list site-500
   site-id 500
  !
 !
!
vSmart# show running-config apply-policy
apply-policy
 site-list site-500
  data-policy service-side-nat-policy from-service
 !
!
```

After you activate the policy, you can see that it has been applied to the router:

```
vEdge# show policy from-vsmart
from-vsmart data-policy service-side-nat-policy
 direction from-service
 vpn-list vpn-1
  sequence 10
```

```
   match
    source-data-prefix-list prefixes-to-nat
   action accept
    nat pool 2
  default-action accept
from-vsmart lists vpn-list vpn-1
 vpn 1
from-vsmart lists data-prefix-list prefixes-to-nat
 ip-prefix 56.0.1.0/24
```

Here is an example of configuring NAT fallback behaviour:

```
vEdge# show policy from-vsmart
from-vsmart data-policy service-side-nat-policy
 direction from-service
 vpn-list vpn-1
 sequence 91
   match
    source-data-prefix-list RFC1918
   action accept
    nat use-vpn 0
    nat fallback
exit
```

Configure Dynamic NAT

By default, when you configure a router to act as a NAT, the router performs dynamic network address translation. In this capacity, the router can perform dynamic NAT for up to 250 IP addresses across NAT pools.

To configure dynamic NAT:

1. In the desired VPN, create the NAT pool interface:

   ```
   vEdge(config-vpn)# interface natpool number
   ```

   The pool can have a number from 1 through 31. You refer to this NAT pool number in the action portion of the centralized data policy that you configure to direct data traffic to the pool. You can configure a maximum of 31 NAT pool interfaces in a VPN.

2. Configure the IP address prefix for the NAT pool interface:

   ```
   vEdge(config-natpool)#  ip address prefix/length
   ```

   The prefix length determines the maximum number of addresses that the router can NAT at the same time. For example, for a /30 prefix length, the router can perform translation on four addresses at a time. For each NAT pool interface, you can configure a maximum of 250 IP addresses.

3. Enable the interface:

   ```
   vEdge(config-natpool)# no shutdown
   ```

4. Enable dynamic NAT:

   ```
   vEdge(config-natpool)# nat
   ```

As mentioned above, the length of the IP address determines the number of IP addresses that the router can NAT at the same time, up to a maximum of 250 across all NAT pools. When all available IP addresses have been used, the router reuses the last IP address multiple times, changing the port number. The port number is chosen at random from the nonreserved port numbers, that is, those port numbers in the range 1024 through 65535. For example, if the IP address is 10.1.17.3/30, the Cisco vEdge device can uniquely NAT four IP

addresses. Let us say that the router maps the fourth IP address to 10.1.20.5, or more specifically to 10.1.20.5:12346 if we include the port number. It would then map the fifth IP address to the same IP address, but with a different port, such as 10.1.20.5:12347. To have the router drop packets when no more IP addresses are available for the translation process, include the following command:

```
vEdge(config)# vpn vpn-id interface  natpool number
vEdge(config-natpool)# no overload
```

Configure Static NAT

You can configure a router acting as a NAT to perform static network address translation (also called 1:1 static NAT) of source IP addresses. You can translate service-side source addresses before sending packets out to the overlay network, and you can translate external addresses before forwarding packets to the service-side network. You can also translate service-side source addresses before sending packets out to another service-side LAN connected to the same router.

For packets originating on the service side of a router, you can statically map the packets' source IP address to another IP address. You do this by creating a NAT pool interface within a service-side VPN. For this interface, you configure a pool of IP addresses to use for network address translation, and then you configure the static address mappings. When the address pool is depleted, you can choose to drop packets that have unmapped source IP addresses. (Dropping these packets is not the default behavior.)

For packets exiting a transport tunnel from a router, you can statically map the packet's source IP address to another IP address, generally to an address that is routable within the service-side network. You configure this in the same way as for NATing packets originating on the service side.

You must create separate NAT pool interfaces to translate source IP addresses for service-side packets and for tunnel packets.

Across all NAT pools, a vEdge router can NAT a maximum of 254 source IP addresses. This is the number of addresses in a /24 prefix, less the .0 and .255 addresses. You cannot configure translation for .0 and .255 addresses.

This section explains how to configure static NAT for translating service-side source IP addresses and for translating external (transport-side) IP addresses. The two procedures are very similar, but we describe them separately for clarity.

Static NATing of Service-Side Addresses

To configure the static NATing of service-side source IP addresses:

1. In the desired VPN, create the NAT pool interface:

   ```
   vEdge(config-vpn)# interface natpool number
   ```

   The pool can have a number from 1 through 31. You refer to this NAT pool number in the action portion of the centralized data policy that you configure to direct data traffic to the pool. You can configure a maximum of 31 NAT pool interfaces in a VPN.

2. Enable the NAT pool interface:

   ```
   vEdge(config-natpool)# no shutdown
   ```

3. Configure the IP address prefix for the NAT pool interface

   ```
   vEdge(config-natpool)# ip address prefix/length
   ```

The prefix length determines the maximum number of source IP addresses that can be NATed in the NAT pool. For example, for a /30 prefix length, a maximum of four source IP addresses can be NATed. For each NAT pool interface, you can configure a maximum of 250 IP addresses.

4. Configure the NAT pool interface to perform network address translation:

   ```
   vEdge(config-natpool)# nat
   ```

5. By default, all IP addresses are translated to an address in the pool of NAT addresses configured in the **ip address** command. The addresses are mapped one to one until the address pool is depleted. Then, the first address is used multiple times, and the port number is changed to a random value between 1024 and 65535. This reuse of the last address is called *overloading*. Overloading effectively implements dynamic NAT.

   To configure static NAT, include the **no overload** command to enforce the mapping of a single source IP address to a single translated IP address:

   ```
   vEdge(config-nat)# no overload
   ```

   With this command, when the maximum number of available IP addresses available to be translated is reached, packets with other IP addresses are dropped.

6. Set the direction in which the NAT pool interface performs static mapping to **inside** to statically translate service-side IP source addresses:

   ```
   vEdge(config-nat)# direction inside
   ```

   Note that the default direction is **inside**.

   A single NAT pool interface can perform static address translation either for service-side source addresses (**direction inside**) or for external source addresses (**direction outside**), but not for both. This means that for a single NAT pool, you can configure only one **direction** command.

7. Define the static address translations for service-side source IP addresses:

   ```
   vEdge(config-nat)# static source-ip ip-address1 translate-ip ip-address2 inside
   ```

   ip-address1 is the source IP address of a device or branch router on the service side of the Cisco vEdge device.

   ip-address2 is the translated source IP address. This is the address that the Cisco vEdge device places in the source field of the packet's IP header when transmitting the packet out the transport network. Because the NAT pool direction is **inside**, this IP address must be in the interface's IP address range. This is the IP address prefix configured in the **ip address** command.

   The **inside** option indicates that it is a service-side, or inside, address that is being statically translated. Note that the **inside** option in the **static** command is different from and independent of the **inside** or **outside** option you specify in the **direction** command. When you are statically NATing service-side addresses, you can statically map both service-side addresses (with a **static**...**inside** command) and transport-side addresses (with a **static**...**outside** command), as described in the next step. The maximum number of service-side source IP addresses that you can statically NAT is equal to the number of addresses available in the interface's prefix range. For example, for a /30 prefix length, you can configure a maximum of four static NAT mappings.

   Once the NAT static address mapping is installed in the router's NAT table, the router can perform source IP address translation in both directions—when a service-side packet is being transmitted into the transport network, and when an external packet (addressed to *ip-address2*) arrives at the router.

8. Define the static address translations for transport-side source IP addresses:

```
vEdge(config-nat)# static source-ip ip-address1 translate-ip ip-address2 outside
```

ip-address1 is the source IP address of an external device or router, that is, of a device at a remote site.

ip-address2 is the translated source IP address. This is the address that the vEdge router places in the source field of the packet's IP header before forwarding the traffic to the service-side network.

The **outside** option indicates that an external IP address is being statically translated. Note that the **outside** option in the **static** command is different from and independent of the **inside** or **outside** option you specify in the **direction** command. When you are statically NATing service-side addresses, you can statically map both service-side addresses (with a **static**...**inside** command) and transport-side addresses (with a **static**...**outside** command), as described in the previous step.

Because the direction of the NAT pool is **inside**, the pool of IP addresses set aside for NATing is used only to NAT service-side source IP addresses. This means that here, you can configure any number of external static address translations.

As a corollary of NATing an external IP address, when a service-side device responds to that external IP address, it simply takes the source IP address from the received packet and places it into the destination IP field in the IP header.

9. Optionally, log the creation and deletion of NAT flows:

```
vEdge(config-nat)# log-translations
```

Static NATing of External Addresses

To configure the static NATing of external source IP addresses:

1. In the desired VPN, create the NAT pool interface:

```
vEdge(config-vpn)# interface natpool number
```

The pool can have a number from 1 through 31. You refer to this NAT pool number in the action portion of the centralized data policy that you configure to direct data traffic to the pool. You can configure a maximum of 31 NAT pool interfaces in a VPN.

2. Enable the NAT pool interface:

```
vEdge(config-natpool)# no shutdown
```

3. Configure the IP address prefix for the NAT pool interface:

```
vEdge(config-natpool)# ip address prefix/length
```

The prefix length determines the maximum number of IP addresses that the router can NAT at the same time in that NAT pool. For example, for a /30 prefix length, the router can perform translation on four addresses at a time. For each NAT pool interface, you can configure a maximum of 250 IP addresses.

4. Configure the NAT pool interface to perform network address translation:

```
vEdge(config-natpool)# nat
```

5. By default, all IP addresses are translated to an address in the pool of NAT addresses configured in the **ip address** command. The addresses are mapped one to one until the address pool is depleted. Then, the last address is used multiple times, and the port number is changed to a random value between 1024 and 65535. This reuse of the last address is called *overloading*. Overloading effectively implements dynamic NAT. To configure static NATing of external addresses, you must include the **no overload** command to enforce the mapping of a single source IP address to a single translated IP address, because the software does not support overloading on the outside NAT pool interface:

```
vEdge(config-nat)# no overload
```

With this command, when the maximum number of available IP addresses available to be translated is reached, packets with other IP addresses are dropped.

6. Set the direction in which the NAT pool interface performs static mapping to **outside** to statically translate external IP source addresses:

   ```
   vEdge(config-nat)# direction outside
   ```

   The default direction is **inside**.

   A single NAT pool interface can perform static address translation either for service-side source addresses (**direction inside**) or for external source addresses (**direction outside**), but not for both. This means that for a single NAT pool, you can configure only one **direction** command.

7. Define the static address translations for external source-IP addresses:

   ```
   vEdge(config-nat)# static source-ip ip-address1 translate-ip ip-address2 outside
   ```

   ip-address1 is the source IP address of a remote device or router on the transport side of the router.

   ip-address2 is the translated source IP address. This is the address that the router places in the source field of the packet's IP header when forwarding the packet into the service-side network. Because the NAT pool direction is **outside**, this IP address must be in the interface's IP address range. This is the IP address prefix configured in the **ip address** command.

   The **outside** option indicates that it is an external, or outside, address that is being statically translated. Note that the **outside** option in the **static** command is different from and independent of the **inside** or **outside** option you specify in the **direction** command. When you are statically NATing external addresses, you can statically map both transport-side addresses (with a **static**...**outside** command) and service-side addresses (with a **static**...**inside** command), as described in the previous step.

   The maximum number of external source IP addresses that you can statically NAT is equal to the number of addresses available in the interface's prefix range. For example, for a /30 prefix length, you can configure a maximum of four static NAT mappings.

   As a corollary of NATing an external IP address, when a service-side device responds to that external IP address, it simply takes the source IP address from the received packet and places it into the destination IP field in the IP header.

Configure Split DNS

When an application-aware routing policy allows a Cisco vEdge device to send application traffic to and receive application traffic from a service VPN, the router performs a Domain Name System (DNS) lookup to determine how to reach a server for the application. If the router does not have a connection to the internet, it sends DNS queries to a router that has such a connection, and that router determines how to reach a server for that application. In a network in which the internet-connect router is in a geographically distant data center, the resolved DNS address might point to a server that is also geographically distant from the site where the service VPN is located.

Because you can configure a Cisco vEdge device to be an internet exit point, it is possible for any router to reach the internet directly to perform DNS lookups. To do this, you create a policy that configures split DNS and that defines, on an application-by-application basis, how to perform DNS lookups.

You configure split DNS with either a centralized data policy or, if you want to apply SLA criteria to the data traffic, an application-aware routing policy. You create these policies on a Cisco vSmart controller, and they are pushed to the Cisco vEdge devices.

CLI Configuration Procedure

Configure Split DNS with a Centralized Data Policy

The following high-level steps show the minimum policy components required to enable split DNS with a centralized data policy:

1. Create one or more lists of overlay network sites to which the centralized data policy is to be applied (in an **apply-policy** command):

   ```
   vSmart(config)# policy
   vSmart(config-policy)# lists site-list list-name
   vSmart(config-lists)# site-id site-id
   ```

   The list can contain as many site IDs as necessary. Include one **site-id** command for each site ID. For contiguous site IDs, you can specify a range of numbers separated with a dash (–).

2. Create lists of applications or application families for which you want to enable split DNS. You refer to these lists in the **match** section of the data policy.

   ```
   vSmart(config)# policy lists
   vSmart(config-lists)# app-list list-name
   vSmart(config-app-list)# (app application-name | app-family family-name)
   ```

3. Create lists VPNs to which the split DNS policy is to be applied (in a **policy data-policy** command):

   ```
   vSmart(config)# policy lists
   vSmart(config-lists)# vpn-list list-name
   vSmart(config-lists)# vpn vpn-id
   ```

4. Create a data policy instance and associate it with a list of VPNs:

   ```
   vSmart(config)# policy data-policy policy-name
   vSmart(config-data-policy)# vpn-list list-name
   ```

5. Create a series of match–action pair sequences:

   ```
   vSmart(config-vpn-list)# sequence number
   ```

   The match–action pairs are evaluated in order, by sequence number, starting with the lowest numbered pair and ending when the route matches the conditions in one of the pairs. Or if no match occurs, the default action is taken (either rejecting the route or accepting it as is).

6. Process the DNS server resolution for the applications or application families contained in an application list. In *list-name*, specify one of the names in a **policy lists app-list** command.

   ```
   vSmart(config-sequence)# match dns-app-list list-name
   ```

7. Configure the match–action pair sequence to process DNS requests (for outbound data traffic) or responses (for inbound data traffic):

   ```
   vSmart(config-sequence)# match dns (request | response)
   ```

8. Accept matching packets, optionally counting and logging them:

   ```
   vSmart(config-sequence)# action  accept [count counter-name] [log]
   ```

9. Enable local internet exit:

```
vSmart(config-sequence)# action accept nat [pool number] [use-vpn 0]
```

10. By default, the DNS servers configured in the VPN in which the policy is applied are used to process DNS lookups for the applications. You can direct DNS requests to a particular DNS server. For a data policy condition that applies to outbound traffic (from the service network), configure the IP address of the DNS server:

    ```
    vSmart(config-sequence)# action  accept redirect-dns ip-address
    ```

    For a data policy condition that applies to inbound traffic (from the tunnel), include the following so that the DNS response can be correctly forwarded back to the service VPN:

    ```
    vSmart(config-sequence)# action accept redirect-dns host
    ```

11. f a route does not match any of the conditions in one of the sequences, it is rejected by default. To accept nonmatching prefixed, configure the default action for the policy:

    ```
    vSmart(config-policy-name)# default-action accept
    ```

12. Apply the policy to one or more sites in the overlay network:

    ```
    vSmart(config)# apply-policy site-list list-name data-policy policy-name (all |
    from-service | from-tunnel)
    ```

Configure Split DNS with an Application-Aware Routing Policy

The following high-level steps show the minimum policy components required to enable split DNS with an application-aware routing policy:

1. Create one or more lists of overlay network sites to which the centralized data policy is to be applied (in an **apply-policy** command):

    ```
    vSmart(config)# policy
    vSmart(config-policy)# lists site-list list-name
    vSmart(config-lists-list-name)# site-id site-id
    ```

    The list can contain as many site IDs as necessary. Include one **site-id** command for each site ID. For contiguous site IDs, you can specify a range of numbers separated with a dash (–).

2. Create SLA classes and traffic characteristics to apply to matching application data traffic:

    ```
    vSmart(config)# policy sla-class sla-class-name
    vSmart(config-sla-class)# jitter milliseconds
    vSmart(config-sla-class)# latency milliseconds
    vSmart(config-sla-class)# loss percentage
    ```

3. Create lists of applications or application families to identify application traffic of interest in the **match** section of the data policy:

    ```
    vSmart(config)# policy lists
    vSmart(config-lists)# app-list list-name
    vSmart(config-app-list)# (app application-name | app-family family-name)
    ```

4. Create lists VPNs to which the split DNS policy is to be applied (in a **policy data-policy** command):

    ```
    vSmart(config)# policy lists
    vSmart(config-lists)# vpn-list list-name
    vSmart(config-lists-list-name)# vpn vpn-id
    ```

5. If you are configuring a logging action, configure how often to log packets to syslog files:

    ```
    vEdge(config)# policy log-frequency number
    ```

6. Create an application-aware routing policy instance and associate it with a list of VPNs:

```
vSmart(config)# policy app-route-policy policy-name
vSmart(config-data-policy-policy-name)# vpn-list list-name
```

7. Create a series of match–pair sequences:

   ```
   vSmart(config-vpn-list)# sequence number
   ```

   The match–action pairs are evaluated in order, by sequence number, starting with the lowest numbered pair and ending when the route matches the conditions in one of the pairs. Or if no match occurs, the default action is taken (either rejecting the route or accepting it as is).

8. Process the DNS server resolution for the applications or application families contained in an application list. In *list-name*, specify one of the names in a **policy lists app-list** command.

   ```
   vSmart(config-sequence-number)# match dns-app-list list-name
   ```

9. Configure the match–action pair sequence to process s DNS requests (for outbound data traffic) or responses (for inbound data traffic):

   ```
   vSmart(config-sequence-number)# match (request | response)
   ```

10. Define the SLA action to take if a match occurs:

    ```
    vSmart(config-sequence)# action sla-class sla-class-name [strict]
    vSmart(config-sequence)# action sla-class sla-class-name [strict] preferred-color
    colors
    vSmart(config-sequence)# action backup-sla-preferred-color colors
    ```

11. For matching packets, optionally count and log them:

    ```
    vSmart(config-sequence)# action count counter-name
    vSmart(config-sequence)# action  log
    ```

12. Enable local internet exit:

    ```
    vSmart(config-sequence-number)# action acccept nat [pool number] [use-vpn 0]
    ```

13. If a packet does not match any of the conditions in one of the sequences, a default action is taken. For application-aware routing policy, the default action is to accept nonmatching traffic and forward it with no consideration of SLA. You can configure the default action so that SLA parameters are applied to nonmatching packets:

    ```
    vSmart(config-policy-name)# default-action sla-class sla-class-name
    ```

14. Apply the policy to one or more sites in the overlay network:

    ```
    vSmart(config)# apply-policy site-list list-name app-route-policy policy-name
    ```

Structural Components of Policy Configuration for Split DNS

Below are the structural components required to configure split DNS on a vSmart controller. The components related to configuring split DNS are explained in the sections below. For an explanation of the data policy and application-aware routing policy components that are not specifically related to split DNS, see *Configure Centralized Data Policy* and *Configure Application-Aware Routing*.

```
policy
  lists
    app-list list-name
      (app application-name | app-family application-family)
    site-list list-name
      site-id site-id
    vpn-list list-name
      vpn-id vpn-id
```

```
 data-policy policy-name
   vpn-list list-name
     sequence number
       match
         dns (request | response)
         dns-app-list list-name
       action accept
         count counter-name
         log
         nat use-vpn 0
         redirect-dns (ip-address | host)
     default-action
       (accept | drop)
apply-policy
  site-list list-name data-policy policy-name (all | from-service | from-tunnel)

policy
  lists
    app-list list-name
      (app application-name | app-family application-family)
    site-list list-name
      site-id site-id
    vpn-list list-name
      vpn-id vpn-id
log-frequency number
sla-class sla-class-name
  jitter milliseconds
  latency milliseconds
  loss percentage
app-route-policy policy-name
   vpn-list list-name
     sequence number
       match
         dns (request | response)
         dns-app-list list-name
       action
         backup-sla-preferred-color colors
         count counter-name
         log
         nat use-vpn 0
         sla-class sla-class-name [strict] [preferred-color colors]
     default-action
       sla-class sla-class-name
apply-policy
  site-list list-name app-route-policy policy-name
```

Lists

A data policy or an application-aware routing policy for split DNS uses the following types of lists to group related items. You configure these lists under the **policy lists** command hierarchy on Cisco vSmart controllers.

Table 37:

List Type	Description	Command
Applications and application families	List of one or more applications or application families running on the subnets connected to the Cisco vEdge device. Each **app-list** can contain either applications or application families, but you cannot mix the two. To configure multiple applications or application families in a single list, include multiple **app** or **app-family** options, specifying one application or application family in each **app** or **app-family**. • *application-name* is the name of an application. The Cisco SD-WAN software supports about 2300 different applications. To list the supported applications, use the ? in the CLI. • *application-family* is the name of an application family. It can be one of the following: **antivirus**, **application-service**, **audio_video**, **authentication**, **behavioral**, **compression**, **database**, **encrypted**, **erp**, **file-server**, **file-transfer**, **forum**, **game**, **instant-messaging**, **mail**, **microsoft-office**, **middleware**, **network-management**, **network-service**, **peer-to-peer**, **printer**, **routing**, **security-service**, **standard**, **telephony**, **terminal**, **thin-client**, **tunneling**, **wap**, **web**, and **webmail**.	**app-list** *list-name* (**app** *application-name* | **app-family** *application-family*)
Sites	List of one or more site identifiers in the overlay network. To configure multiple sites in a single list, include multiple **site-id** options, specifying one site number in each option. You can specify a single site identifier (such as **site-id 1**) or a range of site identifiers (such as **site-id 1-10**).	**site-list** *list-name* **site-id** *site-id*
VPNs	List of one or more VPNs in the overlay network. To configure multiple VPNs in a single list, include multiple **vpn** options, specifying one VPN number in each option. You can specify a single VPN identifier (such as **vpn-id 1**) or a range of VPN identifiers (such as **vpn-id 1-10**).	**vpn-list** *list-name* **vpn** *vpn-id*

In the Cisco vSmart controller configuration, you can create multiple iterations of each type of list. For example, it is common to create multiple site lists and multiple VPN lists so that you can apply data policy to different sites and different customer VPNs across the network.

When you create multiple iterations of a type of list (for example, when you create multiple VPN lists), you can include the same values or overlapping values in more than one of these list. You can do this either on purpose, to meet the design needs of your network, or you can do this accidentally, which might occur when you use ranges to specify values. (You can use ranges to specify data prefixes, site identifiers, and VPNs.) Here are two examples of lists that are configured with ranges and that contain overlapping values:

- **vpn-list list-1 vpn 1-10**

  vpn-list list-2 vpn 6-8

- **site-list list-1 site 1-10**

  site-list list-2 site 5-15

When you configure data policies that contain lists with overlapping values, or when you apply data policies, you must ensure that the lists included in the policies, or included when applying the policies, do not contain overlapping values. To do this, you must manually audit your configurations. The Cisco Catalyst SD-WAN configuration software performs no validation on the contents of lists, on the data policies themselves, or on how the policies are applied to ensure that there are no overlapping values.

If you configure or apply data policies that contain lists with overlapping values to the same site, one policy is applied and the others are ignored. Which policy is applied is a function of the internal behavior of Cisco SD-WAN software when it processes the configuration. This decision is not under user control, so the outcome is not predictable.

VPN Lists

Each data or application-aware policy instance is associated with a VPN list. You configure VPN lists with the **policy data-policy vpn-list** or **policy app-route-policy vpn-list** command. The VPN list you specify must be one that you created with a **policy lists vpn-list** command.

Sequences

Within each VPN list, a data policy or an application-aware policy contains sequences of match–action pairs. The sequences are numbered to set the order in which data traffic is analyzed by the match–action pairs in the policy. You configure sequences with the **policy data-policy vpn-list sequence** or **policy app-aware-policy vpn-list sequence** command.

Each sequence in a policy can contain one **match** command and one **action** command.

Match Parameters

For a data policy or an application-aware routing policy for split DNS, you must the following two match conditions. You configure the match parameters with the **match** command under the **policy data-policy vpn-list sequence** or **policy app-route-policy vpn-list sequence** command hierarchy on Cisco vSmart controllers.

Table 38:

Description	Command	Value or Range
Enable split DNS, to resolve and process DNS requests and responses on an application-by-application basis	**dns-app-list** *list-name*	Name of an **app-list** list. This list specifies the applications whose DNS requests are processed.
Specify the direction in which to process DNS packets	**dns** (**request** | **response**)	To process DNS requests sent by the applications (for outbound DNS queries), specify **dns request**. To process DNS responses returned from DNS servers to the applications, specify **dns response**.

Action Parameters

When data traffic matches the match parameters, the specified action is applied to it. You configure the action parameters with the **action** command under the **policy data-policy vpn-list sequence** or **policy app-route-policy vpn-list sequence** command hierarchy on vSmart controllers.

For application-aware routing policy, the action is to apply an SLA class, which defines the maximum packet latency or maximum packet loss, or both, for DNS traffic related to the application. For information about these action parameters, see *Configure Application-Aware Routing*.

For a centralized data policy that enables split DNS, configure the following actions. You can configure other actions, as described in *Configure Centralized Data Policy*.

Table 39:

Description	Command	Value or Range
Direct data traffic to an Internet exit point on the local router	**nat use-vpn 0**	—
Count matching data packets. Counting packets is optional, but recommended.	**action count** *counter-name*	Name of a counter.
Redirect DNS requests to a particular DNS server. Redirecting requests is optional, but if you do so, you must specify both actions.	**redirect-dns host** **redirect-dns** *ip-address*	For an inbound policy, **redirect-dns host** allows the DNS response to be correctly forwarded back to the requesting service VPN. For an outbound policy, specify the IP address of the DNS server.

Default Action

If a data packet being evaluated does not match any of the match conditions in a policy, a default action is applied. By default, the data packet is dropped. To modify this behavior, include the **policy data-policy vpn-list default-action accept** command.

Applying a Policy

For an application-aware route policy to take effect, you apply it to a list of sites in the overlay network:

```
vSmart(config)# apply-policy site-list list-name app-route-policy  policy-name
```

When you apply the policy, you do not specify a direction (either inbound or outbound). Application-aware routing policy affects only the outbound traffic on the vEdge routers.

For a centralized data policy to take effect, you apply it to a list of sites in the overlay network:

```
vSmart(config)# apply-policy site-list list-name data-policy policy-name (all | from-service
 | from-tunnel)
```

For split DNS to work, you apply a policy to DNS requests originated from a server VPN. If you are specifying the address of a DNS server for a particular application, the *policy-name* data policy must contain a **redirect-dns** *ip-address* action that applies to that application.

```
vSmart(config)# apply-policy policy-name  site-list list-name data-policy policy-name
from-service
```

You also apply a policy to DNS responses being returned from the internet. If you included a **redirect-dns** action in the outbound policy, the *policy-name* data policy must contain a **redirect-dns host** action that applies to the proper application.

```
vSmart(config)# apply-policy policy-name site-list list-name data-policy policy-name
from-tunnel
```

You can apply the same policy to traffic coming from the service VPN and from the tunnel interface between the router and the internet. If the policy specifies use of a specific DNS for a particular application, the policy must contain two sequences for that application, one with a **request-dns** *ip-address* action and the second with a **request-dns host** action.

```
vSmart(config)# apply-policy policy-name site-list list-name data-policy policy-name all
```

Example Configuration

The following example shows a data policy that enables split DNS for a number of applications and counts the DNS traffic:

```
vSmart# show running-config policy
policy
 data-policy split_dns
  vpn-list vpn_1
   sequence 1
    match
     dns-app-list facebook
     dns          request
    !
    action accept
     count facebook_app
    !
   !
   sequence 2
    match
     dns-app-list concur
     dns          request
    !
    action accept
     count concur-app
     nat use-vpn 0
     redirect-dns 75.0.0.1
    !
   !
   sequence 3
    match
     dns-app-list yahoo
    !
    action accept
     count yahoo-app
     nat use-vpn 0
     redirect-dns 75.0.0.1
    !
   !
   sequence 4
    match
     dns-app-list salesforce
    !
    action accept
     count salesforce
     nat use-vpn 0
     redirect-dns 75.0.0.1
    !
   !
   sequence 5
    match
     dns-app-list twitter
     dns          request
    !
    action accept
     count twitter
     nat use-vpn 0
     redirect-dns 75.0.0.1
    !
   !
   sequence 9
    match
     dns-app-list dns_list
     dns          request
```

```
    !
    action accept
     count dns_app_list_count
     nat use-vpn 0
     redirect-dns 75.0.0.1
    !
   !
   sequence 10
    match
     app-list dns_list
    !
    action accept
     count dns_list_count
     nat use-vpn 0
     redirect-dns 75.0.0.1
    !
   !
   default-action accept
  !
 !
 lists
  vpn-list vpn_1
   vpn 1
  !
  app-list concur
   app concur
  !
  app-list dns_list
   app dns
  !
  app-list facebook
   app facebook
  !
  app-list gmail
   app gmail
   app gmail_basic
   app gmail_chat
   app gmail_drive
   app gmail_mobile
  !
  app-list intuit
   app intuit
  !
  app-list salesforce
   app salesforce
  !
  app-list twitter
   app twitter
  !
  app-list yahoo
   app yahoo
  !
  app-list zendesk
   app zendesk
  !
  site-list vedge1
   site-id 500
  !
 !
!
vSmart# show running-config apply-policy
apply-policy
  site-list vedge1 data-policy split_dns all
```

Configure Transport-Side NAT

NAT allows requests coming from the internal (local) network to go out to the external network, but it does not allow request from the external network to come to the internal network. This behavior means that it is impossible for an external device to send a packet to a device on the internal network. It also means that device in the internal network cannot operate as a server with regards to the external network.

To allow requests from the external network to reach internal network devices, you configure the Cisco vEdge device that sits at the edge of the internal network to be a NAT gateway that performs NAT port forwarding (also called *port mapping*). You can also create pools of internal network addresses and dynamically or statically map them to other addresses

Configure NAT Port Forwarding

To allow requests from the external network to reach internal network devices, you configure the Cisco vEdge device that sits at the edge of the internal network to be a NAT gateway that performs NAT port forwarding (also called *port mapping*). With such a configuration, the Cisco vEdge device sends all packets received on a particular port from an external network to a specific device on the internal (local) network.

To configure NAT port forwarding, define one or more port-forwarding rules to send packets received on a particular port from the external network to an internal server:

```
vEdge(config)# vpn 0
vEdge(config-vpn)# interface  ge slot/port
vEdge(config-interface)# nat
vEdge(config-nat)# port-forward port-start port-number1 port-end port-number2 proto (tcp |
 udp) private-vpn vpn-idprivate-ip-addressip-address
```

Use the **port-start** and **port-end** options to define the desired TCP or UDP port or range of ports. *port-number1* must be less than or equal to *port-number2*. To apply port forwarding to a single port, specify the same port number for the starting and ending numbers. When applying port forwarding to a range of ports, the range includes the two port numbers that you specify—*port-number1* and *port-number2*. Packets whose destination port matches the configured port or ports are forwarded to the internal server.

Each rule applies either to TCP or UDP traffic. To match the same ports for both TCP and UDP traffic, configure two rules.

For each rule, specify the private VPN in which the internal server resides and the IP address of the internal server. This VPN is one of the VPN identifiers in the overlay network.

You can create up to 128 rules.

Best Practices for Configuring NAT Port Forwarding

Configuring NAT port forwarding can, in some circumstances, make the Cisco vEdge device vulnerable to brute-force attacks. The following configuration snippet illustrates a case where the router could fall victim to an SSH brute-force attack:

```
system
  aaa
    auth-order local
interface ge0/0
  description    Internet
  ip address 192.168.50.28/28
  nat
   no block-icmp-error
   respond-to-ping
```

```
   port-forward port-start 22 port-end 22 proto tcp
    private-vpn        0
    private-ip-address 192.168.50.28
   !
  !
  tunnel-interface
   encapsulation ipsec
   color public-internet
  !
  no shutdown
 !
```

This configuration creates a port-forwarding rule for TCP port 22, to accept SSH requests from external devices. By itself, this rule provides no opening for brute-force attacks. (As a side note, enabling SSH on a router interface that is connected to the internet is inherently unsafe.) However, problems can arise because of some of the other commands in this configuration:

- **respond-to-ping**—This command allows the Cisco vEdge device to respond to ping requests that are sent from the external network. These ping requests bypass any NAT port-forwarding rules that you have configured. In this configuration, the external network is the Internet, so ping requests can come from anywhere. It is recommended that you do not configure the NAT interface to respond to ping requests. If you need to test reachability, configure this command temporarily and then remove it once the reachability testing is complete.
- **private-vpn 0**—The SSH requests are sent to the WAN transport VPN, VPN 0. A best practice is to forward external traffic to a service-side VPN, that is, to a VPN other than VPN 0 or VPN 512.
- **private-ip-address 192.168.50.28** and **ip address 192.168.50.28/28**—The address of the internal server to which external traffic is being sent is the same as the IP address of the WAN interface. For the private IP address, a best practice is to specify the IP address of a service-side device. If you need to specify a private IP address for one of the interfaces on the Cisco vEdge device, do not use an address in the transport VPN (VPN 0). If you need to use an address in VPN 0, do not use an interface that is connected to the Internet.
- **auth-order local**—This configuration provides only for local authentication, using the credentials configured on the Cisco vEdge device itself. No RADIUS or TACACS server is used to verify the user's SSH login credentials. While this configuration normally does not expose the router to brute-force attacks, here, in the context of the rest of the configuration, it contributes to the router's vulnerability to attack.

Configure NAT Pools

You can configure pools of public IP address and map them to private IP addresses.

First configure a pool of public IP addresses to use for NAT translation:

```
vEdge(config)# vpn 0
vEdge(config-vpn)# interface  interface-name
vEdge(config-interface)# nat
vEdge(config-nat)# natpool range-start ip-address1 range-end ip-address2
```

In the address range, *ip-address1* must be less than or equal to *ip-address2*. The pool can contain a maximum of 32 IP addresses. The addresses must be in the same subnet as the interface's IP address.

Then define the address mapping:

```
vEdge(config)# vpn 0
vEdge(config-vpn)# interface  interface-name
vEdge(config-interface)# nat
```

```
vEdge(config-nat)# static source-ip ip-address1 translate-ip ip-address2 source-vpn vpn-id
 protocol (tcp | udp) source-port number translate-port number
```

In **source-ip**, specify the private source IP address to be NATed. This is the IP address of a device or branch router on the service side of the Cisco vEdge device.

In **translate-ip**, specify the public IP address to map the private source address to. This IP address must be contained in the pool of NAT addresses that you configure with the **natpool** command.

In **source-vpn**, specify the service-side VPN from which the traffic flow is being sent.

In **protocol**, specify the protocol being used to send the traffic flow.

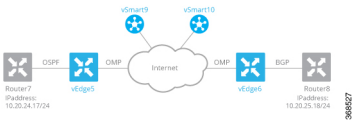
In **source-port** and **translate-port**, specify the number of the source port and the port to which to translate it. The port number can be from 1 through 65535.

You can configure as many static address mappings as there are addresses in the NAT pool.

If you configure a NAT pool but do not configure any static address mappings, NAT translation is done dynamically using the IP addresses in the NAT pool. When a flow terminates, its NATed IP address is released and can be reused.

Service-Side NAT Configuration Example

In this service-side NAT configuration example, two vEdge routers—vEdge5 and vEdge6—are located at two different sites in the overlay network and onnected to each other via the Internet. They are both configured as NATs. Router7 sits in the service side behind vEdge5, and the local network at this site runs OSPF. Router8 sits behind vEdge6 on a network running IBGP.



vEdge5 NATs the source IP address 10.20.24.17, which originates on Router7, translating it to 10.15.1.4. From a NAT perspective on vEdge5, the address 10.20.24.17 is an inside address.

When vEdge6 receives packets with the source IP address 10.15.1.4, it translates the address to 10.16.1.4. From a NAT perspective on vEdge6, the address 10.15.1.4 is an outside address.

In addition, vEdge5 NATs the outside IP source address 10.20.25.18, which originates on Router8 (behind vEdge6), translating it to 10.25.1.1.

The data policies to direct service-side traffic to the NAT are configured on two vSmart controllers, vSmart9 and vSmart10.

By default, OMP advertises all inside NAT pool IP addresses and all static NAT pool IP addresses, so all devices on the overlay network learn these routes automatically. In this example configuration, we configure OSPF and BGP to redistribute outside NAT pool IP addresses. The result is that OSPF on vEdge5 redistributes outside NAT pool IP addresses to its OSPF neighbor, Router7, and BGP redistributes outside NAT pool IP addresses to its BGP neighbor, Router8.

Configure Service-Side NAT on the vEdge Routers

vEdge5 and vEdge6 are vEdge routers at two different sites. They are both connected to the Internet, and they are both are running NAT.

On vEdge5, we configure a NAT pool that can translate four static addresses:

```
vEdge5(config)# vpn 1
vEdge5(config-vpn-1)# interface natpool1
vEdge5(config-natpool1)# ip address 10.15.1.4/30
vEdge5(config-natpool1)# no shutdown
```

Note When you edit the static NAT pool, there might be a previous static NAT pool entry that is retained, causing packet drops to the destination. To avoid this issue, we recommend that you first remove the existing static NAT pool mapping, commit the change, reconfigure a new static NAT pool mapping, and commit again.

With this configuration, the following IP addresses are available for static source IP address mapping: 10.15.1.4, 10.15.1.5, 10.15.1.6, and 10.15.1.7.

We then configure NAT on this interface:

```
vEdge5(config-natpool1)# nat
```

We want to enforce 1:1 static source IP address mapping:

```
vEdge5(config-nat)# no overload
```

If you omit this command, the default behavior is **overload**, which is effectively dynamic NAT. With the default behavior, all IP addresses are translated to an address in the pool of NAT addresses configured in the **ip address** command. The addresses are mapped one to one until the address pool is depleted. Then, the last address is used multiple times, and the port number is changed to a random value between 1024 and 65535. Overloading effectively implements dynamic NAT.

For this NAT pool, we want network address translation to be performed only on inside IP source addresses. Inside address translation is the default behavior. You can also explicitly configure it:

```
vEdge5(config-nat)# direction inside
```

For this example, we configure two NAT mappings. We want to NAT the source IP address 10.20.24.17, which is the IP address of Router7, This address is an inside address; that is, it is an address at the local site. We also want to NAT the source IP address 10.20.25.18, which comes from Router 8, a router behind vEdge6. This is an outside address.

```
vEdge5(config-nat)# static source-ip 10.20.24.17 translate-ip 10.15.1.4 inside
vEdge5(config-nat)# static source-ip 10.20.25.18 translate-ip 10.25.1.1 outside
```

We translate the inside source IP address 10.20.24.17 to 10.15.1.4. Because this NAT pool performs NAT only on inside IP source addresses (**direction inside**), and because 10.20.24.17 is an inside address, the translated address must be one of the addresses in the IP address range 10.15.1.4/30, which is the IP address of the NAT pool interface (configured in the **ip address** command).

We translate the outside address 10.20.25.18 to 10.25.1.1. Because this NAT pool performs NAT only on inside IP source addresses, we can translate outside addresses to any IP address that is routable on the service-side network behind vEdge5.

At vEdge6, we want to translate the source IP address 10.15.1.4, the translated address received from vEdge5, to an address that is routable on the service network behind vEdge6. The NAT pool that we configure on vEdge6 performs NAT only on outside addresses:

```
vEdge6(config)# vpn 1
vEdge6(config-vpn-1)# interface natpool2
vEdge6(config-natpool2)# ip address 10.16.1.4/30
vEdge6(config-natpool2)# no shutdown
vEdge6(config-natpool2)# nat
```

```
vEdge6(config-nat)# direction outside
vEdge6(config-nat)# static source-ip 10.15.1.4 translate-ip 10.16.1.4 outside
vEdge6(config-nat)# no overload
```

Here are the complete configurations for the static NAT pools on the vEdge5 and vEdge6 routers:

```
vEdge5# show running-config vpn 1 interface natpool1
vpn 1
 interface natpool1
  ip address 10.15.1.4/30
  nat
   static source-ip 10.20.24.17 translate-ip 10.15.1.4 inside
   static source-ip 10.20.25.18 translate-ip 10.25.1.1 outside
   no overload
  !
  no shutdown
 !
!

vEdge6# show running-config vpn 1 interface natpool2
vpn 1
 interface natpool2
  ip address 10.16.1.4/30
  nat
   static source-ip 10.15.1.4 translate-ip 10.16.1.4 outside
   direction outside
   no overload
  !
  no shutdown
 !
!
```

Configure Data Policies on vSmart Controllers

To direct service-side traffic to the NAT pool interface, you configure centralized data policies on the vSmart controllers. Our example network has two vSmart controllers, vSmart9 and vSmart10. The data policies must be identical on both of them.

The basic structure of the data policy is to define the match criteria for the packets destined to the NAT interface and then, in the action portion of the policy, to assign or direct the packets to a specific NAT pool. The data policy structure looks like this:

```
For a data-policy
  For a vpn-list
    For a sequence number
      Match specific criteria
      Action accept
        nat pool number
Apply the data-policy to all data traffic
```

In our example, we want a data policy that directs service-side traffic behind the vEdge5 router to the router's NAT pool interface 1 (**interface natpool 1**). Here is one portion of the data policy (specifically, one of the sequences within the policy) that does this, defining the service-side traffic by its source and destination IP addresses:

```
policy
  data-policy accept_nat
    vpn-list vpn_1
      sequence 108
        match
          source-ip      10.1.17.0/24
          destination-ip 10.25.1.0/24
        !
        action accept
```

```
          count nat_108
          nat pool 1
        !
      !
    !
  !
!
apply-policy
  site-list vedge1
    data-policy accept_nat all
  !
!
```

This data policy snippet takes data traffic whose source IP address is in the range 10.1.17.0/24 and destination is 10.25.1.0/24, accepts it, counts it to the file nat_108, and directs it to NAT pool 1. The source IP prefix 10.1.17.0/24 corresponds to traffic originating from Router7. The destination IP prefix 10.25.1.0/24 is an IP address that the vEdge5 router has translated from 10.20.25.18. This latter IP address corresponds to data traffic originating on Router8, so data traffic with a destination IP prefix of 10.25.1.0/24 is delivered to vEdge6 and then to Router8.

The second configuration snippet applies the policy to all traffic passing through the router (**all**).

Configure Route Redistribution by OSPF and BGP

By default, OSPF redistributes routes learned from inside NAT pool prefixes into OSPF. In addition, no routes from any other protocols are distributed into OSPF. The same is true for BGP.

In our configuration, we also want OSPF and BG to redistribute routes learned from outside NAT pool addresses. We also want OSPF to redistribute connected, OMP, and static routes, and we want BGP to redistribute OMP and static routes.

```
vEdge5# show running-config vpn 1 router
vpn 1
 router
  ospf
   redistribute static
   redistribute connected
   redistribute omp
   redistribute natpool-outside
   area 0
    interface ge0/4
     hello-interval 1
     dead-interval  3
    exit
   exit
  !
 !
!

vEdge6# show running-config vpn 1 router
vpn 1
 router
  bgp 1
   timers
    keepalive 1
    holdtime  3
   !
   address-family ipv4-unicast
    redistribute static
    redistribute omp
    redistribute natpool-outside
   !
   neighbor 10.20.25.18
```

```
        no shutdown
        remote-as 2
        timers
         connect-retry          2
         advertisement-interval 1
        !
       !
      !
     !
    !
```

Verify the NAT Configuration

You use two commands to verify that the NAT configuration is operational: **show interface** and **show ip nat interface**.

The **show interface** command output indicates which NAT pool interfaces are configured and provides status about them. The command output for the vEdge5 router shows that NAT pool interface 1 is administratively and operational up. This command output also shows information about the other interfaces configured on vEdge 5.

```
vEdge5# show interface
                                    IF      IF                                                                  TCP
                  AF                ADMIN   OPER    ENCAP                                             SPEED     MSS                 RX       TX
VPN  INTERFACE    TYPE  IP ADDRESS  STATUS  STATUS  TYPE    PORT TYPE  MTU   HWADDR                 MBPS  DUPLEX  ADJUST  UPTIME      PACKETS  PACKETS
-------------------------------------------------------------------------------------------------------------------------------------------------
0    ge0/0        ipv4  10.1.15.15/24  Up      Up      null    transport  1500  00:0c:29:7d:1e:fe  1000  full    1420    0:09:52:43  226385   228332
0    ge0/1        ipv4  10.1.17.15/24  Up      Up      null    service    1500  00:0c:29:7d:1e:08  1000  full    1420    0:07:00:23  1262     10
0    ge0/2        ipv4  -              Down    Down    null    service    1500  00:0c:29:7d:1e:12  -     -       1420    -           0        0
0    ge0/3        ipv4  10.0.20.15/24  Up      Up      null    service    1500  00:0c:29:7d:1e:1c  1000  full    1420    0:07:00:23  1272     10
0    ge0/6        ipv4  57.0.1.15/24   Up      Up      null    service    1500  00:0c:29:7d:1e:3a  1000  full    1420    0:07:00:22  1262     9
0    ge0/7        ipv4  10.0.100.15/24 Up      Up      null    service    1500  00:0c:29:7d:1e:44  1000  full    1420    0:09:52:04  2931     741
0    system       ipv4  172.16.255.15/32 Up    Up      null    loopback   1500  00:00:00:00:00:00  10    full    1420    0:06:59:24  0        0
1    ge0/4        ipv4  10.20.24.15/24 Up      Up      null    service    1500  00:0c:29:7d:1e:26  1000  full    1420    0:07:00:19  26310    25065
1    ge0/5        ipv4  56.0.1.15/24   Up      Up      null    service    1500  00:0c:29:7d:1e:30  1000  full    1420    0:07:00:19  1261     8
1    natpool1     ipv4  -              Up      Up      null    service    1500  00:00:00:00:00:00  10    full    1420    0:05:52:41  0        0
1    natpool7     ipv4  -              Up      Up      null    service    1500  00:00:00:00:00:00  10    full    1420    0:05:52:41  0        0
1    natpool8     ipv4  -              Up      Up      null    service    1500  00:00:00:00:00:00  10    full    1420    0:05:52:41  0        0
1    natpool9     ipv4  -              Up      Up      null    service    1500  00:00:00:00:00:00  10    full    1420    0:05:52:41  0        0
1    natpool10    ipv4  -              Up      Up      null    service    1500  00:00:00:00:00:00  10    full    1420    0:05:52:41  0        0
1    natpool11    ipv4  -              Up      Up      null    service    1500  00:00:00:00:00:00  10    full    1420    0:05:52:41  0        0
1    natpool12    ipv4  -              Up      Up      null    service    1500  00:00:00:00:00:00  10    full    1420    0:05:52:41  0        0
1    natpool13    ipv4  -              Up      Up      null    service    1500  00:00:00:00:00:00  10    full    1420    0:05:52:41  0        0
1    natpool14    ipv4  -              Up      Up      null    service    1500  00:00:00:00:00:00  10    full    1420    0:05:52:41  0        0
1    natpool15    ipv4  -              Up      Up      null    service    1500  00:00:00:00:00:00  10    full    1420    0:05:52:41  0        0
1    natpool16    ipv4  -              Up      Up      null    service    1500  00:00:00:00:00:00  10    full    1420    0:05:52:41  0        0
512  eth0         ipv4  10.0.1.15/24   Up      Up      null    service    1500  00:50:56:00:01:0f  1000  full    0       0:09:52:42  19482    12745
```

Similarly, on the vEdge6 router, we can check that its NAT pool 2 interface is up:

```
vEdge6# show interface
                                    IF      IF                                                                  TCP
                  AF                ADMIN   OPER    ENCAP                                             SPEED     MSS                 RX       TX
VPN  INTERFACE    TYPE  IP ADDRESS  STATUS  STATUS  TYPE    PORT TYPE  MTU   HWADDR                 MBPS  DUPLEX  ADJUST  UPTIME      PACKETS  PACKETS
-------------------------------------------------------------------------------------------------------------------------------------------------
0    ge0/0        ipv4  10.1.16.16/24  Up      Up      null    transport  1500  00:0c:29:d7:63:18  1000  full    1420    0:09:58:47  271786   294577
0    ge0/1        ipv4  10.1.18.16/24  Up      Up      null    service    1500  00:0c:29:d7:63:22  1000  full    1420    0:07:06:18  1274     10
0    ge0/2        ipv4  -              Down    Down    null    service    1500  00:0c:29:d7:63:2c  -     -       1420    -           0        0
0    ge0/3        ipv4  10.0.21.16/24  Up      Up      null    service    1500  00:0c:29:d7:63:36  1000  full    1420    0:07:06:18  1285     9
0    ge0/7        ipv4  10.0.100.16/24 Up      Up      null    service    1500  00:0c:29:d7:63:5e  1000  full    1420    0:09:58:04  2971     746
0    system       ipv4  172.16.255.16/32 Up    Up      null    loopback   1500  00:00:00:00:00:00  10    full    1420    0:07:05:28  0        0
1    ge0/4        ipv4  10.20.25.16/24 Up      Up      null    service    1500  00:0c:29:d7:63:40  1000  full    1420    0:07:06:15  51457    50250
1    ge0/5        ipv4  60.0.1.16/24   Up      Up      null    service    1500  00:0c:29:d7:63:4a  1000  full    1420    0:07:06:15  1273     8
1    ge0/6        ipv4  61.0.1.16/24   Up      Up      null    service    1500  00:0c:29:d7:63:54  1000  full    1420    0:07:06:15  1255     8
1    natpool2     ipv4  -              Up      Up      null    service    1500  00:00:00:00:00:00  10    full    1420    0:05:56:37  0        0
1    natpool12    ipv4  -              Down    Down    null    service    1500  00:00:00:00:00:00  -     -       1420    -           0        0
512  eth0         ipv4  10.0.1.16/24   Up      Up      null    service    1500  00:50:56:00:01:10  1000  full    0       0:09:58:39  17650    11555
```

To display information about the NAT pools themselves, use the **show ip nat interface** command. Here is the command output for the vEdge5 router in tabular format and for vEdge6 in nontabular format:

```
vEdge5# show ip nat interface
                                                                                FILTER  FIB FILTER                  NUMBER IP
VPN  IFNAME     MAP TYPE                FILTER TYPE                             COUNT   COUNT    IP                 POOLS
-------------------------------------------------------------------------------------------------------------------------
1    natpool1   endpoint-independent    address-port-restricted                 0       0        10.15.1.4/30       4
1    natpool7   endpoint-independent    address-port-restricted                 0       0        10.21.26.15/32     1
1    natpool8   endpoint-independent    address-port-restricted                 0       0        10.21.27.15/32     1
1    natpool9   endpoint-independent    address-port-restricted                 0       0        10.21.28.15/32     1
```

```
1     natpool10  endpoint-independent  address-port-restricted  0        0        10.21.29.15/32  1
1     natpool11  endpoint-independent  address-port-restricted  0        0        10.21.30.15/32  1
1     natpool12  endpoint-independent  address-port-restricted  0        0        10.21.31.15/32  1
1     natpool13  endpoint-independent  address-port-restricted  0        0        10.21.32.15/32  1
1     natpool14  endpoint-independent  address-port-restricted  0        0        10.21.33.15/32  1
1     natpool15  endpoint-independent  address-port-restricted  0        0        10.21.34.15/32  1
1     natpool16  endpoint-independent  address-port-restricted  0        0        10.21.35.15/32  1
```

```
vEdge6# show ip nat interface
ip nat interface nat-vpn 1 nat-ifname natpool2
 mapping-type     endpoint-independent
 filter-type      address-port-restricted
 filter-count     0
 fib-filter-count 0
 ip               10.16.1.4/30
```

Verify Routes and Route Redistribution

We configured OSPF and BGP to redistributes routes learned from outside NAT into OSPF and BGP, respectively. (We also configured OSPF and BGP ro redistribute static and OMP routes, and we configured OMP to redistribute routes learned from directly connected devices.)

To see where routes have been learned from, look at the Protocol field in the output of the **show ip routes** command.

Looking on the vEdge5 router, we see that OSPF has redistributed 10.15.1.4/30, a route learned from an inside NAT (these routes are redistributed by default) and 10.25.1.1/32, a route learned from an outside NAT. The vEdge5 router translates the IP address 10.25.1.1 from 10.20.25.18. Both these routes have a next-hop interface of natpool1, which is the NAT pool we configured to run static NAT.

```
vEdge5# show ip routes
Codes Proto-sub-type:
  IA -> ospf-inter-area,
  E1 -> ospf-external1, E2 -> ospf-external2,
  N1 -> ospf-nssa-external1, N2 -> ospf-nssa-external2,
  e -> bgp-external, i -> bgp-internal
Codes Status flags:
  F -> fib, S -> selected, I -> inactive,
  B -> blackhole, R -> recursive

                                      PROTOCOL  NEXTHOP     NEXTHOP          NEXTHOP
VPN    PREFIX              PROTOCOL   SUB TYPE  IF NAME     ADDR             VPN      TLOC IP          COLOR            ENCAP  STATUS
-------------------------------------------------------------------------------------------------------------------------------------
0      0.0.0.0/0           static     -         ge0/0       10.1.15.13       -        -                -                -      F,S
0      10.0.20.0/24        connected  -         ge0/3       -                -        -                -                -      F,S
0      10.0.100.0/24       connected  -         ge0/7       -                -        -                -                -      F,S
0      10.1.15.0/24        connected  -         ge0/0       -                -        -                -                -      F,S
0      10.1.17.0/24        connected  -         ge0/1       -                -        -                -                -      F,S
0      57.0.1.0/24         connected  -         ge0/6       -                -        -                -                -      F,S
0      172.16.255.15/32    connected  -         system      -                -        -                -                -      F,S
1      2.2.0.0/16          static     -         -           -                -        -                -                -      B,F,S
1      4.4.4.4/32          static     -         -           -                -        -                -                -      B,F,S
1      9.0.0.0/8           omp        -         -           -                -        172.16.255.16    lte              ipsec  F,S
1      10.1.17.0/24        static     -         ge0/4       10.20.24.17      -        -                -                -      F,S
1      10.1.18.0/24        omp        -         -           -                -        172.16.255.16    lte              ipsec  F,S
1      10.2.2.0/24         omp        -         -           -                -        172.16.255.11    lte              ipsec  F,S
1      10.2.3.0/24         omp        -         -           -                -        172.16.255.21    lte              ipsec  F,S
1      10.15.1.4/30        natpool-inside  -    natpool1    -                -        -                -                -      F,S
1      10.20.24.0/24       ospf       -         ge0/4       -                -        -                -                -      -
1      10.20.24.0/24       connected  -         ge0/4       -                -        -                -                -      F,S
1      10.20.25.0/24       omp        -         -           -                -        172.16.255.16    lte              ipsec  F,S
1      10.25.1.0/24        static     -         -           -                -        -                -                -      B,F,S
1      10.25.1.1/32        natpool-outside  -    natpool1    -                -        -                -                -      F,S
1      56.0.1.0/24         connected  -         ge0/5       -                -        -                -                -      F,S
1      60.0.1.0/24         omp        -         -           -                -        172.16.255.16    lte              ipsec  F,S
1      61.0.1.0/24         omp        -         -           -                -        172.16.255.16    lte              ipsec  F,S
512    10.0.1.0/24         connected  -         eth0        -                -        -                -                -      F,S
```

The vEdge6 router translates the outside source IP address 10.15.1.4 to 10.16.1.4. The route table on vEdge6 shows this route and that it has been learned from an outside NAT. The next-hop interface for this prefix is natpool2.

```
vEdge6# show ip routes
Codes Proto-sub-type:
  IA -> ospf-inter-area,
  E1 -> ospf-external1, E2 -> ospf-external2,
  N1 -> ospf-nssa-external1, N2 -> ospf-nssa-external2,
  e -> bgp-external, i -> bgp-internal
Codes Status flags:
  F -> fib, S -> selected, I -> inactive,
  B -> blackhole, R -> recursive

                                    PROTOCOL  NEXTHOP     NEXTHOP          NEXTHOP
VPN    PREFIX              PROTOCOL SUB TYPE  IF NAME     ADDR             VPN      TLOC IP          COLOR            ENCAP  STATUS
-------------------------------------------------------------------------------------------------------------------------------------
0      0.0.0.0/0           static   -         ge0/0       10.1.16.13       -        -                -                -      F,S
0      10.0.21.0/24        connected -        ge0/3       -                -        -                -                -      F,S
0      10.0.100.0/24       connected -        ge0/7       -                -        -                -                -      F,S
0      10.1.16.0/24        connected -        ge0/0       -                -        -                -                -      F,S
0      10.1.18.0/24        connected -        ge0/1       -                -        -                -                -      F,S
0      172.16.255.16/32    connected -        system      -                -        -                -                -      F,S
1      2.2.0.0/16          static   -         -           -                -        -                -                -      B,F,S
1      4.4.4.4/32          omp      -         -           -                -        172.16.255.15    lte              ipsec  F,S
1      9.0.0.0/8           static   -         -           -                -        -                -                -      B,F,S
1      10.1.17.0/24        omp      -         -           -                -        172.16.255.15    lte              ipsec  F,S
1      10.1.18.0/24        static   -         ge0/4       10.20.25.18      -        -                -                -      F,S
1      10.2.2.0/24         omp      -         -           -                -        172.16.255.11    lte              ipsec  F,S
1      10.2.3.0/24         omp      -         -           -                -        172.16.255.21    lte              ipsec  F,S
1      10.15.1.4/30        omp      -         -           -                -        172.16.255.15    lte              ipsec  F,S
1      10.16.1.4/30        natpool-outside -  natpool2    -                -        -                -                -      F,S
1      10.20.24.0/24       omp      -         -           -                -        172.16.255.15    lte              ipsec  F,S
1      10.20.25.0/24       connected -        ge0/4       -                -        -                -                -      F,S
1      10.25.1.0/24        omp      -         -           -                -        172.16.255.15    lte              ipsec  F,S
1      56.0.1.0/24         omp      -         -           -                -        172.16.255.15    lte              ipsec  F,S
1      60.0.1.0/24         connected -        ge0/5       -                -        -                -                -      F,S
1      61.0.1.0/24         connected -        ge0/6       -                -        -                -                -      F,S
512    10.0.1.0/24         connected -        eth0        -                -        -                -                -      F,S
```

View Interface Statistics

To display packet receipt and transmission statistics for the interfaces, use the **show interface statistics** command. The output shows the following statistics:

```
vEdge5# show interface statistics natpool1 | notab
interface vpn 1 interface natpool1 af-type ipv4
 rx-packets 0
 rx-octets  0
 rx-errors  0
 rx-drops   0
 tx-packets 0
 tx-octets  0
 tx-errors  0
 tx-drops   0
 rx-pps     0
 rx-kbps    0
 tx-pps     0
 tx-kbps    0
```

To display NAT-specific interface statistics, use the **show ip nat interface-statistics** command. The output shows the following statistics for each NAT pool:

```
vEdge5# show ip nat interface-statistics
ip nat interface-statistics nat-vpn 1 nat-ifname natpool1
 nat-outbound-packets      0
 nat-inbound-packets       0
 nat-encode-fail           0
 nat-decode-fail           0
 nat-map-add-fail          0
 nat-filter-add-fail       0
 nat-filter-lookup-fail    0
 nat-state-check-fail      0
 nat-policer-drops         0
 outbound-icmp-error       0
 inbound-icmp-error        0
 inbound-icmp-error-drops  0
```

```
 nat-fragments             0
 nat-fragments-fail        0
 nat-unsupported-proto     0
 nat-map-no-ports          0
 nat-map-cannot-xlate      0
 nat-filter-map-mismatch   0
 nat-map-ip-pool-exhausted 0
```

View the Data Policy Pushed to the vEdge Routers

To view and verify the data policy pushed from the vSmart controllers to the two vEdge routers, use the **show policy from-vsmart** command. The following is the command output for the vEdge5 router. The output on vEdge6 is identical.

```
vEdge5# show policy from-vsmart
from-vsmart data-policy accept_nat
direction all
vpn-list vpn_1
  sequence 100
   match
    source-ip      10.20.24.0/24
    destination-ip 10.20.25.0/24
   action accept
    count nat
    nat pool 1
  sequence 101
   match
    source-ip      10.20.24.0/24
    destination-ip 10.1.15.13/32
   action accept
    count nat_inet
    nat use-vpn 0
  sequence 102
   match
    dscp 15
   action accept
    count nat_dscp
    nat use-vpn 0
  sequence 104
   match
    source-ip      10.1.18.0/24
    destination-ip 10.20.24.0/24
   action accept
    count nat2
    nat pool 1
  sequence 105
   match
    source-ip      10.1.18.0/24
    destination-ip 10.1.17.0/24
   action accept
    count nat3
    nat pool 1
  sequence 106
   match
    source-ip      10.1.17.0/24
    destination-ip 10.20.25.0/24
   action accept
    nat pool 1
  sequence 107
   match
    source-ip      10.15.1.0/24
    destination-ip 10.20.25.0/24
   action accept
```

```
    nat pool 2
  sequence 108
   match
    source-ip      10.1.17.0/24
    destination-ip 10.25.1.0/24
   action accept
    count nat_108
    nat pool 1
  sequence 109
   match
    source-ip      10.20.24.0/24
    destination-ip 10.25.1.0/24
   action accept
    count nat_109
    nat pool 1
  default-action accept
from-vsmart lists vpn-list vpn_1
vpn 1
```

Configurations for Each Network Device

For each of the network devices in this configuration example, this section shows the portions of the configuration relevant to the service-side NAT configuration.

vEdge5 Router

The vEdge5 router is located at site 500, has a system IP address of 172.16.255.15, and has one connection to the Internet:

```
system
 host-name         vm5
 system-ip         172.16.255.15
 site-id           500
!
vpn 0
 interface ge0/0
  ip address 10.1.15.15/24
  tunnel-interface
   encapsulation ipsec
   color lte
   hello-interval  60000
   hello-tolerance 120
   no allow-service bgp
   allow-service dhcp
   allow-service dhcpv6
   allow-service dns
   allow-service icmp
   no allow-service sshd
   no allow-service netconf
   no allow-service ntp
   no allow-service ospf
   no allow-service stun
  !
  no shutdown
 !
!
```

In VPN 1, NAT pool 1 runs 1:1 static NAT:

```
vpn 1
 interface natpool1
  ip address 10.15.1.4/30
  nat
```

```
   static source-ip 10.20.24.17 translate-ip 10.15.1.4 inside
   static source-ip 10.20.25.18 translate-ip 10.25.1.1 outside
   no overload
  !
  no shutdown
 !
```

VPN 1 also has a number of other NAT pool interfaces:

```
 interface natpool10
  ip address 10.21.29.15/32
  no shutdown
 !
 interface natpool11
  ip address 10.21.30.15/32
  no shutdown
 !
 interface natpool12
  ip address 10.21.31.15/32
  no shutdown
 !
 interface natpool13
  ip address 10.21.32.15/32
  no shutdown
 !
 interface natpool14
  ip address 10.21.33.15/32
  no shutdown
 !
 interface natpool15
  ip address 10.21.34.15/32
  no shutdown
 !
 interface natpool16
  ip address 10.21.35.15/32
  no shutdown
 !
 interface natpool7
  ip address 10.21.26.15/32
  no shutdown
 !
 interface natpool8
  ip address 10.21.27.15/32
  no shutdown
 !
 interface natpool9
  ip address 10.21.28.15/32
  no shutdown
 !
 ip route 2.2.0.0/16 null0
 ip route 4.4.4.4/32 null0
 ip route 10.1.17.0/24 10.20.24.17
 ip route 10.25.1.0/24 null0
!
```

OSPF runs in VPN 1 and is configured to redistribute routes learned from outside NAT prefixes into OSPF:

```
vpn 1
 router
  ospf
   timers spf 200 1000 10000
   redistribute static
   redistribute connected
   redistribute omp
   redistribute natpool-outside
```

```
   area 0
    interface ge0/4
     hello-interval 1
     dead-interval  3
    exit
   exit
  !
 !
!
```

vEdge6 Router

The vEdge6 router is located at site 600, has a system IP address of 172.16.255.16, and has one connection to the Internet:

```
system
 host-name         vm6
 system-ip         172.16.255.16
 site-id           600
!
vpn 0
 interface ge0/0
  ip address 10.1.16.16/24
  tunnel-interface
   encapsulation ipsec
   color lte
   no allow-service bgp
   allow-service dhcp
   allow-service dhcpv6
   allow-service dns
   allow-service icmp
   no allow-service sshd
   no allow-service netconf
   no allow-service ntp
   no allow-service ospf
   no allow-service stun
  !
  no shutdown
 !
!
```

VPN 1 has one NAT pool for static address translation:

```
vpn 1
 interface natpool12
  ip address 10.1.155.4/30
  shutdown
 !
 interface natpool2
  ip address 10.16.1.4/30
  nat
   static source-ip 10.15.1.4 translate-ip 10.16.1.4 outside
   direction outside
   no overload
  !
  no shutdown
 !
 ip route 2.2.0.0/16 null0
 ip route 9.0.0.0/8 null0
 ip route 10.1.18.0/24 10.20.25.18
!
```

BGP runs in VPN 1 and is configured to redistribute routes learned from outside NAT prefixes into BGP:

```
vm6# show running-config vpn 1 router
vpn 1
 router
  bgp 1
   timers
    keepalive 1
    holdtime  3
   !
   address-family ipv4-unicast
    redistribute static
    redistribute omp
    redistribute natpool-outside
   !
   neighbor 10.20.25.18
    no shutdown
    remote-as 2
    timers
     connect-retry          2
     advertisement-interval 1
    !
   !
  !
 !
!
```

Router7 and Router8

Router7 sits in the local site behind the vEdge5 router, and it is an OSPF peer with vEdge5. Router8 sits behind the vEdge6 router and is an IBGP peer with vEdge6.

In our example network, both these routers are configured on vEdge software routers. However, there is nothing in their configuration that specifically relates to static NAT, so we do not show the configurations for these two devices.

vSmart9 and vSmart10 vSmart Controllers

You configure the data policy that runs on the vEdge routers to direct data traffic to the NAT interfaces on the vSmart controllers. The vSmart controllers then push the data policy to the appropriate vEdge routers. The configure data policy must be identical on all vSmart controllers in the overlay network to ensure reproducible data traffic handling in the network.

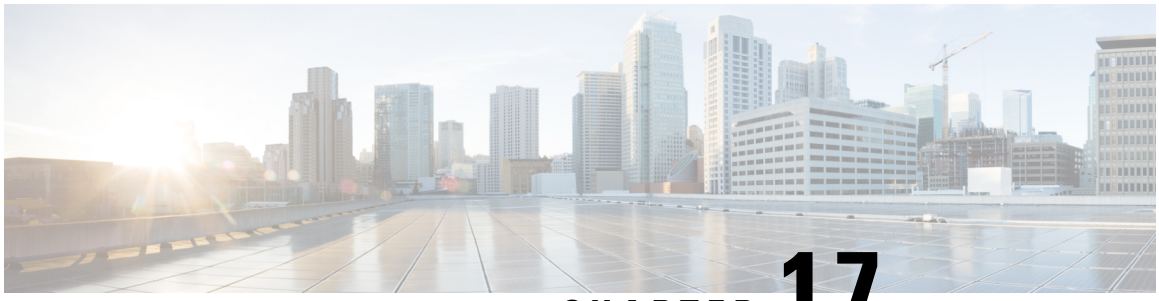
Here is the complete policy configuration for the two vSmart controllers in our example:

```
policy
 data-policy accept_nat
  vpn-list vpn_1
   sequence 100
    match
     source-ip      10.20.24.0/24
     destination-ip 10.20.25.0/24
    !
    action accept
     count nat
     nat pool 1
    !
   !
   sequence 101
    match
     source-ip      10.20.24.0/24
     destination-ip 10.1.15.13/32
    !
    action accept
```

```
  count nat_inet
  nat use-vpn 0
 !
!
sequence 102
 match
  dscp 15
 !
 action accept
  count nat_dscp
  nat use-vpn 0
 !
!
sequence 104
 match
  source-ip      10.1.18.0/24
  destination-ip 10.20.24.0/24
 !
 action accept
  count nat2
  nat pool 1
 !
!
sequence 105
 match
  source-ip      10.1.18.0/24
  destination-ip 10.1.17.0/24
 !
 action accept
  count nat3
  nat pool 1
 !
!
sequence 106
 match
  source-ip      10.1.17.0/24
  destination-ip 10.20.25.0/24
 !
 action accept
  nat pool 1
 !
!
sequence 107
 match
  source-ip      10.15.1.0/24
  destination-ip 10.20.25.0/24
 !
 action accept
  nat pool 2
 !
!
sequence 108
 match
  source-ip      10.1.17.0/24
  destination-ip 10.25.1.0/24
 !
 action accept
  count nat_108
  nat pool 1
 !
!
   sequence 109
 match
  source-ip      10.20.24.0/24
```

```
     destination-ip 10.25.1.0/24
    !
    action accept
     count nat_109
     nat pool 1
    !
   !
   default-action accept
  !
 !
 lists
  vpn-list vpn_1
   vpn 1
  !
  site-list east
   site-id 100
   site-id 500
  !
  site-list vedge1
   site-id 500
  !
  site-list vedge2
   site-id 600
  !
  site-list vedges
   site-id 500
   site-id 600
  !
  site-list west
   site-id 200
   site-id 400
   site-id 600
  !
  prefix-list prefix_list
   ip-prefix 10.20.24.0/24
  !
 !
!

vm9# show running-config apply-policy
apply-policy
 site-list vedge1
  data-policy accept_nat all
 !
 site-list vedge2
  data-policy accept_nat all
 !
!
```



CHAPTER 17

Lawful Intercept 2.0

Note To achieve simplification and consistency, the Cisco SD-WAN solution has been rebranded as Cisco Catalyst SD-WAN. In addition, from Cisco IOS XE SD-WAN Release 17.12.1a and Cisco Catalyst SD-WAN Release 20.12.1, the following component changes are applicable: **Cisco vManage** to **Cisco Catalyst SD-WAN Manager**, **Cisco vAnalytics** to **Cisco Catalyst SD-WAN Analytics**, **Cisco vBond** to **Cisco Catalyst SD-WAN Validator**, **Cisco vSmart** to **Cisco Catalyst SD-WAN Controller**, and **Cisco Controllers** to **Cisco Catalyst SD-WAN Control Components**. See the latest Release Notes for a comprehensive list of all the component brand name changes. While we transition to the new names, some inconsistencies might be present in the documentation set because of a phased approach to the user interface updates of the software product.

Table 40: Feature History

Feature Name	Release Information	Description
Lawful Intercept 2.0	Cisco vManage Release 20.9.1	This feature introduces Lawful Intercept Version 2.0. In the Lawful Intercept 2.0 feature, key information is provided to a law enforcement agency (LEA) by the Cisco Catalyst SD-WAN routers and control components so that they can decrypt the Cisco Catalyst SD-WAN IPsec traffic captured by the Managed Service Provider (MSP). This helps the LEA decrypt the encrypted network traffic information. For information on Lawful Intercept 1.0, see the chapter Lawful Intercept in the Cisco Catalyst SD-WAN Policies Configuration Guide.

Feature Name	Release Information	Description

Lawful Intercept 2.0 Enhancements	Cisco vManage Release 20.10.1	This feature enhances the Cisco SD-WAN Manager GUI and the troubleshooting options available for the Lawful Intercept feature in Cisco Catalyst SD-WAN.
		• Cisco SD-WAN Manager GUI enhancements:
		  • A **Sync to vSmart** button to synchronize a newly created intercept configuration with the Cisco SD-WAN Controller.
		  • A toggle button to enable or disable an intercept.
		  • A progress page to display the status of synchronization and activation.
		  • A red dot on the task list icon in the Cisco SD-WAN Manager toolbar to indicate any new lawful intercept tasks.
		  • A task list pane to view a list of active and completed lawful intercept tasks.
		  • An intercept retrieve option **Get IRI** to retrieve key information or Intercept Related Information (IRI) from the Cisco SD-WAN Controller.
		• Ability to troubleshoot Cisco SD-WAN Controller and Cisco SD-WAN Manager using the debug logs and admin tech files.

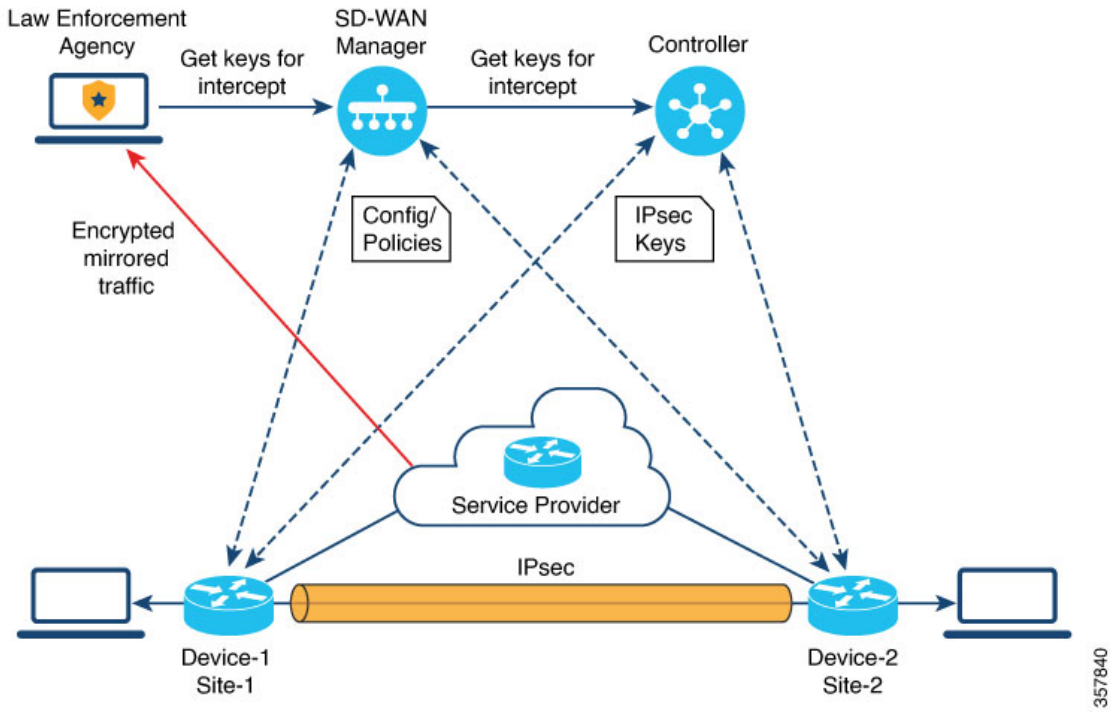
Lawful Intercept 2.0 Enhancements	Cisco Catalyst SD-WAN Manager Release 20.12.1	This feature extends Lawful Intercept to multitenancy mode, and provides support for Cisco SD-WAN Manager clusters. For more information on Cisco SD-WAN Manager clusters, see Cluster Management.

Information About Lawful Intercept 2.0

Cisco Catalyst SD-WAN's Lawful Intercept feature allows an LEA to get a copy of network traffic for analysis or evidence. This is also referred as traffic mirroring. See the chapter Lawful Intercept in the Cisco Catalyst SD-WAN Policies Configuration Guide.

From Cisco vManage Release 20.9.1, Cisco Catalyst SD-WAN implements a new architecture for Lawful Intercept , as shown in the following figure.

Figure 14: Lawful Intercept 2.0 Architecture



The following are the characteristics of the new architecture:

- Traffic mirroring is outside the scope of Cisco Catalyst SD-WAN. The LEA works with the corresponding service provider to capture network traffic for mirroring.

Note In the illustration above, the service provider is an underlay connection and the IPsec tunnel is an overlay connection.

- Because the captured network traffic is encrypted, Cisco SD-WAN Manager and Cisco SD-WAN Controller provide key information to the LEA.
- The LEA retrieves the keys from Cisco SD-WAN Manager to decrypt Cisco Catalyst SD-WAN IPsec traffic. The LEA ensures that they retrieve key information is retrieved during each rekey period. The rekey period is provided by the service provider. For more information about retrieving keys, see Retrieve an Intercept, on page 267. For information on rekey period, see Configure Data Plane Security Parameters.

A Lawful Intercept administrator is solely responsible for configuring intercepts and creating Lawful Intercept API users who perform Lawful Intercepts. A Cisco SD-WAN Manager administrator can create an account for the Lawful Intercept administrator; the administrator must be a member of the **li-admin** group. For more information about creating an account for a Lawful Intercept administrator, see Create Lawful Intercept Administrator.

Prerequisites for Cisco Catalyst SD-WAN Lawful Intercept 2.0

- A Cisco SD-WAN Controller must be set to **Manager mode**.
- For more information about decrypting the IPsec traffic in Cisco Catalyst SD-WAN, contact Cisco Support or Cisco Sales team.

Benefits of Cisco Catalyst SD-WAN Lawful Intercept 2.0

- It is not necessary to configure edge devices for Lawful Intercepts.

Note To configure an intercept, an administrator must select the edge devices that have to be included in the intercept. This is necessary because the key information that is retrieved from Cisco SD-WAN Manager also includes the keys for the selected devices.

- The service provider captures the data traffic for interception. Traffic is not intercepted from the edge devices.

Configure Lawful Intercept 2.0 Workflow

Note The Lawful Intercept feature can be configured only through Cisco SD-WAN Manager, and not through the CLI.

To configure Lawful Intercept in Cisco SD-WAN Manager, perform the following steps:

1. Create Lawful Intercept Administrator
2. Create Lawful Intercept API User
3. Create an Intercept

Create a Lawful Intercept Administrator

Using the Admin account in Cisco SD-WAN Manager, create an account for the Lawful Intercept administrator.

1. From the Cisco SD-WAN Manager menu, choose **Administration** > **Lawful Intercept**.
2. Click **Add User** to create a Lawful Intercept administrator user account.
3. In the **Full Name** field, enter a full name for the Lawful Intercept administrator.

4. In the **User Name** field, enter a user name for the Lawful Intercept administrator. The user name must be prefixed with **li-**.
5. In the **Password** field, enter a password for the Lawful Intercept administrator.
6. Confirm the password in the **Confirm Password** field.
7. From the **User Group** drop-down list, choose **li-admin**, and then click **Add**.

   The newly created Lawful Intercept administrator user account is displayed in the **Users** window.

Create a Lawful Intercept API User

The Lawful Intercept API User account is for those users of LEA who log in and retrieve key information using Cisco SD-WAN Manager's REST API. These are the users who perform a lawful intercept of the Cisco Catalyst SD-WAN IPsec traffic.

The LEA use **`https://{vmanage_ip}/dataservice/li/intercept/retrieve/<intercept_id>`** to retrieve the key information.

To create a Lawful Intercept API user, perform the following steps:

1. Log in to Cisco SD-WAN Manager as a Lawful Intercept administrator.

Note When a Lawful Intercept administrator logs in to Cisco SD-WAN Manager, only the **Monitor** and **Administration** options are available in the Cisco SD-WAN Manager menu.

2. From Cisco SD-WAN Manager menu, choose **Administration** > **Lawful Intercept**.
3. Click **Add User** to create an Lawful Intercept API user account.
4. In the **Full Name** field, enter a full name for the Lawful Intercept API user.
5. In the **User Name** field, enter a user name for the Lawful Intercept API user. The user name must be prefixed with **li-**.
6. In the **Password** field, enter a password for the Lawful Intercept API user.
7. Confirm the password in the **Confirm Password** field.
8. From the **User Group** drop-down list, choose **li-api**, and click **Add**.

   The newly created Lawful Intercept API user account is displayed in the **Users** window. The LEA can log in to Cisco SD-WAN Manager using the Lawful Intercept API user account to retrieve key information.

Create an Intercept

Minimum supported release: Cisco vManage Release 20.9.1 and Cisco Catalyst SD-WAN Control Components Release 20.9.1

Configure an intercept to collect intercept data. To configure an intercept, do the following:

1. From the Cisco SD-WAN Manager menu, choose **Administration** > **Lawful Intercept**.
2. Click the **Intercepts** tab, and then click **Add Intercepts**.
3. Cisco Catalyst SD-WAN Manager Release 20.12.1 and later releases:

   From the **Tenant** drop-down list, choose a tenant. For more information about adding a tenant, see Add a New Tenant.
4. In the **Intercept ID** field, enter a number. Enter a minimum of two digits and a maximum of 25 digits.
5. In the **Description** field, enter a description for the intercept.
6. By default the **Enable** toggle button is enabled. However, the intercept remains in an inactive state after it is created.
7. Click **Next**.

   In single-tenant mode, the **Add Edge Devices** pop-up window displays all the edge devices in the Cisco Catalyst SD-WAN network.

   Cisco Catalyst SD-WAN Manager Release 20.12.1 and later releases:

   In multi-tenant mode, the **Add Edge Devices** pop-up window displays all the single-tenant edge devices associated with the selected tenant.
8. Click one or more edge device names to add to the intercept and click **Next**.

   Cisco SD-WAN Manager provides the keys for the edge devices selected here.

Note Specify an intercept warrant for all the edge devices that are added to the intercept.

When an edge device is added for interception, all its peer devices, which are connected in the same network, are also available for Lawful Interception.

9. The **Add LI API users** pages displays all the LI-API users created by the Lawful Intercept administrator.
10. Click one or more user names to add to the intercept. The users selected here can retrieve key information that is required for interception from Cisco SD-WAN Manager. For information on how keys are retrieved for an intercept, see Retrieve an Intercept.
11. Click **Summary**

    The summary of the intercept is displayed.
12. Click **Submit**. The **Intercepts** page displays the configured intercept.
13. Click **Sync to vSmart** to synchronize the configured intercept configuration in Cisco SD-WAN Manager with Cisco SD-WAN Controller.

    A progress page displays the status of the synchronization and activation. After successful synchronization, the **Activate State** field displays a green check mark.

Note The **Activate State** field displays a green check mark status only if Cisco SD-WAN Controller is set to **Manager** mode.

If there are any additional Lawful Intercept tasks, a red dot is displayed on the task list icon in the Cisco SD-WAN Manager toolbar. Click the tasks list icon to view a list of all the active and completed Lawful Intercept tasks. You can view up to 500 latest Lawful Intercept tasks.

If an intercept is modified, the **Sync to vSmart** button is enabled. Click **Sync to vSmart** to synchronize the intercept configuration in Cisco SD-WAN Manager with Cisco SD-WAN Controller.

Note The **Sync to vSmart** button is enabled only when a new intercept is created, or when an intercept is edited or deleted.

To retrieve key information that is required for interception, click **…**, and then click **Get IRI**. The IRI is retrieved from Cisco SD-WAN Controller and displayed in Cisco SD-WAN Manager.

Retrieve an Intercept

An LEA is responsible to periodically retrieve key information because this information is required to decrypt the traffic captured by the MSP.

An LEA can retrieve key information by using Cisco Catalyst SD-WAN Manager REST APIs.

1. An LEA logs in to Cisco SD-WAN Manager as a Lawful Intercept API user.
2. After a Lawful Intercept API user is authenticated, the LEA sends a request using the Cisco SD-WAN Manager REST APIs specifying the intercept ID that it wants to get the key information for.
3. When a request from the LEA is received by Cisco SD-WAN Manager, Cisco SD-WAN Manager forwards the request to the Cisco SD-WAN Controller on which intercepts are configured.
4. Cisco SD-WAN Controller then retrieves the key information for the specified intercept ID and returns the key information to Cisco SD-WAN Manager in JSON format.