Policy Framework Applications

This article builds on the discussion in the previous article, *Viptela Policy Framework Basics*, describing policy applications that control the distribution of route information in the overlay network, and that direct data traffic flows through the network. All the policies we discuss here are vSmart policies, also called centralized policies, which are policies that are configured on the vSmart controller.

We discuss the following applications of the vSmart policy framework:

- Application-aware routing policy looks at the network and path characteristics of the data plane connections between vEdge routers, compares them to configured SLA parameters, and computes optimal paths for data traffic.
- Control policy uses OMP vRoute and TLOC information to enable services and engineer specialized routing applications.
- Data policy
- VPN membership policy

Application-Aware Routing Policies

Application-aware routing tracks packet loss and latency on the data plane connections between vEdge routers, to compute optimal paths for data traffic in the overlay network. The loss and latency data supplements the standard routing parameters—such as route prefixes, metrics, and link-state information—in determining traffic paths through the network.

Application-aware routing policy is defined by means of a specially designed vSmart policy, called an app-route-policy. This policy specifies the performance characteristics in the form an SLA, which defines the required latency and loss conditions on the data plane connection that trigger the policy to take effect.

Application-aware routing is created in three portions of the configuration:

- Configure the sla-class, which defines the required latency and loss for the traffic that is to be affected by a given app-route-policy.
- Configure the app-route-policy, which specifies the traffic that is to belong to an sla-class. (This is done in a fashion similar to a data-policy.) The app-route-policy references a vpn-list to dictate which VPNs at the listed sites benefit from the policy.
- Apply the app-route-policy towards the desired overlay network sites.

The following is an example of an app-route-policy configuration:
**sla-class** defines the required performance metrics for the application.

Here, the sla-class is named EF. For the performance metrics, the maximum acceptable packet loss is 2 percent and the maximum acceptable packet latency is 100 milliseconds. If either one of these values is greater than the configured value, the data tunnel is considered invalid. In other words, if the latency is greater than 100 milliseconds or if the packet loss is greater than 2 percent, that tunnel is not considered as a valid data path.

**vpn-list** defines the VPNs where the policy is applied to at the target sites. You create the vpn-list in the lists section, and you apply it in the definition of the app-route-policy itself.

Here, the app-route-policy applies only to the vpn-list called app_vpn, which affects only data traffic in VPN 1.

**app-route-policy** defines the actual application route policy. In the policy definition, you link the matched traffic (in the match portion of the policy) with the required sla-class, which is called in the action portion of the policy.

Here, the action portion of the app-route-policy calls the previously defined sla-class EF.

---

**Cflowd Flow Data Collection**

Cflowd monitors traffic flowing through vEdge routers in the overlay network and exports flow information to a collector, where it can be processed by an IPFIX analyzer. Cflowd periodically sends template reports to flow collector. These reports contain information about the flow and data extracted from the IP headers of transiting flows.

Cflowd flow collection is enabled by means of a vSmart policy, specifically, by a vSmart data policy. The parameters for capturing and exporting flow data are defined in two sections of the policy:

- A cflowd-template configures the flow cache behavior and flow export. For the flow cache, the cflowd-template defines how often the sampled flows should be sent to a collection. For the flow export, the cflowd-template specifies the location of the flow collection. You must configure a cflowd-template, but it need not contain any parameters. With no parameters, the data flow cache on vEdge nodes is managed using default settings, and no flow export occurs.
• A data-policy selects the traffic subject to flow data collection. The data-policy can be configured to be very specific or as a general flow collection filter, depending on requirements.

Both the cflowd-template and the data-policy components are controlled and configured on the vSmart controller, and they are both distributed from the vSmart controller to the affected vEdge routers. This design eases the configuration and application of traffic flow monitoring.

The following is an example of configuring a cflowd-template and a data-policy:

```
apply-policy
site-list site100
data-policy cflowd_data all
cflowd-template cflowd_temp
!
!

policy
data-policy cflowd_data
vpn-list cflowd_vpn
sequence 10
match protocol 17
| action accept
cflowd
| default-action drop
!
!
cflowd-template cflowd_temp
flow-active-timeout 60
flow-inactive-timeout 60
collector vpn 100 address 1.1.1.1 port 4739 transport transport_udp
!
!
```

data-policy (in the purple policy section of the configuration) defines the actual traffic subject to flow data collection. Here, data-policy cflowd_data matches all UDP traffic (protocol 17) and applies flow monitoring to it. The default-action is drop, so all non-UDP data traffic is dropped.

data-policy (in the green apply-policy section) applies data-policy cflowd_data to the vEdge routers at sites that are part of the site-list named site100. The all option applies the policy to data traffic entering and leaving the vEdge routers.

cflowd-template (in the brown policy section) creates a template to manage cache management and flow export settings. Here, we define the template called cflowd_temp that gives the location flow collector location (in VPN 100, at address 1.1.1.1 and port 4739) and exports flow information every 60 seconds for both active and inactive flows.

cflowd-template (in the green apply-policy section) applies the cflowd template to the vEdge routers in site-list site100. The cflowd-template and data-policy are applied at the same time.

Control Policy Applications and Services

vSmart control policies are put in place to influence routing in the overlay domain. Using OMP routing information, these policies can enable services and engineer specialized routing applications. Control policy is a powerful tool for any type of path management. Having it be centralized and managed on vSmart controllers greatly simplifies policy operations throughout the network.

In this section, we describe the following types of vSmart control policies:
Service Chaining with Control Policy

When security devices and resources are located at one site in the network and routers and hosts dispersed throughout the network need to access these resources, you can use vSmart policies to redirect data traffic to those resources before it is forwarded towards its final destination. In the Viptela policy framework, this process is called service chaining, because an intermediate destination at which a service is located is being attached, or chained, to a route. These centralized security devices and resources are called services.

vSmart policy that implement service chaining selectively directs traffic to standard services—firewalls, Intrusion Detection Systems (IDSs), and Identity Providers (IDPs)—or to custom services that you define.

As an example of how service chaining can be used, traffic from some departments (in the figure below, those in VPN 1) may require firewall protection when traveling to and from the Internet or external networks, while traffic from other departments may not (in VPN 2). Departments in VPN 1 are located at Sites 1 and 3, and their traffic needs to use the firewall service at Site 4. The Viptela solution applies a vSmart service-chaining policy to the affected VPN.

The configuration for the service chaining has the following components:
• The service originator advertises the firewall service.

• A control-policy directs traffic in VPN 1 that is headed from the overlay network to the Internet through the firewall service. This is the policy that is applied to Sites 1 and 3.

• A control-policy directs incoming traffic destined for VPN 1 through the firewall service. Control-policy is unidirectional, which is why we need to define two control policies. This policy is applied to Site 2.

• Application of the two policies.

First, the service originator advertises the firewall service. The originator is the vEdge router at Site 4, which advertises a firewall device at IP address 1.1.1.1 that is available to VPN 1. This vEdge router advertises this information in the OMP service routes that it sends to the vSmart controller.

```
vpn 1
   service FW address 1.1.1.1
```

The vSmart control policies have these components:
In **vpn-list**, the list called *closed_vpn* contains VPN 1, the protected VPN. We need this list for the match condition in control-policy service-chain. If we decide later that traffic from other VPNs needs firewall protection, we can simply add these VPNs to the *closed_vpn* VPN list.

In **site-list**, we create the lists used in the **apply-policy** commands. The site-list *branch_sites* holds the two sites in VPN 1, and site-list *site2* contains the site that connects to the Internet.

**control-policy service-chain** defines the target routes for which the firewall service is needed. These routes are external BGP prefixes in VPN 1. A default-action of accept means that nonmatching prefixes, such as those in other VPNs, pass through the policy unchanged.

**control-policy service-chain-return** defines the target routes for the firewall service in the return path. These are the prefixes in Sites 1 and 3 that are in VPN 1. Again, the default-action accept means that nonmatching prefixes pass through the policy unchanged.

**apply-policy site-list branch_sites** applies the control-policy service-chain for the upstream firewall service to Sites 1 and 3. This control-policy has the following effect on outbound traffic towards the Internet:

- For traffic originating from VPN 1 at Sites 1 and 3, the next hop for outgoing BGP prefixes is changed to point to the location where the firewall service is hosted.
- The firewall receives the traffic from its directly connected vEdge router, processes it, and then follows its route table, which points back to the connected vEdge router. It is very likely that the firewall device simply has a default route and nothing more in its route table.
- Because no policy is applied to the vEdge router adjacent to the firewall device, when it performs a route table lookup, the next hop for the BGP prefixes is the node that originally sourced the traffic into OMP. This node is the vEdge router at Site 2, the router that originally learned the prefixes from the Internet and advertised them into OMP.

To summarize, in the upstream direction (from the overlay network to the Internet), the original next hop for BGP
prefixes is used for all traffic except for VPN 1 traffic from Sites 1 and 3.

**apply-policy site-list site2** applies the control-policy service-chain-return for incoming (downstream) firewall service. This control-policy has the following effect on traffic inbound from the Internet:

- For prefixes that were originally sourced from Sites 1 and 3 in VPN 1, Site 2 changes the next hop to point to the firewall service.
- The firewall receives the traffic from the directly connected vEdge router, processes it, and, per its route table, sends it back to its connected vEdge router.
- Because no policy has been applied to the vEdge router at Site 4, the original next hops for the OMP-sourced prefixes are used, and the traffic is directed towards Sites 1 and 3.

### Traffic Engineering

Traffic engineering is a mechanism for controlling traffic flow through links in the network, optimizing the path to suit the needs of the network. MPLS is one way to integrate traffic engineering into Layer 3 applications, such as routing IP traffic. In the Viptela network, you can use vSmart policy directly to traffic-engineer paths through the overlay network. In the example illustrated below, even though the best path from Carrier 1 to Carrier 4 is through Carrier 2, we instead want traffic to go via Carrier 3.

The vSmart policy configuration to establish traffic engineering has these components:
site-list has two lists. One identifies the sites in the overlay network where we apply the policy (Site 1). The second is used in the match condition, to identify traffic headed towards Site 4.

tloc-list is where you actually engineer the traffic path. You do this by setting the preference on two of the TLOCs. (In the configuration commands, you identify the TLOCs by their IP addresses, not by "site3" and "site4"). The preference to use TLOC 3, and so to have traffic go via Carrier 3, is higher that the preference to use TLOC 4 (and go via Carrier 2 to Carrier 4).

control-policy traffic_eng defines the target prefixes for which traffic engineering is required. The policy matches prefixes associated with routes to Site 4. The action in this policy is to change the data packet's TLOC properties to those defined in the tloc-list prefer_carrier3 list, so the traffic prefers Site 3 (and Carrier 3) as its next hop.

apply-policy applies the traffic engineering policy towards the target sites in site-list site1. The policy is applied in the outbound direction, which means that any route changes that occur as a result of the policy are done after the vSmart controller sends the OMP route update and before the update is received by the vEdge router at Site 1.

Extranet VPNs

When services that reside in a VPN must be shared across users residing in multiple other VPNs (see illustration below), you can create a vSmart extranet VPN control policy. This policy advertises the prefixes in the service VPNs to the other (client) VPNs, and it advertises the client prefixes to the service VPN, so that the vEdge routers connected to these VPNs have reachability information for each other.
In this example, the shared services, which are file servers are located in VPN 100, and users in VPNs 2 and 3 need access to the file servers (and users in VPN 1 do not).

We need three lists—two VPN lists and one site list. One VPN lists contains the member of the service VPN (VPN 100), and the second the members of the client VPNs (VPNs 2 and 3). Both VPN lists are used in the control-policy match conditions. The third list, which contains all the sites in the overlay network, is used in the apply-policy statement.

The control-policy has two sequence entries, one that matches the client_vpn routes and exports (advertises) them to the service VPN, and one that matches the service_vpn routes and advertises them to the client VPNs:
To have the vSmart control-policy take effect, apply it inbound direction for prefixes for each affect VPN to be imported into the vSmart controller's route table:

```
apply-policy
  site-list all_sites
  control-policy extranet in
```

---

**Data Policy**

vSmart data policy is a powerful tool for any type of data plane–centered traffic management. Because it is centralized and managed on a vSmart controller data policy greatly simplifies policy operations. Data policies are used to enable a number of services, including:

- Service chaining
- Cflowd traffic monitoring
- Traffic policing and counting

You configure and apply data policies on the vSmart controller, which then pushes the policy to the vEdge routers. It is the routers that enforce the configured policy in the data plane. A data policy acts on an entire VPN and is not interface-specific.

Some of the policy applications that can be enabled with control policies can also be enabled with data policies (and also with traditional routing policy). Enabling applications with data policies instead of control policies can be simpler to administer, because the policies apply only to a single node and they do not interact across the overlay network.

---

**Service Chaining with Data Policy**

Earlier, we showed a way to configure service chaining with control-policy. For the same network, here is the configuration done with data-policy.
As before, the service originator (Site 4) advertises the firewall service. Here is the configuration for the vEdge router at Site 4:

```
vpn 1
  service FW address 1.1.1.1
```

Then we configure the actual policy on the vSmart controller:
First we create two lists:

**site-list** identifies the branch sites. These are the sites included in the apply-policy command, the sites that receive the data-policy. Sites 1 and 3 in our figure serve the protected VPN, VPN 1, so the list contains these two sites.

**vpn-list**, as in the control-policy version, identifies the VPNs whose traffic needs to pass through the firewall service.

**data-policy service-chain** defines the target DSCP value for traffic that requires the firewall service.

Here, the policy, called closed_vpn, sets the criteria that packets in the protected VPN must match to be directed to the firewall service. The match in this example is done on QoS, using the DSCP value of 10.

**apply-policy site-list branch_sites** applies the data policy to the branch sites in the list. data-policy is always applied on a directional& basis: from-service, from-tunnel, or both (**all**).

In a situation where a control-policy was considered to be a better fit for a service-chaining application, it could be deployed in conjunction with a data-policy to enable cflowd data flow collection.

---

**Using a NAT for Local Internet Exit**

Rather than have a single exit point from the overlay network to the Internet, vSmart data-policy can provide local Internet exit from vEdge routers. You implement this using a data-policy that includes a NAT directive. The data-policy is configured on the vSmart controller, so local Internet exit is managed centrally.

The figure below illustrates a topology that provides local Internet exit to departments that are part of VPN 2. The users in these departments are located at all three of the sites in the overlay network.
Here are the configuration components that set up the NAT function on the vEdge routers so that traffic passing destined for the Internet can travel directly from the vEdge router to the Internet, without having to go to a centralized or other site before exiting to the Internet:

- **data-prefix-list** identifies the source IP prefixes whose traffic is destined for the NAT, and hence for the Internet
- **vpn-list** identifies the affected VPNs.
- **site-list** groups together the three sites that participate in VPN 1.
- **data-policy** in the **policy** section directs matching traffic towards the NAT, and hence towards the Internet.
- **data-policy** in the **apply-policy** section is applied in the from-service direction to match incoming traffic from the service side to the vEdge router.

### VPN Membership Policy

The default behavior of the Viptela OMP architecture is to advertise any configured VPN to any node on which it is configured. This design automatically establishes connectivity without unnecessary configuration and operational overhead. However, certain VPNs may be of a sensitive nature such that their membership must be tightly controlled.

VPN membership policy serves to restrict the distribution of VPN information from the vSmart controllers to those VPNs that are explicitly approved. With a VPN membership policy, a vEdge router that is not explicitly allowed to participate in a VPN may have the VPN configured, but it can see only local connectivity and routing information.

You can use VPN membership policy to establish both whitelist and blacklist behavior.
For VPN membership policies, you configure the following components:

**vpn-lists** define the VPN match data. The example shown to the left has two VPN lists: **vpn-list sites_1** contains VPNs 10 and 20, and **vpn-list sites_2** contains VPNs 30 and 40.

**site-lists** are used in two places: in the policy match condition and when applying the policy to the affected sites in the overlay network.

**vpn-membership** defines the policies themselves, which can act as either whitelists or blacklists for VPN filtering.

**apply-policy** applies the VPN membership policies in both directions, to determine which VPNs are allowed from a given site.

---

**Additional Information**

- Configuring Centralized Control Policy
- Configuring Centralized Data Policy
- Viptela Policy Framework Basics