SD-WAN

CONTENT

1. Introduction to SD-WAN.....1

Why do we need SD-WAN?
What is SD-WAN?
How Cisco SD-WAN works?

2. Key features of Cisco SD-WAN.....16

Application Quality of Experience (AppQoE)
Interconnecting Multiple Clouds
Direct Internet Access (DIA)
RESTful APIs

3. Cisco SD-WAN Control Plane.....31

Underlay vs Overlay Routing
OMP Overview
OMP Best-Path Selection
QUIZ: Control Plane Fundamentals

4. Cisco SD-WAN Data Plane.....46

What is a TLOC?
TLOC Color and Carrier
Tunnel Groups
TLOCs and NAT
Data Plane Encryption
VPN Segmentation
QUIZ: Data Plane Fundamentals

5. Cisco SD-WAN Deployment.....66

Controllers Identity and Whitelisting
WAN Edge Deployment
Last Resort Circuit
TLOC Extension

6. Cisco SD-WAN Management Plane.....81

Cisco SD-WAN Policies
Cisco SD-WAN Templates
vManage Mode

7. Cisco SD-WAN Home Lab.....91

Cisco SD-WAN on EVE-NG
Packet loss, Latency and Jitter on EVE-NG

8. Centralized Control Policies.....103

What is a Centralized Control Policy?

Inbound vs Outbound Control Policy

LAB 1: Hub-and-Spoke - Restricting spoke-to-spoke tunnels

LAB 2: Hub-and-Spoke - Allowing hub-to-spoke routing

LAB 3: Hub-and-Spoke - Enabling spoke-to-spoke communication

LAB 4: Traffic Engineering - TLOC Preference

LAB 5: Traffic Engineering - End-to-End Path Tracking

Lab 6: VPN Membership Policy - Isolating guest users

QUIZ: Centralized Control Policies

9. Centrilized Data Policies.....150

What is a Centralized Data Policy?

10. Application-Aware Routing Policies.....153

What is a Centralized Data Policy?

12. Cloud OnRamp.....158

Cloud onRamp for SaaS

Cloud onRamp for IaaS

Cloud onRamp with AWS - Design Options

1. INTRODUCTION TO SD-WAN

Why do we need SD-WAN?

Inefficiencies of Traditional WAN

In 2020/2021, businesses are embracing digital transformation more rapidly than ever expected. Remote working and online meetings are now standards. Many applications are moved to the public cloud and many services are now available over the Internet. Companies want to reduce costs and manage their infrastructure more effectively. However, the traditional wide-area network (WAN) was designed to connect users at remote sites to applications hosted in the company's data center. Dedicated leased lines and MPLS circuits were used to provide secure and reliable connectivity to the DC. Although some applications are now in public clouds and the Internet, the traffic from the remote sites must come to the DC first and then be routed to the Public Cloud and back. This concept is visualized in figure 1.

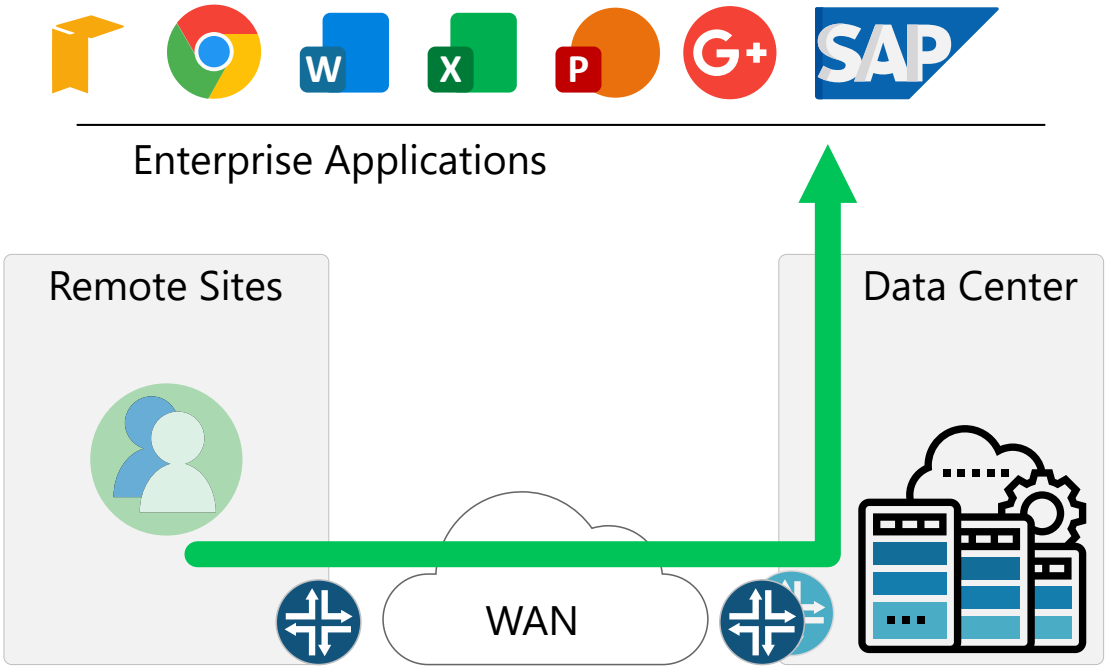


Figure 1. Traditional WAN architecture - Centralized Connectivity

This WAN design no longer works well in a digital world where applications are out of the data center, and the users consuming those applications are using a diverse set of mobile devices. As businesses are rapidly adopting Software-as-a-service (SaaS) and Infrastructure-as-a-service (IaaS) models, it is pretty common to have ERP applications hosted in AWS, office applications such as Office 365 being used over the Internet, company-specific apps hosted in the HQ data center, and 3rd party applications hosted in another datacenter. In this scenario, the traditional WAN connectivity between the branches and the DC is not the most effective way to connect to all applications and creates the following inefficiencies:

- **Costs -** Increased bandwidth demands are forcing companies to upgrade their private WAN circuits, which is expensive;
- **Higher Latency -** Moving the traffic from remote sites to the DC and then to the Cloud increases the overall round-trip times;

- **Availability -** Running everything through the company's data center creates a Single-point-of-failure;
- **Velocity -** Deploying private MPLS circuits is a slow and tedious process that usually slows down the roll-out of new remote sites.

With the adoption of Public Cloud, companies started rethinking their WAN designs. Organizations decided to equip their remote site with Direct Internet Access links and to offload cloud-native applications directly over the Internet. Internet availability then becomes a very important part of the branch/campus operations.

But how do you make sure all remote sites have an Internet connection in 99.99% of the time? Well, the answer is - you purchase at least two independent Internet connections from at least two service providers. Combining this with the emergence of 4G/5G and having in mind that commodity internet circuits offer higher capacity at significantly lower price points, it is a natural consequence that companies started exploring ways to rely less on Private WAN and take advantage of the Internet circuits.

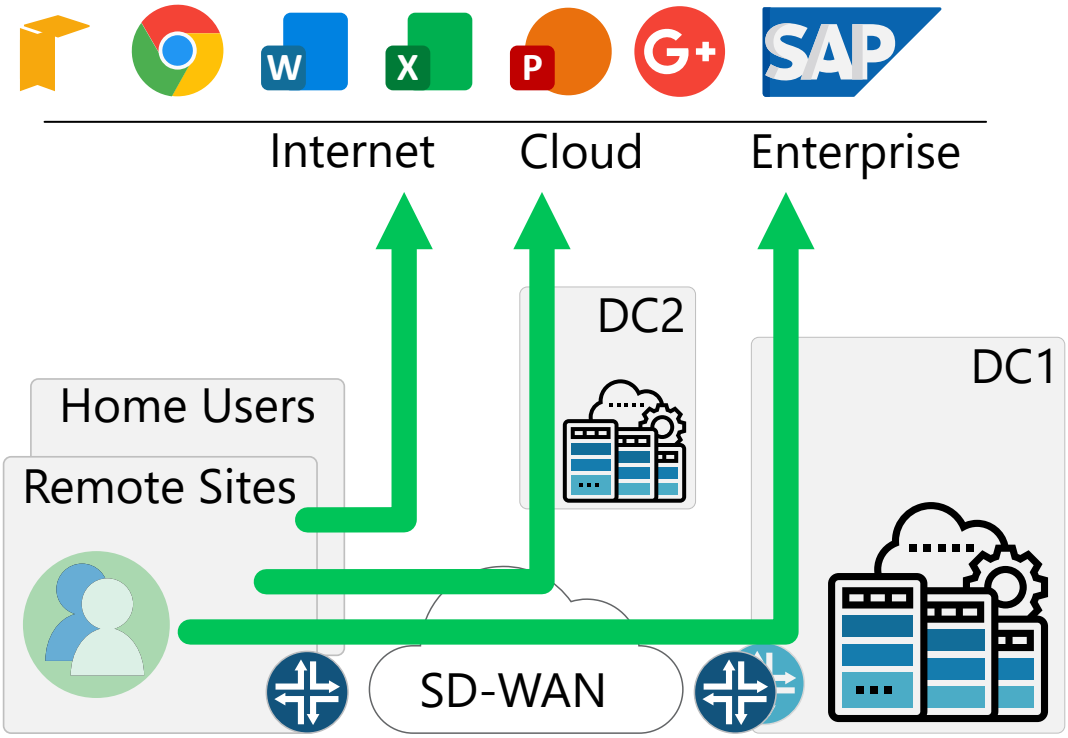


Figure 2. Software-Defined WAN architecture - Decentralized Connectivity

Software-defined WAN (SD-WAN) solutions have been designed to address these challenges. SD-WAN is part of the broader technology trend called software-defined networking (SDN). This is a new centralized approach to network management that abstracts the underlying network infrastructure away from the services and applications that run over the network.

De-centralized Network Management

For many years, networks have been deployed and operated in a decentralized manner, meaning each device is individually managed and operated by network administrators. Let's compare this to a centralized approach. If I take you back to the old days of personal computers. Installing new hardware or software required users to configure the individual elements of the PC. For example, you bought a new sound card for your PC. You install the card and then you must go to the website of the manufacturer and download the drivers for your operating system. Then you install the drivers and resolve any incompatibility issues. And only then you start listening to music, eventually.

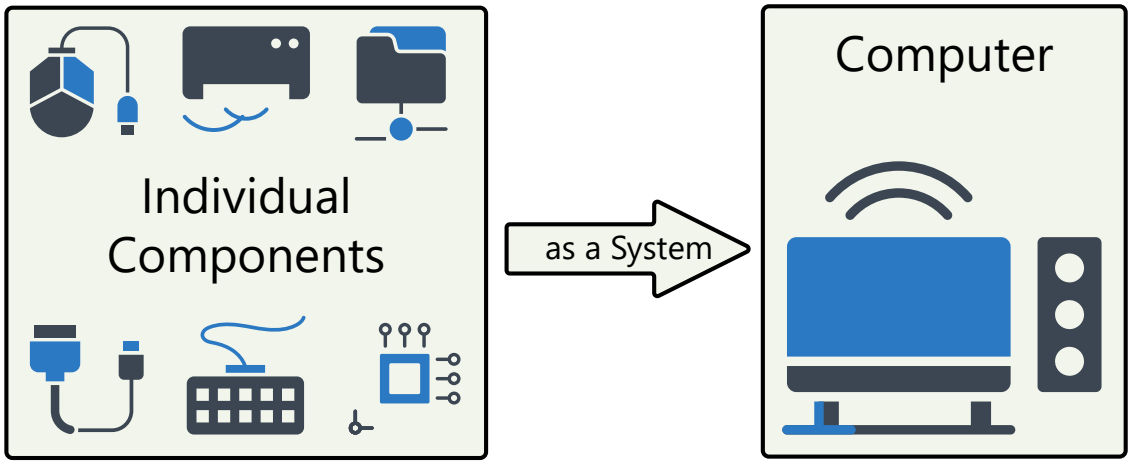


Figure 3. Personal Computer as a System

Now compare this to the process nowadays. You just plug the hardware component you bought and the PC handles everything else for you. You just signify the intent to listen to music and the operating system configures all underlying components necessary to play the music. You are using a personal computer as one system and not as a group of individual components.

And the one million dollar question is - Why can't we apply this logic to IP Networks? Why the network can't be thought of and administered as a system instead of a collection of individual devices such as routers, switches, and firewalls?

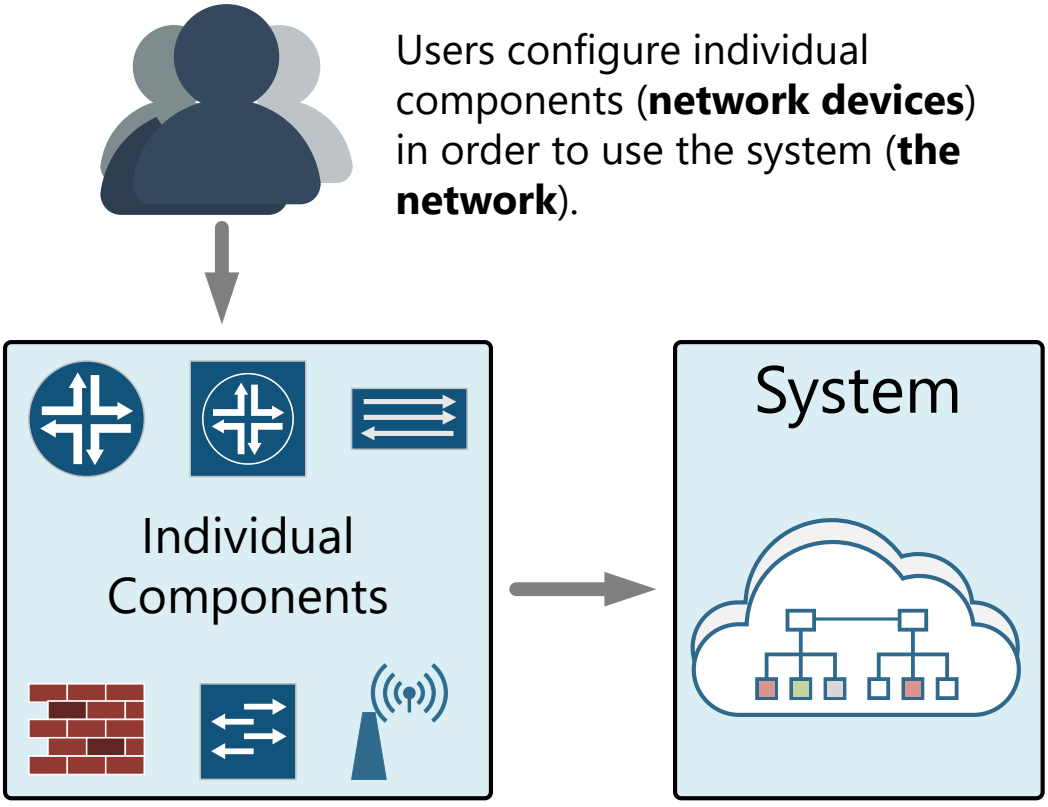


Figure 4. Using Traditional Networks

As of today, most networks in the world are still being operated and configured manually using traditional "per device" methods. Although many of us will agree that this device per device approach has many significant drawbacks such as:

Risk of human error - Many kinds of research have been made over the years that shows that most network outages happen because of a configuration error (ultimately a human error).

Services velocity - Many network engineers would agree that the network is the slowest of all technology verticals when it comes to enabling a new feature or service. Configuring each network device one by one using CLI is just not a scalable approach. Don't get me wrong, we all love the command line, but it was not designed to make massive scale configuration changes to multiple devices at the same time.

Analytics - In traditional networks, where devices are managed in a decentralized manner, almost no one knows the "big picture". In many cases, different devices are operated by different teams, and a centralized collection of analytic data and network-wide configuration sanity is a very hard task.

Benefits of SD-WAN

Many business and technical researchers agree that the next-generation networks would be deployed and operated As-a-System and not as a collection of individual network devices. Software-Defined WAN (SD-WAN) is a centralized approach to managing and operating large-scale WAN networks.

Single Management Plane

One of the main ideas of SD-WAN is to administrate the WAN through a single centralized management plane, and the system itself to manage the underlying network devices. This would provide many benefits, business opportunities, and a better overall user experience.

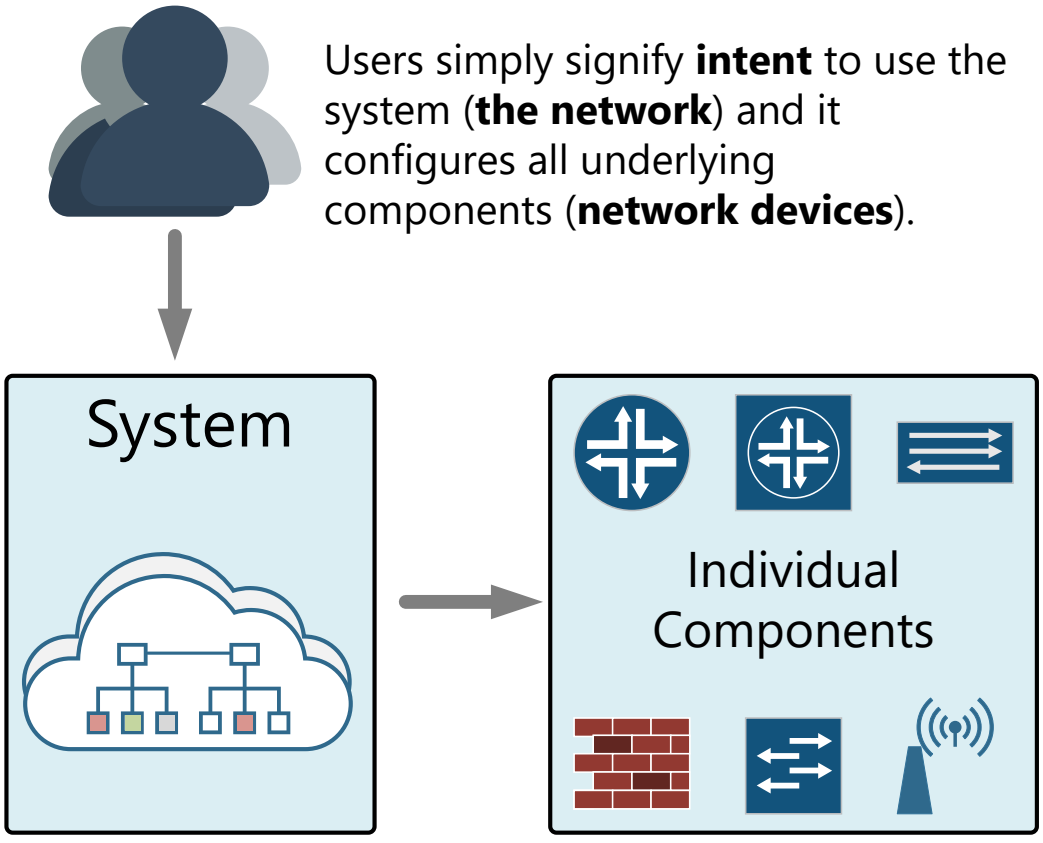


Figure 5. Using Networks in an Intent-Based manner

Let's look at the most obvious advantages:

- **Automation -** Operating and administrating a network as a system in a centralized manner removes most of the complexity of large-scale networks. The whole idea of centralized management is to use Automation. It creates simplified consistent deployment and operational models.
- **Reduced cost -** SD-WAN's automation allows leveraging any combination of transport services such as Broadband Internet circuits, 4G/5G, MPLS, and any other public transport - to securely connect users to applications.
- **Improved uptime -** Centralized management and automation could eliminate most human errors such as misconfigurations, wrong designs, and deployments. This would significantly improve the overall uptime of the network.
- **More secure -** Centrally managed networks can easily apply end-to-end security policy across the enterprise network which is very hard to do using the box-to-box approach.
- **Better analytics -** To be honest, in traditional large-scale networks, very few people (often nobody) know and could grasp the big picture. Because of the high number of network devices and the large volume of analytics data, most of the time the data is very hard to read and interpret quickly. Treating a network as one system enables a single management console that presents data from multiple sources in a unified display.

Forwarding based on auxiliary information

In traditional WAN networks, routers make forwarding decisions based on a limited set of information. Traditional routing protocols usually consider only the link bandwidth and link status. However, having multiple different WAN transport attached to a remote site and wanting to use them in an active-active manner requires a more complex routing forwarding process. I have read one of the best analogies of this in the Cisco book "Cisco SD-WAN Cloud scale architecture".

Consider the analogy of traveling by car. Prior to the emergence of navigation software such as Google Maps, for the travel from New York to Boston, a paper road map was typically used to identify the best route. If there was road closure or delay along the route, the driver would be forced to find an alternative route based on limited information. This is the way WAN routers operate in a traditional WAN network. Each router makes its own autonomous decisions about how to route the packets, based on a limited view of the topology around it.

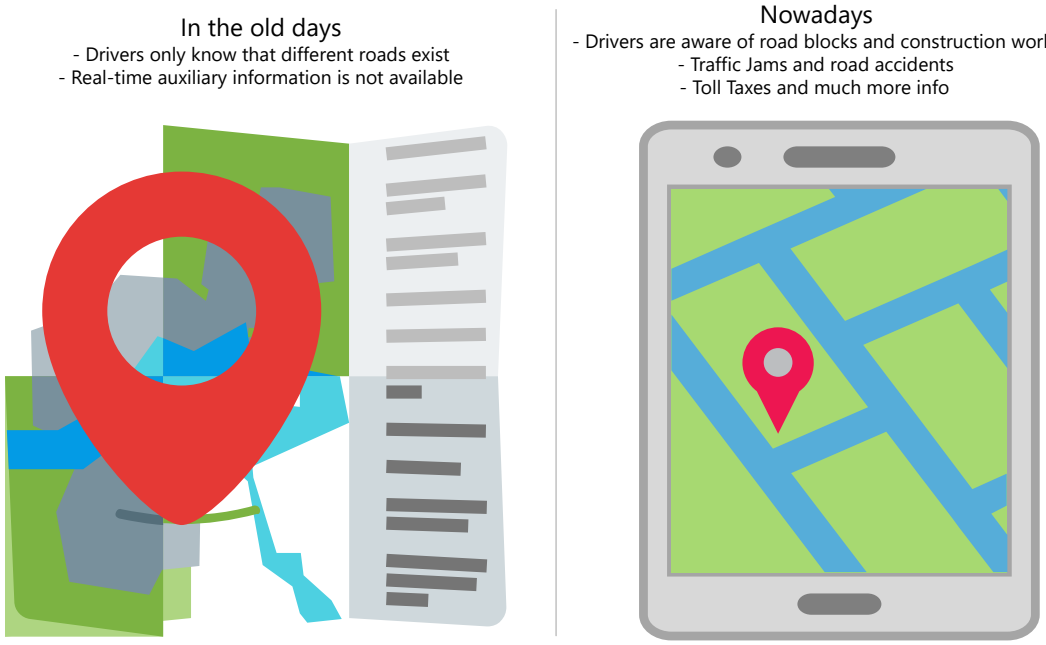


Figure 6. Analogy between routing decisions and road navigation

Now compare this approach to today's road navigation with GPS. Navigation software such as Google Maps can help a driver to avoid road closures, accidents, travel delays, and inefficient routes. This is possible because the navigation software relies on satellites in the sky that have a real-time sophisticated view of the road network. With SD-WAN, edge routers can now rely on the centralized control/management plane for auxiliary information on how to forward the traffic. In the same way, as the GPS helps drivers avoid travel delays, SD-WAN helps routers avoid jitter, packet loss, and latency in the network.

Cisco's SD-WAN solutions

Cisco offers two different SD-WAN products through its acquisitions of Meraki and Viptela. Both products are full-fledged SD-WAN solutions and have several overlapping features. However, Cisco has made it clear that Meraki and Viptela are geared toward two different markets.

- **Meraki** is designed for small and mid-sized companies that want simplicity and ease of use above everything else. Deploying the Meraki SD-WAN solution is easier than Viptela and if the organization does not have any specific niche requirements, it would definitely be the right choice.
- **Viptela** has more advanced features available and requires a sophisticated network design and architecture. The product is designed for large-scale enterprise-level networks and has a high degree of customization.

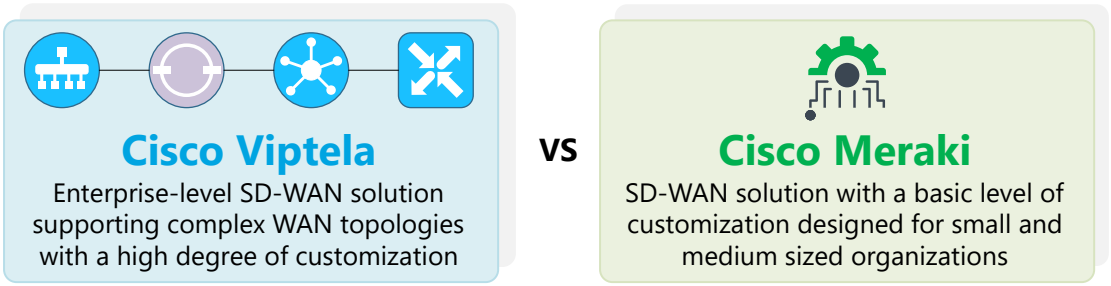


Figure 7. Cisco's SD-WAN solutions

In this course, we are going to deep-dive into the Cisco Viptela SD-WAN and won't cover the Meraki product.

What is SD-WAN?

Traditional WAN was designed to route traffic from remote sites to the company's data centers using private MPLS circuits. However, the business trends have moved the applications out of the data center and into the public clouds such as Microsoft Azure and Amazon Web Services (AWS). Nowadays, moving the users' traffic from branches to the enterprise DC and then out to the cloud or the Internet and back is inefficient, expensive, and not scalable. Also, the rapid digital transformation of enterprises creates new requirements for security, cloud and Internet connectivity, WAN management, and application performance.

Cisco SD-WAN is a Wide Area Network (WAN) overlay architecture that applies the principles of Software-Defined Networking (SDN) into the traditional WAN. It is designed to meet the needs of modern enterprise applications and the rapidly growing security requirements.

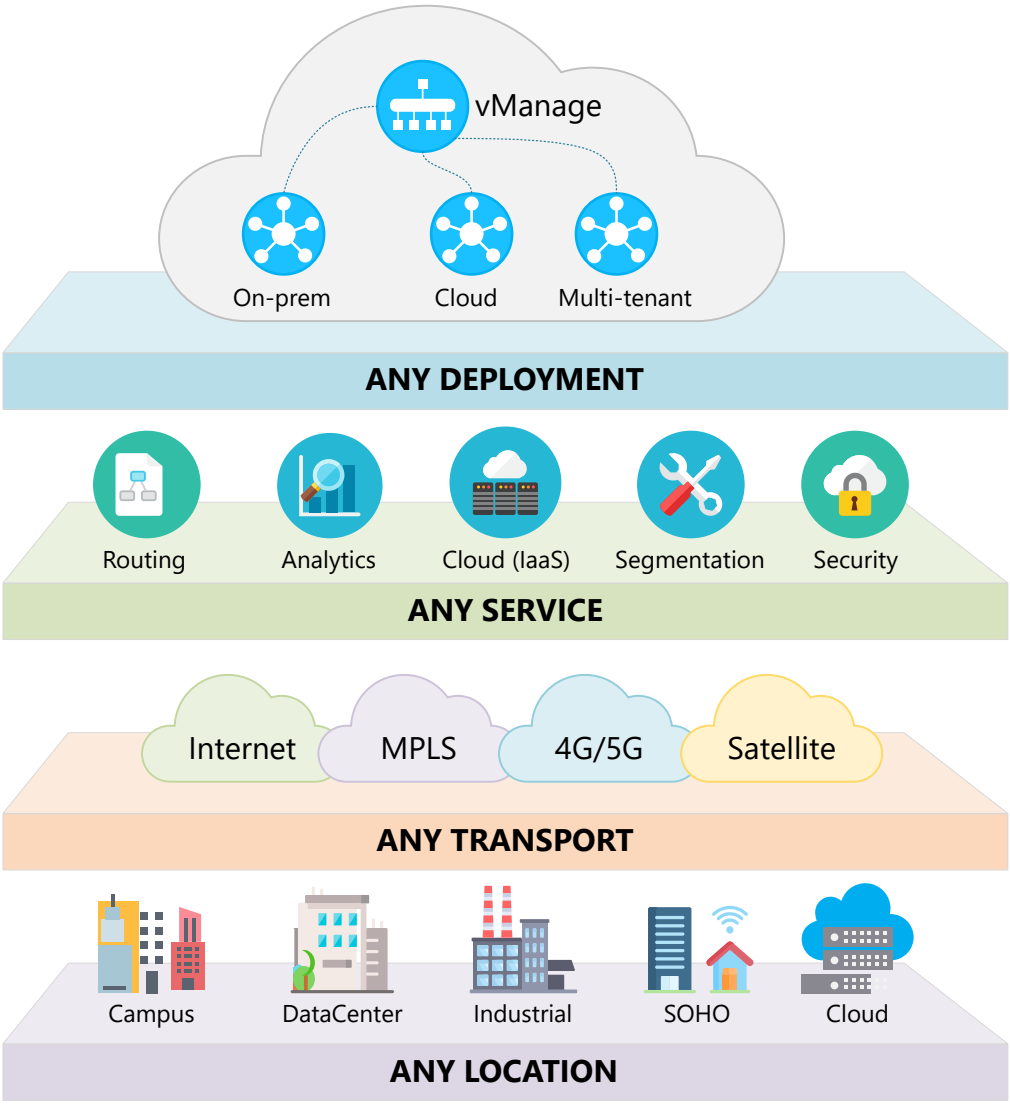


Figure 1. Cisco SD-WAN Architecture

Cisco Viptela SD-WAN solution provides the following improvements over the traditional WAN design:

- Connecting any location in a fast, secure, and highly available manner using Zero-Touch Provisioning (ZTP).
- Establishing a transport-independent WAN using any type of underlying transport.
- Abstracting the underlying WAN infrastructure away from the services and applications that run over the network such as WAN Routing, Segmentations, Analytics, IaaS, and Multitenancy.
- Providing end-to-end security from remote sites to the Internet, Cloud, and SaaS applications.
- Providing a single pane of glass (SPOG) for management, analytics, and configuration policy across the enterprise WAN.
- Providing southbound REST APIs that enable enterprises to create their own unique services and meet any niche requirements.

Figure 1 summarizes the key architectural improvements over the Traditional WAN design. Let's now look at the components of the Cisco SD-WAN solution.

SD-WAN Components

Cisco Viptela SD-WAN solution is made up of four segregated planes - Orchestration plane, Management Plane, Control Plane, and Data Plane. Each plane has its own functions and responsibilities and is abstracted away from the other planes. For example, if you replace a device in the data plane, that does not affect the control/management or orchestration plane. The same applies if you replace a controller in the Control plane or the Management Plane.

Compare this to the Tradition WAN design where each device participates in the data plane (forwarding actual packets), in the control plane (for example running OSPF, BGP, PIM and participate in the topology formation), and in the management plane (is actively managed via CLI).

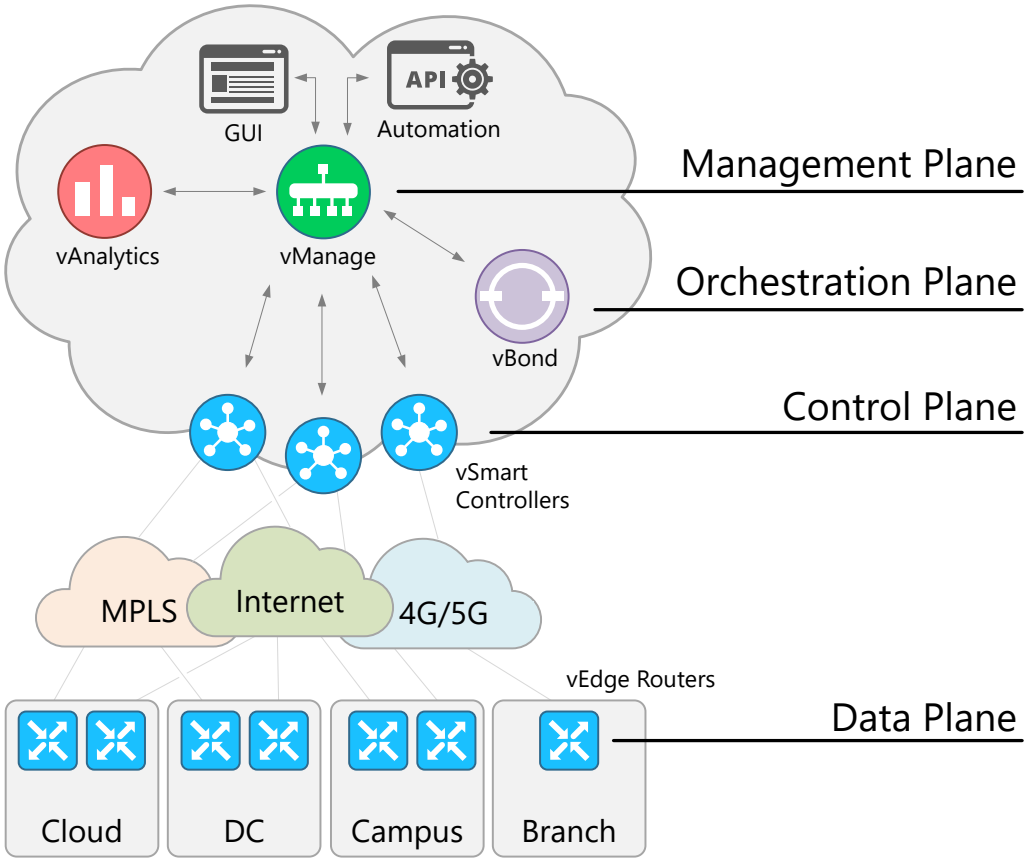


Figure 2. Cisco SD-WAN Components

Cisco vManage

Cisco vMange is the Management Plane of the SD-WAN system. It runs the user interface of the system and is the dashboard network administrators interact with daily. It is responsible for collecting network telemetry data, run analytics, and alert on events in the SD-WAN fabric. It is also the tool that admins use to create device templates, push configurations, and perform overlay traffic engineering.

Cisco vManage can be deployed on-prem, in the public cloud, or in the Cisco cloud-hosted environment. It is significantly resource-intensive and most customers go with the cloud options.

Cisco vBond

Cisco vBond is the Orchestration Plane of the SD-WAN system. Its job is to orchestrate the process of onboarding new unconfigured devices to the SD-WAN fabric. It is responsible for the authentication and whitelisting of vEdge routers and control/management information distribution.

Cisco vSmart

Cisco vSmart is the Control Plane of the SD-WAN system. vSmart controllers are the brain of the overlay fabric. They advertise routing, policies, and security. They are positioned as hub routers in the control plane topology and all vEdge routers peer with all vSmart controllers. For experienced network engineers, vSmart controllers are like BGP Route-reflectors or DMVPN NHRP routers. However, it is important to understand these appliances are not part of the Data Plane and do not participate in packet forwarding.

Cisco vEdge

Cisco vEdge devices represent the Data Plane of the SD-WAN system. They sit at the WAN edge and establish the network fabric and join the SD-WAN overlay. If you look at the architecture shown in figure 1, everything southbound of the vEdge routers is typically traditional networking - offices, data centers, and branches. Everything northbound of the vEdge routers is the SD-WAN system itself. vEdge routers exchange routing information with the vSmart controllers over the Overlay Management Protocol (OMP). If for example, we have a campus network running OSPF. At the vEdge devices, the OSPF routes are redistributed into the SD-WAN fabric to the vSmart controllers via OMP and then the vSmart controllers populate this routing information to other vEdge devices if it is required by the WAN topology.

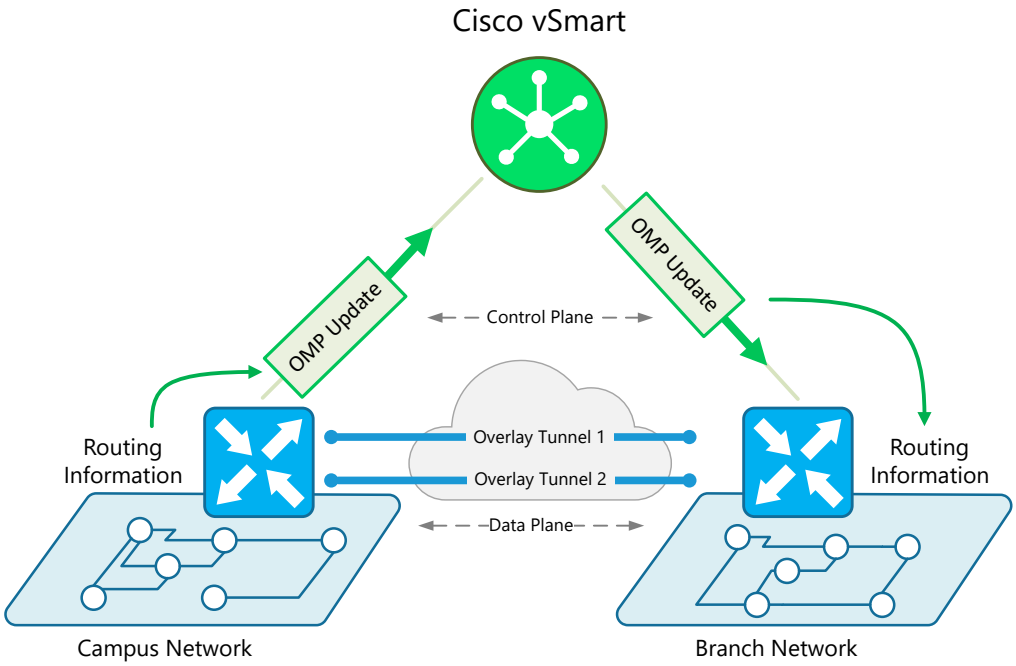


Figure 3. Cisco SD-WAN OMP Protocol

The WAN Edge routers could be Viptela platforms or Cisco IOS-XE devices. They can be virtual or physical appliances. vEdges are auto-configured by the system. Back in the Viptela days, this process was called Zero-Touch Provisioning (ZTP) and nowadays with the Cisco devices, it is called Cisco Plug-and-Play (PnP). Both terms actually mean the same and are interchangeable.

Overlay Management Protocol (OMP)

The Cisco vSmart controllers use the Overlay Management Protocol (OMP) to manage the overlay network fabric. Upon joining the SD-WAN fabric, each vEdge router establishes one permanent secure connection to the vSmart controller via each available transport as shown in figure 4. These connections, usually DTLS, are then used by the vEdges to exchange control plane information to the controller such as prefixes, crypto keys, and policy information.

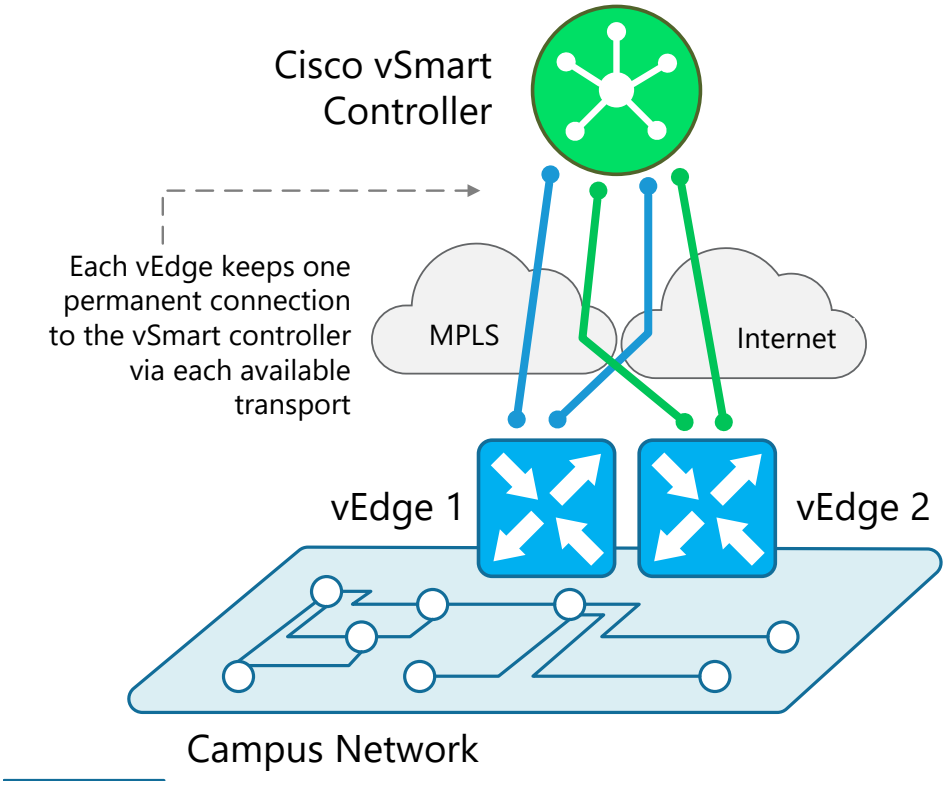


Figure 4. Cisco vEdges OMP peering

It is important to note that OMP peering is never made between the vEdge routers onsite. This is due to the separation of control and data plane in the SD-WAN architecture.

Three types of routes are advertised with OMP:

1. OMP routes (vRouter) are prefixes at the local site that are redistributed into OMP and advertised towards the controllers. These might be OSPF or BGP routes, or any other routing information present on the site.
2. TLOC routes (Transport locations) are the tunnel endpoints on the WAN Edge routers that connect to the transport networks. These routes are represented by three components- the system IP address, link color, and encapsulation type.
3. Service routes are used to exchange services such as firewall, IPS, application-specific optimizations, and load-balancers.

Let's leave the things here and continue with our exploration of the Cisco SD-WAN architecture in the next lesson.

Key Takeaways

Let me try to summarize into a short table what the Software-Defined WAN brings compared to the Traditional WAN design.

	Traditional WAN	Software-Defined WAN
Integration	Hardware Centric	Software Centric
Administration	Manual box-to-box	Automated
Extension	Closed	Programmable via REST APIs
Operations	Reactive	Predictive
Drivers	Network Intent	Business Intent

Differences between the Software-Defined WAN and Traditional WAN

How Cisco SD-WAN works?

In the last lesson, we looked at the main architectural components of the Cisco SD-WAN solution. Now in this lesson, we are going to try to explain how everything fits together to create an operational overlay fabric that can move user flows more securely and intelligently than Traditional WAN.

SD-WAN Deployment

When a company decides to migrate its traditional WAN architecture to Software-Defined WAN, the thing that always comes first is to deploy the controllers. The next step is to migrate the main data centers and hub sites and lastly the remote sites such as campuses and branches.

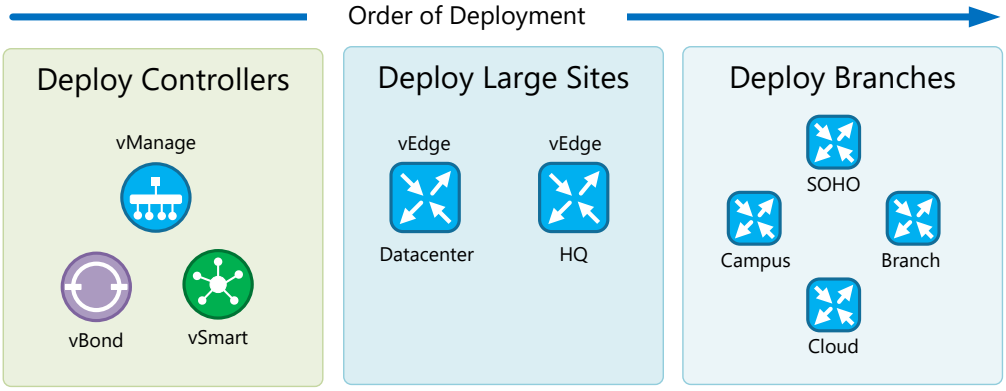


Figure 1. Cisco SD-WAN Order of Deployment

The main idea for doing it in this sequence is to have the hub sites route the traffic between the SD-WAN and non-SD-WAN sites for the period of the migration. Of course, if it is a brand new ground-up deployment, the sequence does not matter that much.

Controllers Deployment Options

One of the main advantages of the Software-Defined WAN is that the controllers can be deployed in the public cloud. This can significantly reduce the CAPEX/OPEX costs and improve the overall availability and redundancy of the management plane/control plane. Compare this model to the scenario in which you have all controllers deployed on-premises. You need to accommodate rack space, power, cooling, physical servers, hypervisor, and virtual machines or containers. You have to manage redundancy and backups on your own. Using the cloud options, you can consume the management/control plane as IaaS (Infrastructure-as-a-Service) or even SaaS (Software-as-a-Service).

Cisco offers the following options to customers to choose from:

- **Cisco-hosted cloud -** The information that I have found regarding the existing deployments shows that most customers (above 90%) opt for this approach. This is also the vendor's recommended model because Cisco takes care of provisioning all controllers, they handle the backup and disaster recovery. The customer is basically consuming the SD-WAN control plane as a Software-as-a-Service (SaaS) by using the vManage to create custom configuration templates for their device and administer the overlay fabric.
- **Public cloud -** The customer could decide to host the controllers in the public clouds such as Azure and AWS. In this scenario, the controllers could be managed by a service provider or by the customer.
- **On-prem -** Of course, the controllers can be deployed in the company's data centers or private clouds. In this scenario, the customer is responsible for backups and disaster recoveries. This is usually the case with financial and government institutions that must be compliant with regional regulators.

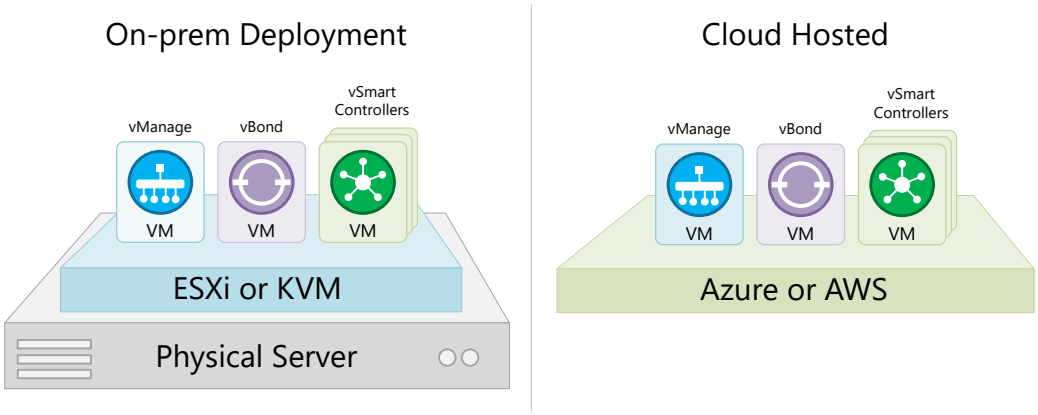


Figure 2. Cisco SD-WAN Deployment Options

Once the controllers are up and running, they must establish secure connections between them. As of 2020/2021, there are two options to choose from when it comes to the underlying secure protocol - TLS which uses TCP transport, and DTLS which uses UDP transport. By default, all controllers use the DTLS option.

If the SD-WAN is deployed in a zero-trust environment, figure 3 shows the Layer 4 information for all permanent connections between the controllers. Note that each core on vManage and vSmart makes a permanent DTLS connection to the vBond resulting in four connections between vManage and vBond and two connections between vSmart and vBond.

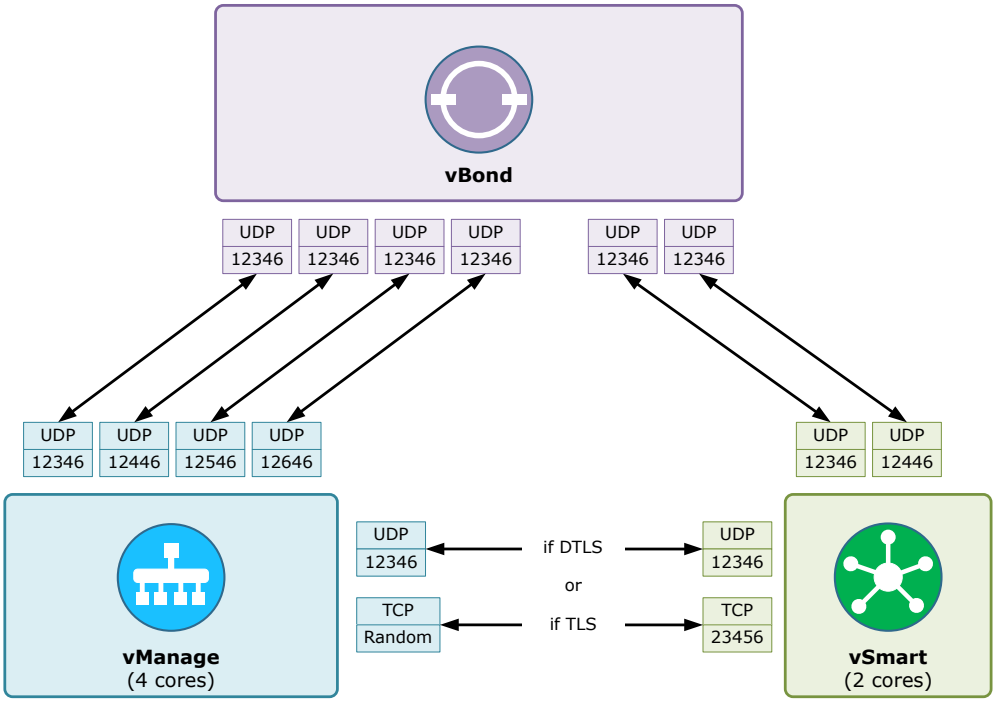


Figure 3. Cisco SD-WAN Control Connections

WAN Edge Routers Onboarding

Secure onboarding of the WAN edge devices is a very important part of the SD-WAN solution.

Identity and Whitelisting

The Cisco SD-WAN solution uses a whitelisting model for authenticating and trusting the vEdge devices. This means that before a WAN edge router is allowed to join the control plane, it has to be known by all SD-WAN controllers beforehand. Each device is uniquely identified by its Chassis ID and certificate serial number.

Controllers reachability

Once the SD-WAN controllers are deployed and have valid certificates, WAN edge routers can start the onboarding process. At this point, the most important thing is to make sure that the vEdge appliances have reachability to all controllers via all available transports. It sounds like an easy and straightforward step at first, but when you look more deeply into it, you can see that there are some decisions to be made.

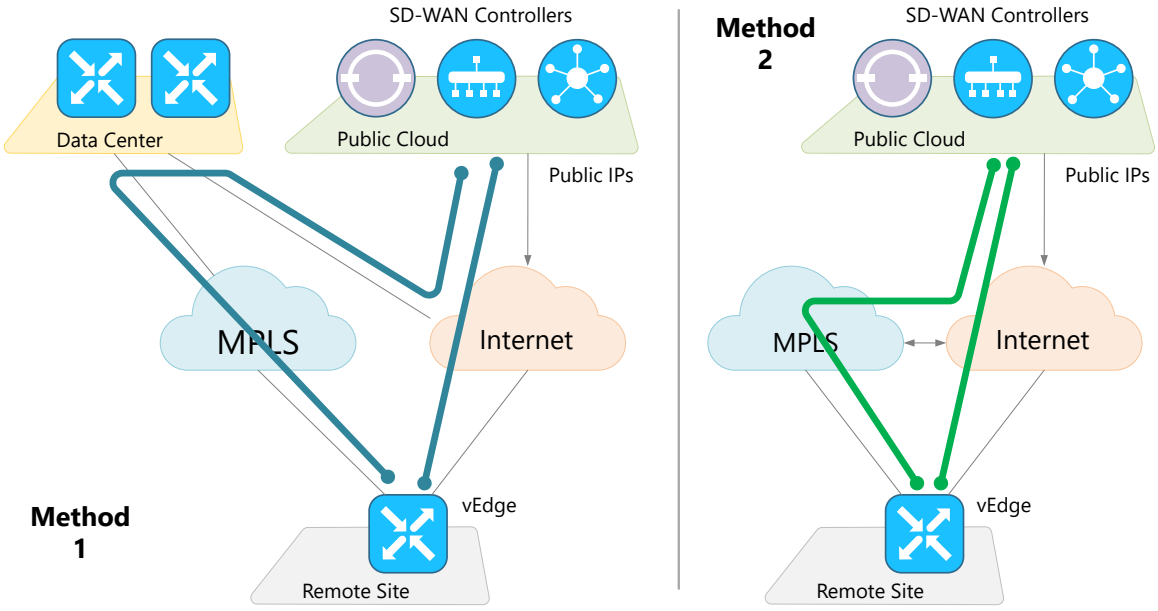


Figure 4. Control Plane Communications

The Edge device tries to establish a control connection over each provisioned transport, first initiating one to the vBond orchestrator before attempting to connect to the other controllers. All available transports are tried one at a time, starting with the WAN connection to the lowest interface number. Let's look at the typical scenario where a remote site has one private MPLS circuit and one broadband Internet connection. The WAN edge device would try to connect to controllers over the Internet and the MPLS line. But if the controllers are deployed in a public cloud or any other 3rd part cloud, would the public IP addresses of the controllers be reachable over the MPLS circuit by default? I guess not, there is no MPLS service having all public prefixes redistributed into it.

There are three common implementations that solve this issue:

1. The MPLS has reachability to the public cloud by being routed through a data center or regional hub that has both transport. This method is shown on the left side of figure 4.
2. The public routable IP addresses of the controllers are redistributed into the MPLS cloud and the provider edge router advertises them to the vEdge. This method is shown on the right side of figure 4.
3. It is possible to establish a control plane connection through the Internet connection only. The Edge would be able to join the SD-WAN fabric but would not have a control plane redundancy, so this approach is not recommended at all.

Onboarding process

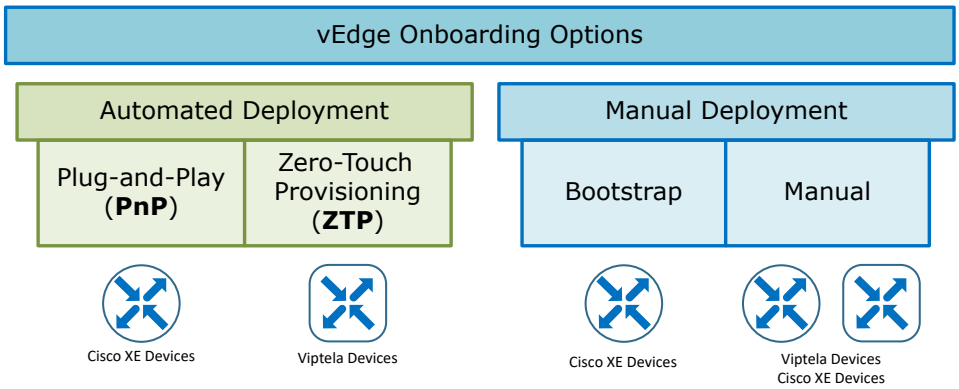


Figure 5. WAN Edge Routers Onboarding Options

Joining the overlay fabric

Let's now put everything together and follow each step of the process of joining a WAN edge device to the overlay fabric. Let's say that for this example we are going to use a Viptela vEdge 1000.

- Step 0. IP Reachability - Upon bootup, the vEdge device obtains an IP address, default gateway, and DNS information via DHCP. If no DHCP service is available onsite, this information could be configured manually using CLI or using a configuration template.
- Step 1. Zero-Touch Provisioning - The WAN Edge router tries to reach the ZTP server by resolving the URL ztp.viptela.com and uses HTTPS to get information about the SD-WAN vBond orchestrator along with the organization name.
- Step 2. Authentication - The WAN edge device authenticates to the orchestrator with its root-certificate and serial number. If the authentication is successful, the vBond sends back the vManage and vSmart
- controller information.
- Step 3. Connection to the Management Plane - The Edge then establishes a secure connection to the vManage and downloads the configuration using NETCONF.
- Step 4. Connection to the Control Plane - If all previous steps are successful, the router establishes a secure connection to the vSmart controllers and joins the SD-WAN overlay fabric

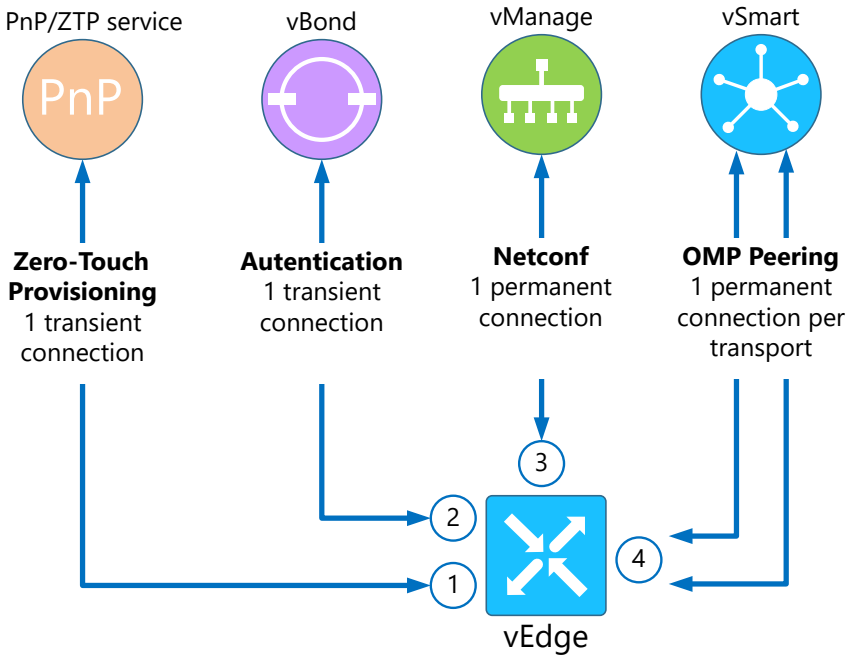


Figure 6. Cisco SD-WAN vEdge Connections

The process of jointing the SD-WAN overlay is visualized in Figure 6. Note that some secure connections are transient and some are permanent.

SD-WAN Operation, Administration, and Management

The beauty of the Software-Defined WAN is that it is operated as-a-System rather than the traditional box-to-box approach. This opens up a whole new world of possibilities when it comes to the Operation, Administration, and Management (OAM) of the solution. Some of the main benefits are:

- **Centralized management -** and operational simplicity, resulting in reduced change and deployment times.
- **Transport-independent overlay -** Because the underlay transport is abstracted away from the overlay fabric, any combination of transports can be used in an active/active fashion. This significantly reduces the bandwidth costs of the company.
- **Sophisticated security -** If you compare the traditional control plane security of OSPF and BGP to the control plane encryption of the SD-WAN, the latter is obviously more comprehensive using certificate identity with a zero-trust security model.
- **Application visibility -** Real-time analysis and application visibility are a core part of the solution. This enables the enforcement of service-level agreements (SLA) and tracking of specific performance metrics

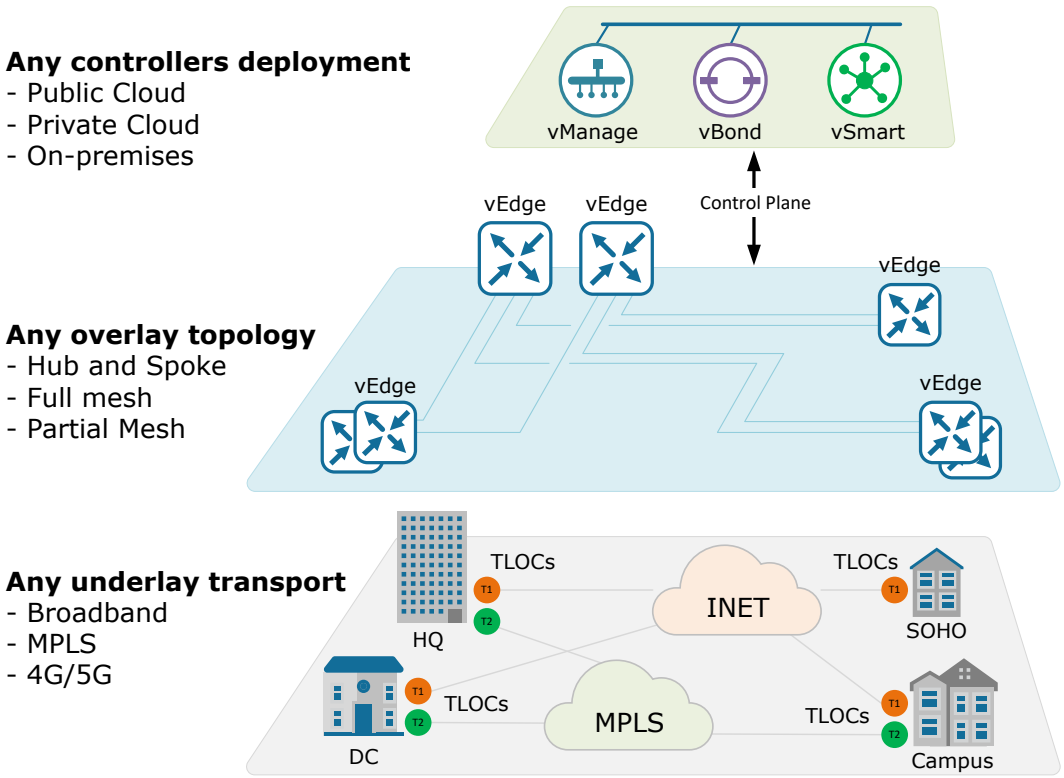


Figure 7. SD-WAN Established Overlay

We obviously won't be able to go deeply into any step of the process of establishing the overlay fabric. However, this is a short summary of how Cisco SD-WAN works. We are going to deep-dive into each step of the process in the next sections of this course.

2. KEY FEATURES OF CISCO SD-WAN

Application Quality of Experience (AppQoE)

From a network perspective, delivering an optimal application experience has never been an easy task. Traditional networks have been designed to move packets without caring too much about the applications.

As a network engineer myself, I have managed many Traditional WAN environments and know how little visibility and control over the app flows there is. The task has got a lot harder with the introduction of the Internet-native and Cloud-native apps. In traditional WAN deployments that are operated in a box-to-box fashion, typically there is 3, 5, or 8 queues QoS policy applied on the egress transport interfaces and that's about it. Is it enough though? The answer is obviously not.

The Business Need

Let's imagine that you are a network engineer in a big enterprise that has a traditional WAN architecture. One day the company starts streaming a real-time event from a remote branch to the data center and out to the Internet. From the company's perspective, this video stream is a business-critical application and must work as optimally as possible. Just think for a few seconds on each of these real-world problems.

- How do you make sure that if the quality of a circuit drops and packet-loss appears, the stream is immediately sent over another circuit?
- What if there is congestion on one of the WAN links. How do you make sure that low-priority traffic is automatically moved to lower bandwidth links?
- What if the stream viewers are mostly on the Internet. Maybe it is a better solution to just steer the traffic directly to the Internet from the remote branch. How do you do that with traditional WAN?

These are just a few examples of hard problems that cannot be easily solved at scale using the legacy box-to-box operational model. In today's enterprises where everything is getting digital, there are at least several business-critical applications. Ensuring optimal experience for these applications is a multidimensional problem that cannot be solved with a single network tool such as QoS.

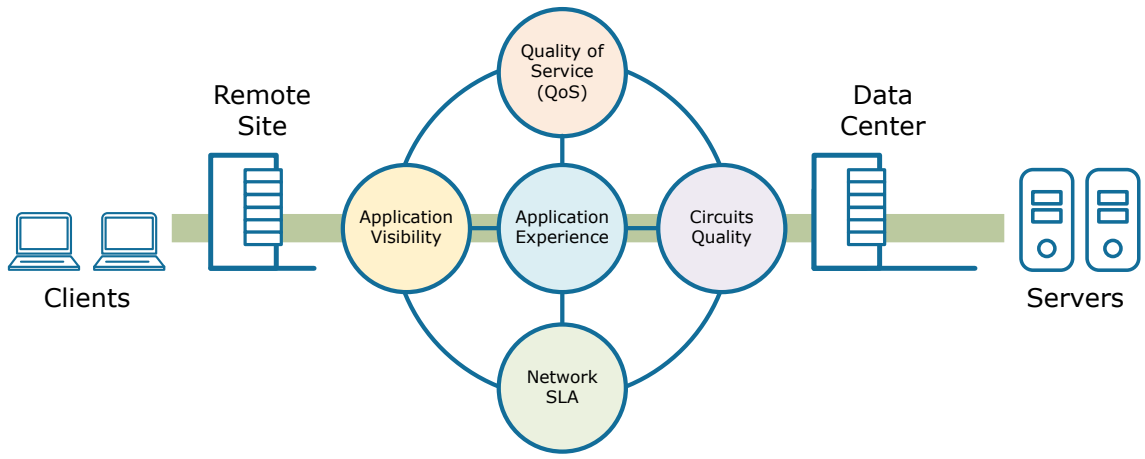


Figure 1. Cisco SD-WAN Tools for improving application experience

Cisco SD-WAN solution has a set of capabilities that can improve the overall Application Quality of Experience (AppQoE) of business-critical applications. The package of tools includes some well-known network protocols working in conjunction with some innovative technologies to create the following collection of AppQoE features:

Bidirectional Forwarding Detection (BFD)

- Quality of Service (QoS)
- Forward Error Correction (FEC)
- Packet Duplication
- Fragmentation Avoidance
- Software-Defined Application Visibility and Control (SD-AVC)
- Application-aware routing (AAR)
- TCP Flow Optimization
- Cloud onRamp for SaaS (We will have another lesson dedicated to this feature, so it is just briefly mentioned here)

You may have dealt with some of these features before. In this lesson, we are going to try to briefly go through most of them.

Quality of Service

Cisco SD-WAN solution creates a transport-independent overlay fabric, leveraging tunneling techniques such as GRE and IPsec to encapsulate and encrypt traffic before it is sent over all available circuits on the WAN edge routers. By default, this traffic encapsulation "hides" the original packet inside a new one, and therefore the end-to-end QoS marking is lost. Cisco WAN edge routers have the ability to copy the DSCP value of the original IP packet into the outer IP header. There is also the ability to rewire the DSCP value to match a specific class of service of the service provider's circuit. This gives the ability to map specific applications into the correct QoS classes on the SP side.

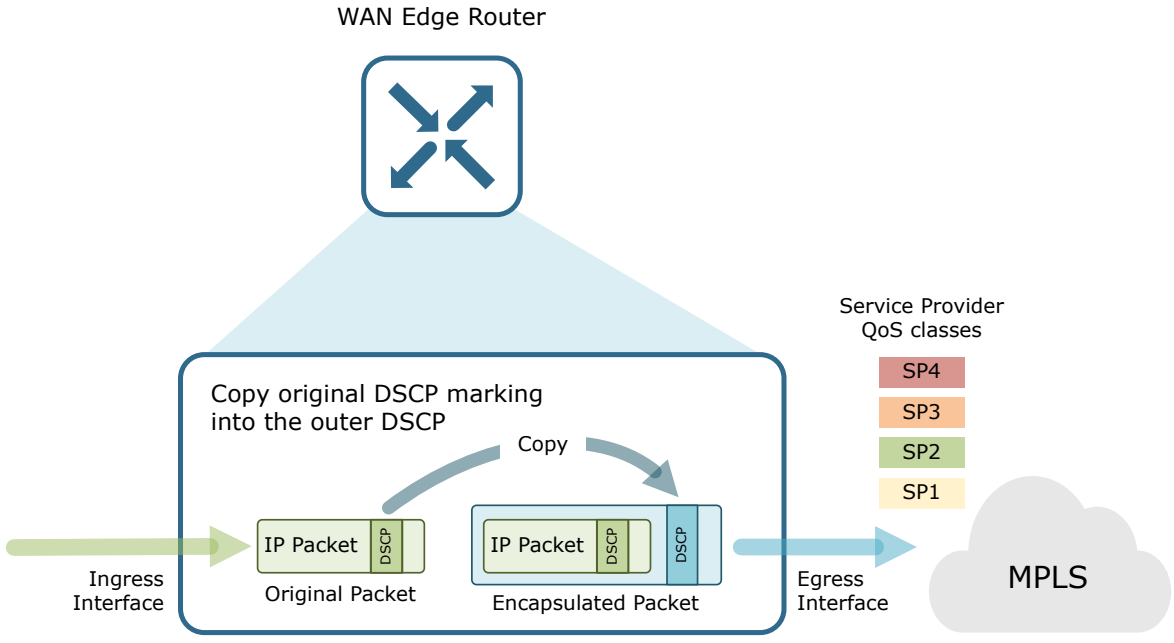


Figure 2. Mapping app DSCPs to service provider classes

Avoiding packet fragmentation

As we all know, the IP protocol was designed for use on a variety of links. In the wide-area networks especially we can see that this is true. We have edge routers connected to the WAN via many different access technologies such as SDH, DSL, Ethernet, LTE, satellite links, etc. Each one of these links may enforce a different maximum transmission unit (MTU) value. Overlay features such as tunneling and IPsec can further lower the MTU. The IP protocol accommodates these differences by allowing devices to break larger packets into a number of pieces that can be reassembled later on. This process is called IP packet fragmentation and leads to inefficiencies in the packet flow by adding unnecessary latency and jitter. MTU issues in traditional WAN are well-known to network engineers.

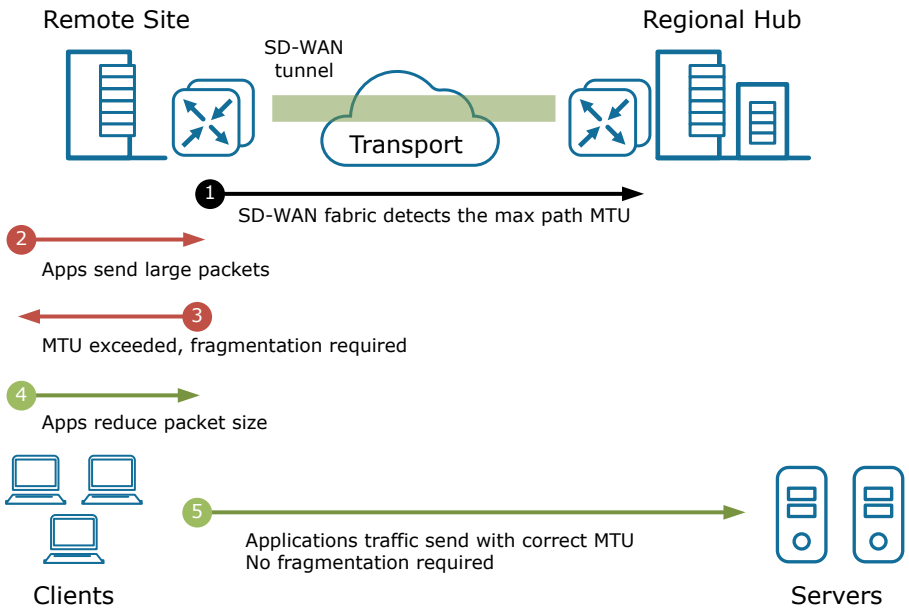


Figure 3. Cisco SD-WAN Path MTU Discovery Process

Applications have the option to explicitly prohibit fragmentation by setting a DF (do-not-fragment) flag in the IP header. But this relies on the success of another process called path MTU discovery (PMTU) that is used to discover the smallest end-to-end MTU value along the traffic path. If this process fails and the DF-bit is set, the application flow would not be able to traverse the network and reach its destination. Cisco SD-WAN proactively discovers the path MTU across the overlay fabric and participate in the hosts' PMTU process by notifying them of the available MTU as shown in Figure 3.

Circuits Quality

One of the main advantages of SD-WAN is that it can use any available transport at any location in an active-active fashion. This typically means that all Internet links are utilized for application traffic. However, we all know that Internet circuits do not have guaranteed quality, and packet-loss may occur at any given time. Cisco SD-WAN provides the following features that protect business-critical apps from packet loss and allows them to work reliably over the Internet.\

Forward Error Correction (FEC)

The Forwarding Error Correction (FEC) feature allows critical apps to work well over unreliable WAN links usually Internet circuits. The mechanism behind it is borrowed from RAID arrays logic. For example, in a RAID4 array, if one disk fails it can be replaced with a new one and the information can be reconstructed based on the metadata stored in the parity disk. The FEC follows the same logic, for each group of four packets, one "parity packet" is inserted. At the receiver end, if one of the four packets is lost, it can be reconstructed based on the parity metadata. It is basically a trade-off between CPU cycles and circuit reliability. The process is visualized in figure 4.

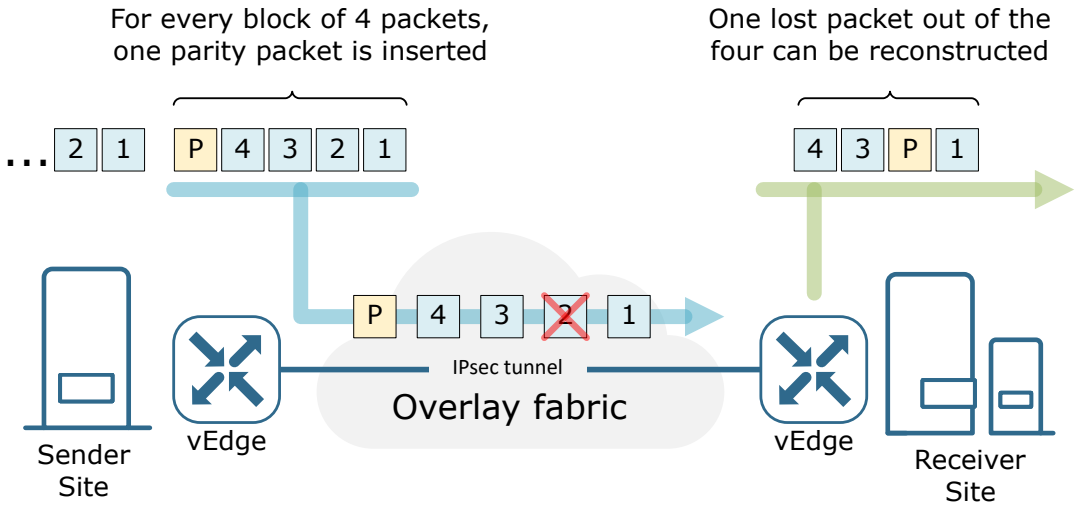


Figure 4. Cisco SD-WAN Forward Error Correction (FEC)

In summary, the FEC capability protects applications from incurring packet loss on the transient network path. The feature has the following characteristics:

- **Per tunnel -** It is enabled on a per tunnel basis. This gives the flexibility to be enabled only on unreliable WAN links.
- **Dynamically invoked -** FEC can be turned on permanently or it can be dynamically invoked if the SD-WAN fabric detects a certain amount of packet loss.
- **Application traffic only -** The feature can only be used for application traffic and not for control plane flows such as BFD.
- **Only one packet out of four can be reconstructed -** It cannot remedy high packet loss.

Packet Duplication

Packet duplication is another SD-WAN capability that is used to increase application reliability. When turned on, the sending WAN edge router can transmit the same traffic flow across multiple WAN links ultimately sending at least two copies of each packet. At the receiving side, the vEdge device can compensate for lost packets by using these multiple copies of the same flow and discard the unnecessary duplicates.

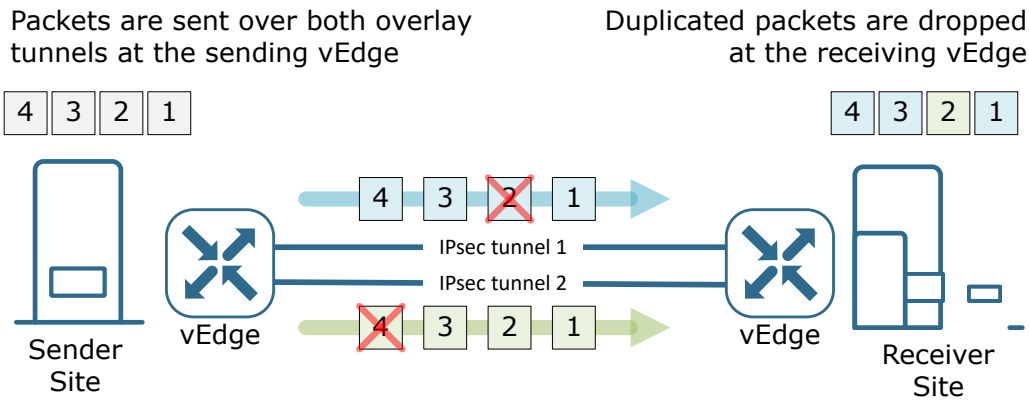


Figure 5. Cisco SD-WAN Packet Duplication

In summary, the Packet Duplication capability protects against packet loss for critical applications such as Voice at the expense of increased bandwidth consumption. The feature has the following characteristics:

- Protocol agnostic - It works for any transport protocols TCP or UDP.
- Works only over multiple tunnels.
- Duplicates are discarded on the receiver.

Software-Defined Application Visibility and Control (SD-AVC)

Software-Defined Application Visibility and Control (SD-AVC) is a service that uses the capabilities of Cisco WAN Edge devices to identify, aggregate, and communicate application data in order to make decisions like prioritizing app traffic using QoS, group applications based on business relevance, or choose different network paths based on real-time SLA statistics.

Cisco SD-WAN devices have a Deep Packet Inspection engine integrated that can go up to Layer 7 of the OSI model and recognize thousands of applications. This is absolutely necessary in order to be able to apply policies against a particular app or service.

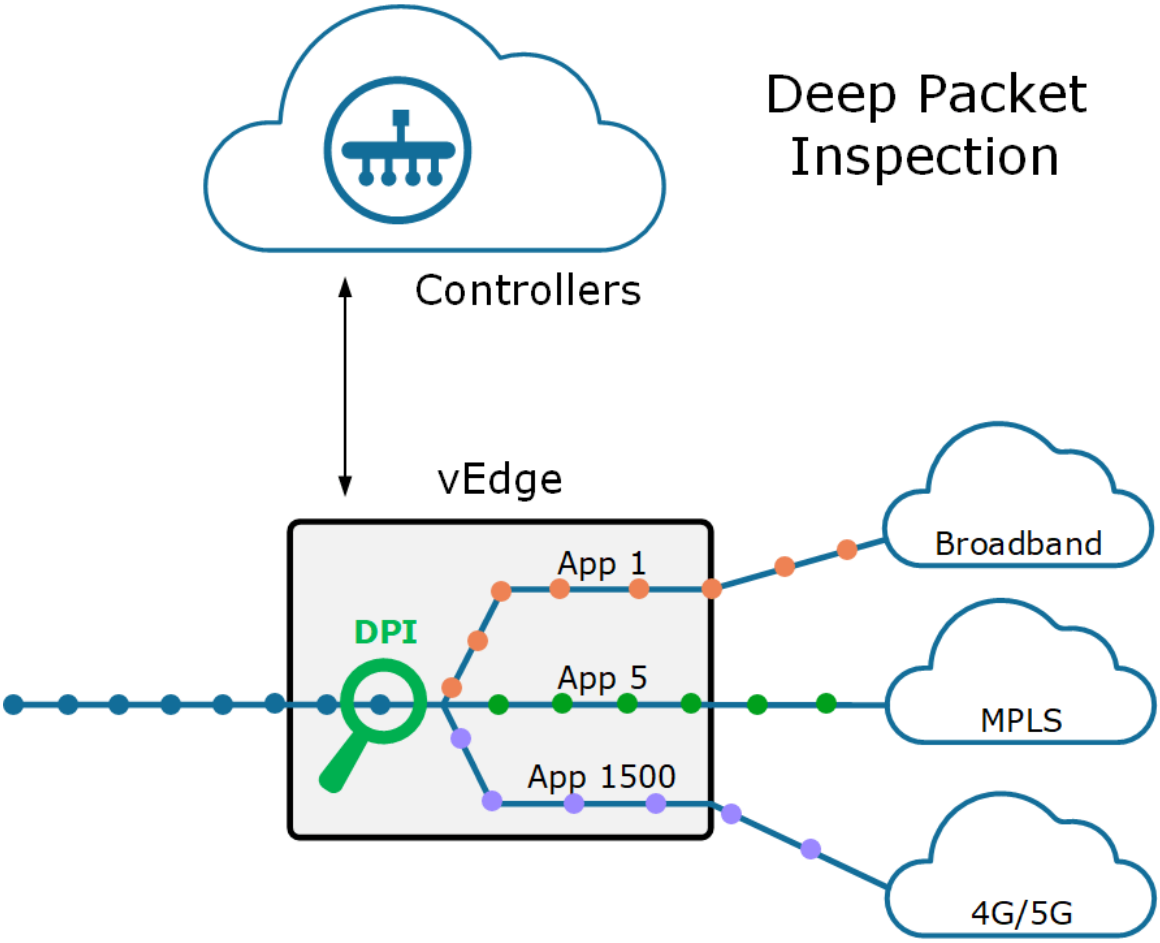


Figure 6. vEdge Deep Packet Inspection

Some engineers may argue that we have always had DPI engines and app recognition features like Cisco NBAR. However, the key point here is that updating large volumes of new application signatures across the device fleet is not feasible at all using the legacy box-to-box configuration model. You have to operate the network As-a-System to be able to do that at scale, and that is what SD-WAN allows us to do.

Application-aware routing (AAR)

Application-aware routing is a feature that dynamically chooses the optimal path for a business-critical application based on a pre-defined SLA policy. These policies can be defined in two major ways:

1. A specific path is configured to be taken while the path meets the SLA. For example, an MPLS circuit is configured as primary for VoIP traffic.
2. Any path that is compliant with the SLA can be used. For example, if the Internet circuit meets the latency, jitter, and packet loss requirements, it can be used for Voice traffic as well.

Let's look at the example shown in figure 7. Application X has a pre-defined SLA policy - latency <= 200ms, packet loss below 3%, and jitter below 15ms. At the moment only path 2 and 3 are meeting this SLA though. Therefore, only these paths can be used for this app.

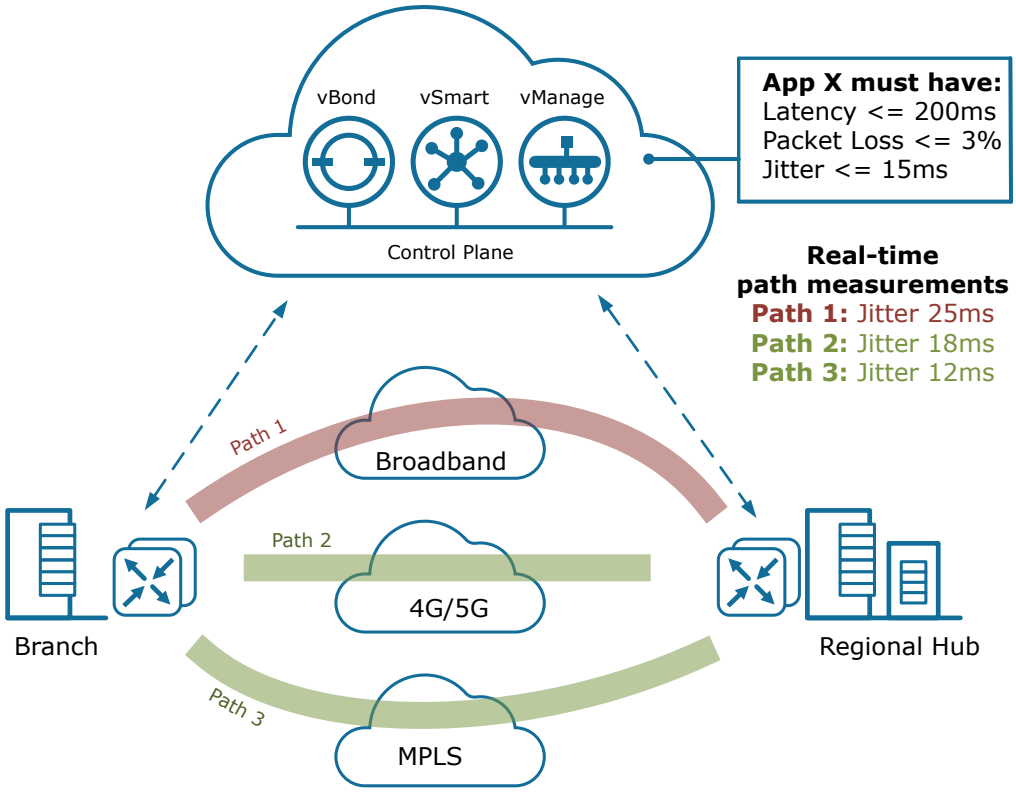


Figure 7. Cisco SD-WAN Application-aware routing

TCP Optimization

The Cisco SD-WAN TCP Optimization feature terminates TCP connections locally at the WAN edge routers and uses TCP Selective Acknowledgment (SACK) in order to better control the TCP-Window-Size and maximize the throughput through the WAN links. Every network engineer has seen a bandwidth consumption graph of a TCP flow. It typically has the following pattern - steep increase, sharp fall down with 50%, steep increase again and 50% fall down again and so on. The goal of this optimization tool is to normalize this graph by dynamically controlling the Window Size. However, this must be used with caution because it breaks some fundamentals network principles such as the end-to-end transport layer connectivity.

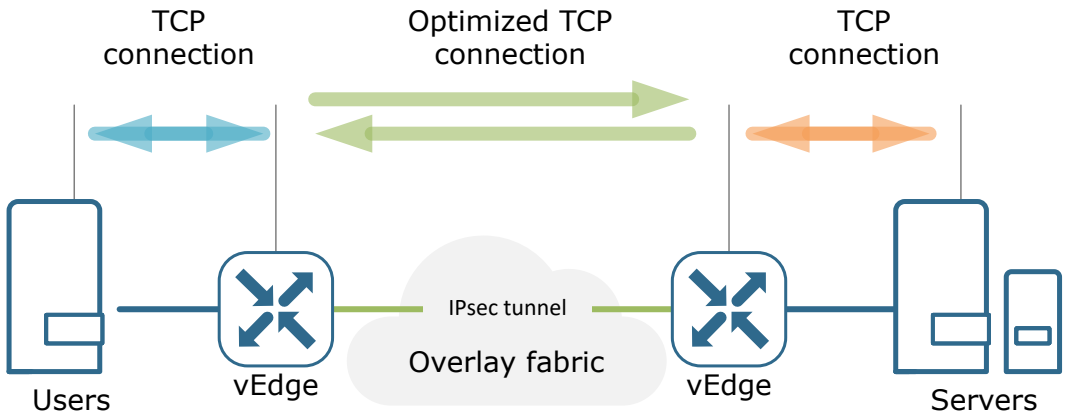


Figure 8. Cisco SD-WAN TCP Optimization

Summary

The Application Quality of Experience (AppQoE) is a multidimensional problem that cannot be solved by a single tool. The Cisco SD-WAN solution has introduced a set of features and capabilities that can improve the overall application experience and optimize the network reliability for business-critical apps. Let me try to make a short summary of all AppQoE tools in the following table.

SD-WAN Feature	Description
Bidirectional Forwarding WAN edge Detection (BFD)	BFD is a well-known network protocol used to detect faults between two devices connected by an IPsec tunnel and to measure the tunnel characteristics.
Quality of Service (QoS)	QoS is a well-known network tool that is used for classification and marking of application traffic.
Forward Error Correction (FEC)	The FEC capability protects applications from incurring packet loss when traversing unreliable WAN links. It works by inserting one parity packet in every group of 4 and then using the metadata in this parity packet to reconstruct any lost one.
Packet Duplication	The Packet Duplication capability protects against packet loss for critical applications such as Voice at the expense of increased bandwidth consumption by sending two copies of each packet via two different WAN links.
Fragmentation Avoidance	The SD-WAN overlay fabric detects the MTU value on all tunnels and helps end hosts successfully identify what MTU value to use.
Software-Defined Application Visibility and Control (SD-AVC)	SD-AVC is a service that uses the DPI engine of Cisco WAN Edge devices to identify aggregate, and communicate application data in order to make decisions like prioritizing app traffic using QoS, group applications based on business relevance, or choose different network paths based on real-time SLA statistics.
Application-aware routing (AAR)	AAR dynamically chooses the optimal path for a business-critical application based on a pre-defined SLA policy
TCP Flow Optimization	This a feature that terminates TCP sessions at the local WAN edge devices and aggregates them into one optimized TCP session. The goal is to better utilize the available WAN bandwidth by controlling the TCP windows size.

Table 1. Cisco SD-WAN features that improve the Application Quality of Experience (AppQoE)

Interconnecting Multiple Clouds

I think every network engineer has already seen that applications are moved to the cloud on a massive scale in recent years. Some apps are even developed and deployed directly in the cloud. These are known as Cloud-native applications or "born in the cloud". This industry shift has created a lot of new challenges to network architectures such as the following:

- How do you ensure that apps are taking the most optimal path to the Cloud? Since most networks have multiple Internet circuits and eventually a direct connection to the particular cloud environment, the best-path selection on a per-application basis is a complex task.
- How do you provide a flexible and secure connection to the Cloud from every remote site? Since most traditional WAN architectures do not allow direct connections from branches to the Cloud over the Internet, the traffic is routed from remote sites to a regional hub/datacenter and then out to the Internet. This adds additional latency and creates bandwidth bottlenecks.

Yes, migrating apps to the public cloud presents new challenges to network engineers. However, there is another, more rare unicorn - interconnecting multiple clouds, that creates an even harder challenge.

The Business Need

Inevitably, when multiple business-critical applications are migrated to the same public cloud, some fundamental questions start to arise:

- **What if the cloud fails? -** Yes, even public clouds experience large-scale outages. Many enterprises have decided to create instances of their most critical apps in an alternative public cloud for redundancy and disaster recovery.
- **Vendor lock-in -** Overtime, using only one provider makes the company heavily dependent on that vendor. Also as a consumer, an enterprise may want to have the freedom to demand custom pricing and have legroom for negotiations. This inevitably leads to a multi-cloud model.
- **Different geographies -** Not every public cloud provider is available in any geographical area in the world. For example, at the time of writing this article, AWS does not have an infrastructure region in Switzerland. This forces some enterprises into the multi-cloud model.

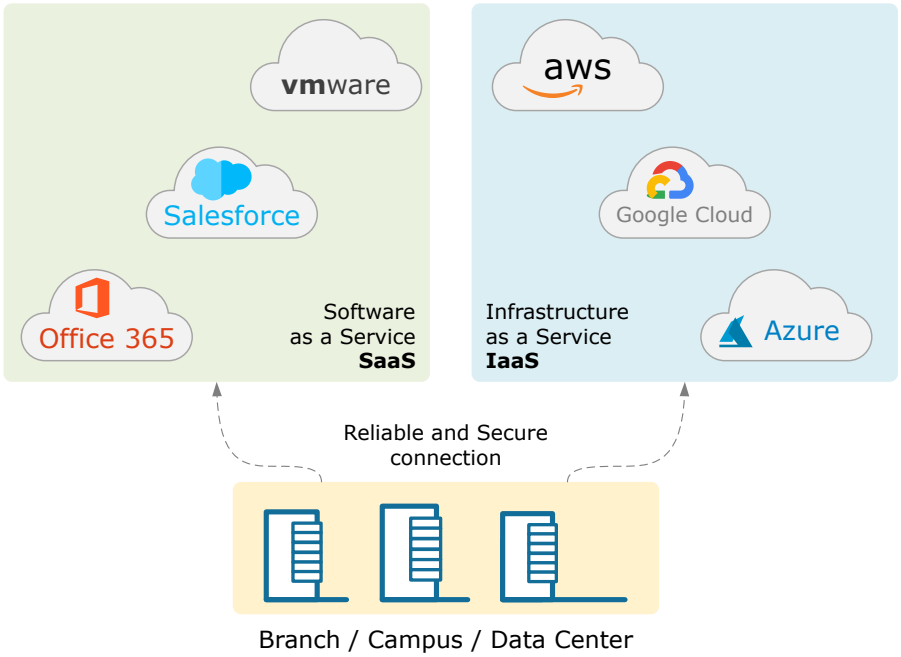


Figure 1. Enterprise Cloud Services

Some companies end up with a multi-cloud operation just naturally. This typically happens when some departments move workloads into one cloud provider and other departments migrate other apps to a different provider.

Cisco SD-WAN as Multi-Cloud Interconnect

Cisco SD-WAN provides the ability to extend the company's WAN to the public cloud, ultimately connecting any WAN location to any cloud platform in a secure and automated fashion.
It ensures the connectivity requirements by using enhanced routing techniques such as application-aware routing to the cloud IaaS applications and adjusting the IPsec routes in real-time based on the pre-defined quality metrics (packet loss, latency, and jitter).

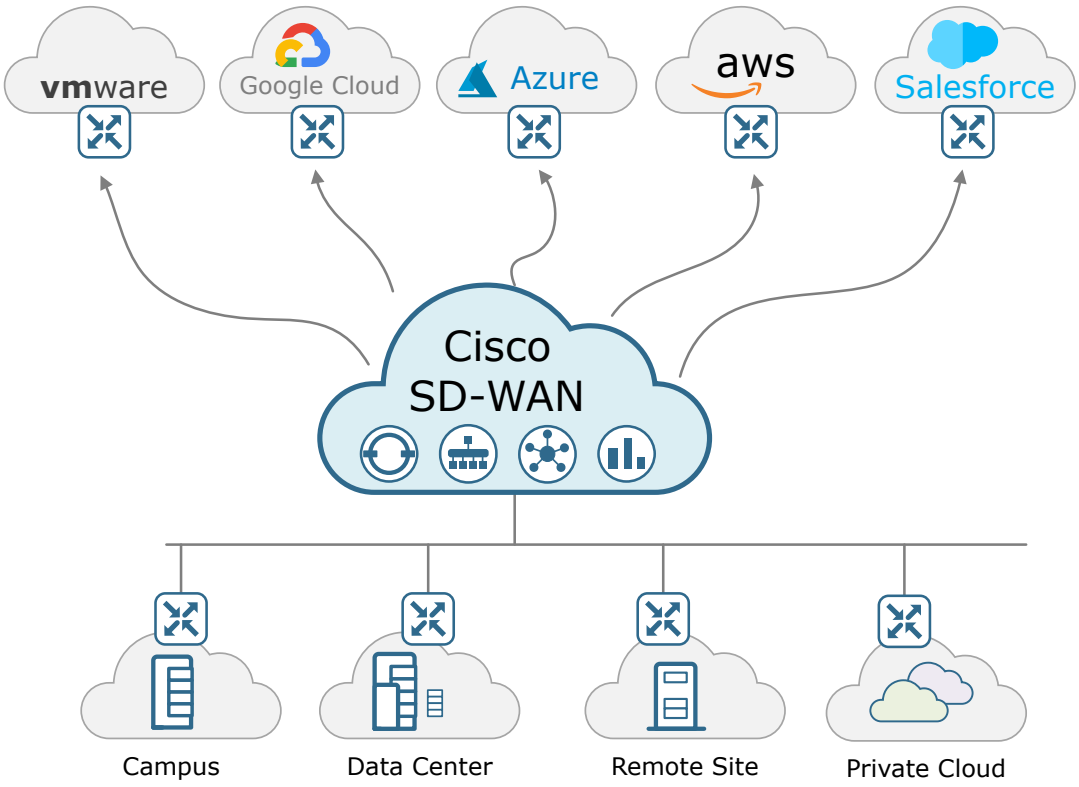


Figure 2. Cisco SD-WAN as Multi-Cloud Interconnect

- **Automated connectivity provisioning -** Cisco SD-WAN extends the overlay fabric to public clouds. This provides the ability to choose the most optimal entry point for all data centers and hub locations in real-time.
- **Application and network telemetry in and out of all clouds for reporting -** Cisco SD-WAN provides the ability to unify the network management by creating a single pane infrastructure that has visibility across the entire enterprise network for more sophisticated management of network resources and services. This single-pane view can provide a unified centralized management of all resources including physical, virtual, and cloud.
- **Dynamic routing, multipathing, and deterministic failover behavior using OMP -** Because the enterprise overlay fabric basically includes the virtual WAN edge routers hosted in the clouds, all network settings could be managed in a centralized fashion using the SD-WAN control plane. This gives the ability to create custom network topologies based on the company's needs.

Direct Internet Access (DIA)

The Business Need

In traditional wide-area designs, Internet traffic from remote sites is sent first to a centralized data center or hub site. Then the traffic is pushed through the company's security stack and only then it is routed out to the Internet. The returning traffic also traverses the security stack before it is sent back to the remote site. This is typically done because the cost of installing and operating a security stack in every remote location is very high.

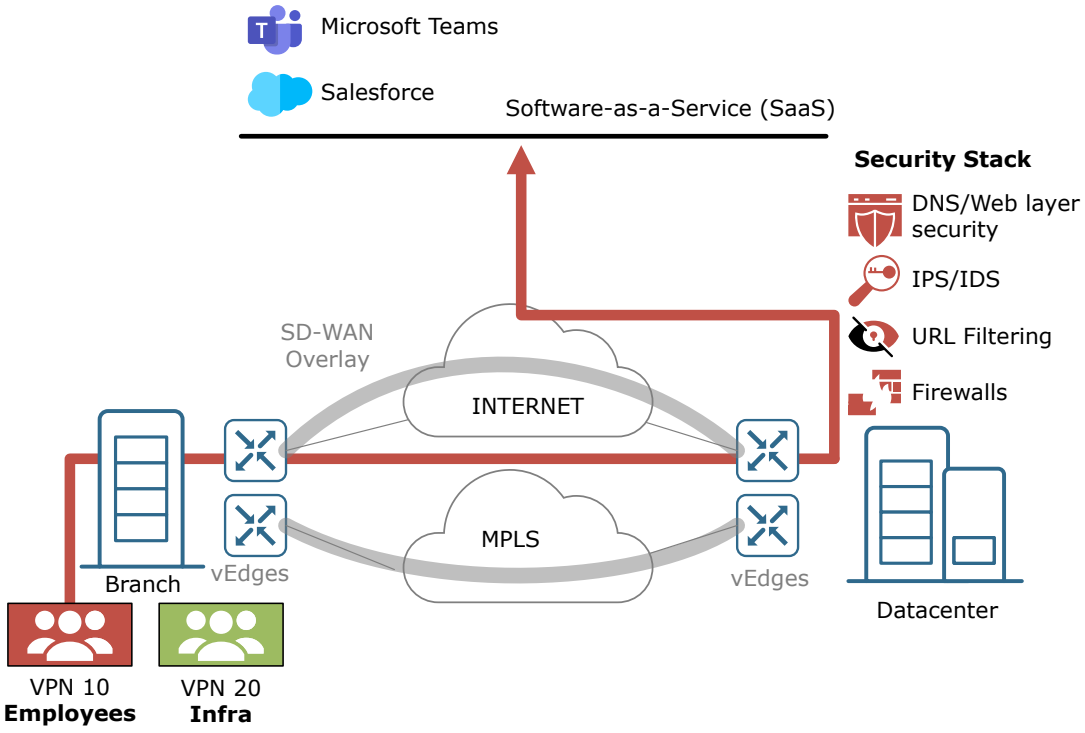


Figure 1. Backhauling Internet traffic through a Datacenter

However, the fast adoption of Internet-based applications and teleworking created the following issues with this WAN design:

- **Scale -** With the ever-increasing Internet demands at each remote branch and the use of SaaS apps such as Office 365 and Salesforce, backhauling the Internet traffic from every branch to the data center creates a bottleneck at the DC's Internet circuits. In addition, the centralized security stack and the network devices at the DC must scale vertically with the number of branches.
- **Apps Performance -** By re-routing traffic from the branch to DC to the Internet and back, applications incur increased latency. Depending on the underlying geography, this can result in significant performance degradation for some business-critical apps.
- **Cost -** Having said the above, it is obvious that the centralized hub site must scale vertically when the number of remote sites increases. This implies higher hardware and WAN costs at the data center location.

Cisco SD-WAN allows for a better more scalable approach to Internet usage at remote sites. The feature is called Direct Internet Access and as the name implies, it allows particular users and applications to access SaaS/IaaS services directly through the local Internet circuits.

Cisco SD-WAN Direct Internet Access (DIA)

Cisco SD-WAN Direct Internet Access is a solution that improves the user experience for SaaS applications at remote sites by eliminating the performance degradations related to backhauling Internet traffic to central data centers. DIA allows control of Internet access on a per VPN basis.

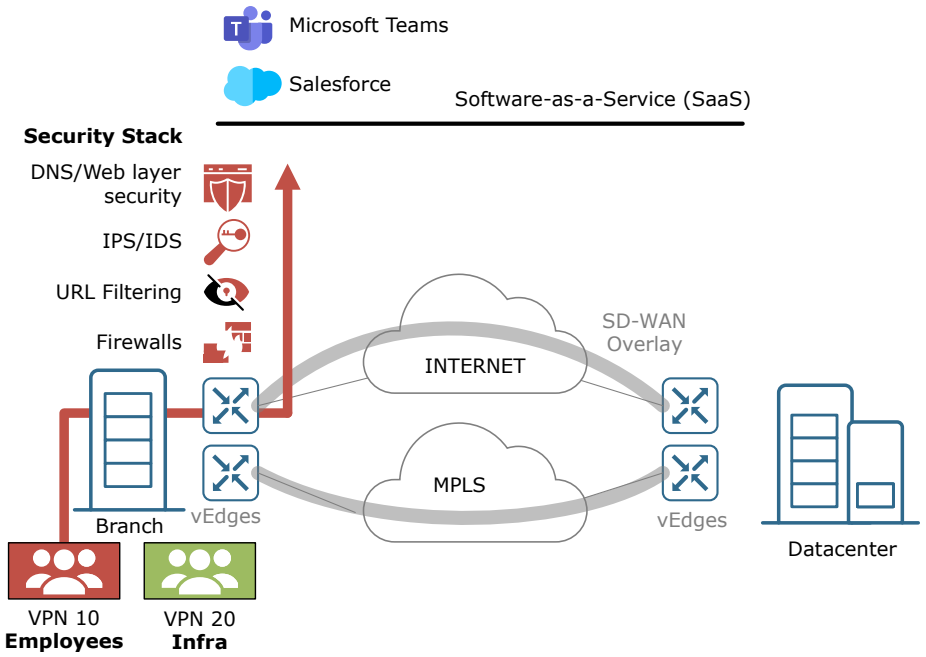


Figure 2. Cisco SD-WAN Direct Internet Access

Security

Cisco SD-WAN allows pushing the security stack directly on the WAN edge devices onsite. This reduces the need for security appliances at every branch, by providing inbuilt security features which include DNS security, Application-aware firewall, URL filtering, IPS/IDS, and Advanced Malware Protection (AMP).

In addition, instead of enabling the security stack at the WAN edge routers, the DIA feature allows for routing the traffic through a cloud security provider. In this case, the traffic from a particular remote site is routed to the cloud security provider through point-to-point IPsec tunnels. The cloud security provider then pushes the traffic through the predefined security policies and route it out to the Internet.

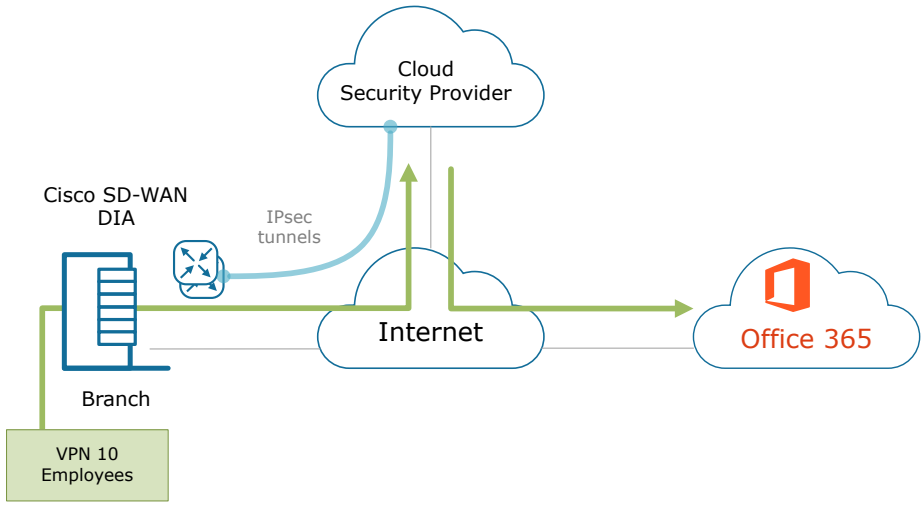


Figure 3. DIA traffic through a cloud service provider

Guest Access

Cisco SD-WAN provides an easy and secure way to create an isolated Guests segment that is isolated from the enterprise network and has its own security policies. Typical DIA traffic policies include:

- Restricting bandwidth usage of guests users
- Restricting access to certain Internet resources
- Restricting access to internal enterprise resources
- Protecting the network from malicious content

RESTful APIs

Before we start with this lesson, I'd like to quickly explain some terms and concepts upfront.

- **What is an API? (Application Programming Interface) -** API is just an interface that allows software to interact with another software;
- **What is SDK? (Software Development Kit) -** SDK is a set of tools used to develop software for a specific platform;
- **What is the difference between API and SDK? -** API is purposely built to allow specific communication between applications. SDKs allow for the creation of applications including APIs.

And the non-technical explanation - the SDK represents an entire office: all engineers, salespersons, furniture, and all office gear. An API represents just the Internet lines that allow communication in and out of the office.

Ok, now back to Cisco SD-WAN...

The Business Need

Legacy network devices were designed to be managed and operated by humans. They were not made to communicate natively to other software. That's why traditional networks are not extensible and automated in a native way. Nowadays, all other IT verticals are software-driven, so businesses are pushing the networks in that direction as well.

Cisco SD-WAN has been designed with automation and extensibility in mind. Cisco vManage provides northbound RESTful APIs that allow customers to build their own unique business logic on top of the SD-WAN solution. For example, enterprises can integrate their existing OSS (Operational Support System) and BSS (Billing Support System) tools and consume telemetry data, automate incident tickets creation and lifecycle, and automate the deployment of new services.

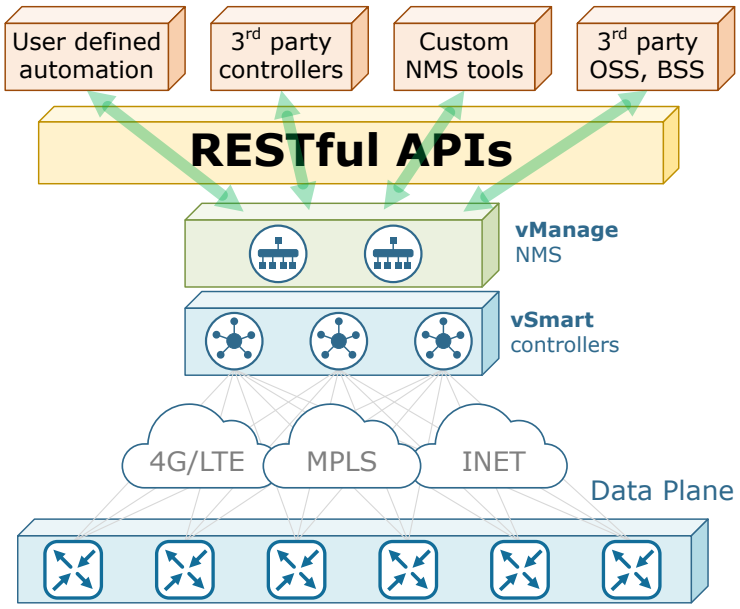


Figure 1. Cisco SD-WAN REST APIs

The northbound APIs open a new world of possibilities to network engineers as well. Many trivial operational tasks that consume lots of time and effort in a large-scale environment can be easily automated. For example, configuration audits, network/security audits, inventory reports, automated backup/restore, 3rd-party tools integration, and so on.

Let's take a look at the most typical use-cases.

User-defined Automation

vManage REST APIs provide endless possibilities for the automation of user-defined tasks. You can basically write software that can interact with the Cisco SD-WAN solution without human supervision. For the most simple cases, this software would be a python script. However, for more advanced scenarios, sophisticated Ansible playbooks can be leveraged.

3rd-Party Controllers

One of the beauties of the vManage RESTful APIs is that they can be leveraged by other domain controllers, such as Cisco DNA Center or Cisco ACI, in order to deliver a unified single management plane across multiple technology verticals. This allows for the emergence of a single-pane-of-glass tool for service orchestration, configuration, administration, and troubleshooting. Such integration might finally allow for a full-scale intent-based network that can enforce business intend across the entire infrastructure from user to applications to cloud.

Technically, the REST APIs allow for the integration of non-Cisco SD-WAN controllers as well but at the current stage, interoperability between SD-WAN vendors is not present.

3rd-Party OSS, BSS, and SP tools

The vManage REST APIs allow Service Providers and MSPs to integrate their existing operational and billing systems (OSS/BSS). Most typical examples of such integrations are:

- Service and data usage statistics are collected using the RESTful APIs and then feed into the service provider's billing system;
- Using the RESTful API, service providers can design custom self-servicing portals that allow users to subscribe to new services or change the operational status of existing ones. For example, purchasing and deploying new SD-WAN security features, network segmentation and slicing, and so on;
- Automated deployment of new customers;
- CI/CD pipeline integrations, automated change scheduling, and rollbacks.
- SIEM (Security Information and Event Management) integrations, that can perform automated remediative actions against security alarms and incidents.

vManage's Embedded API Library

Cisco includes very sophisticated API documentation as part of the vManage software using the following URL:

https://[vManage-IP-address]:8443/apidocs

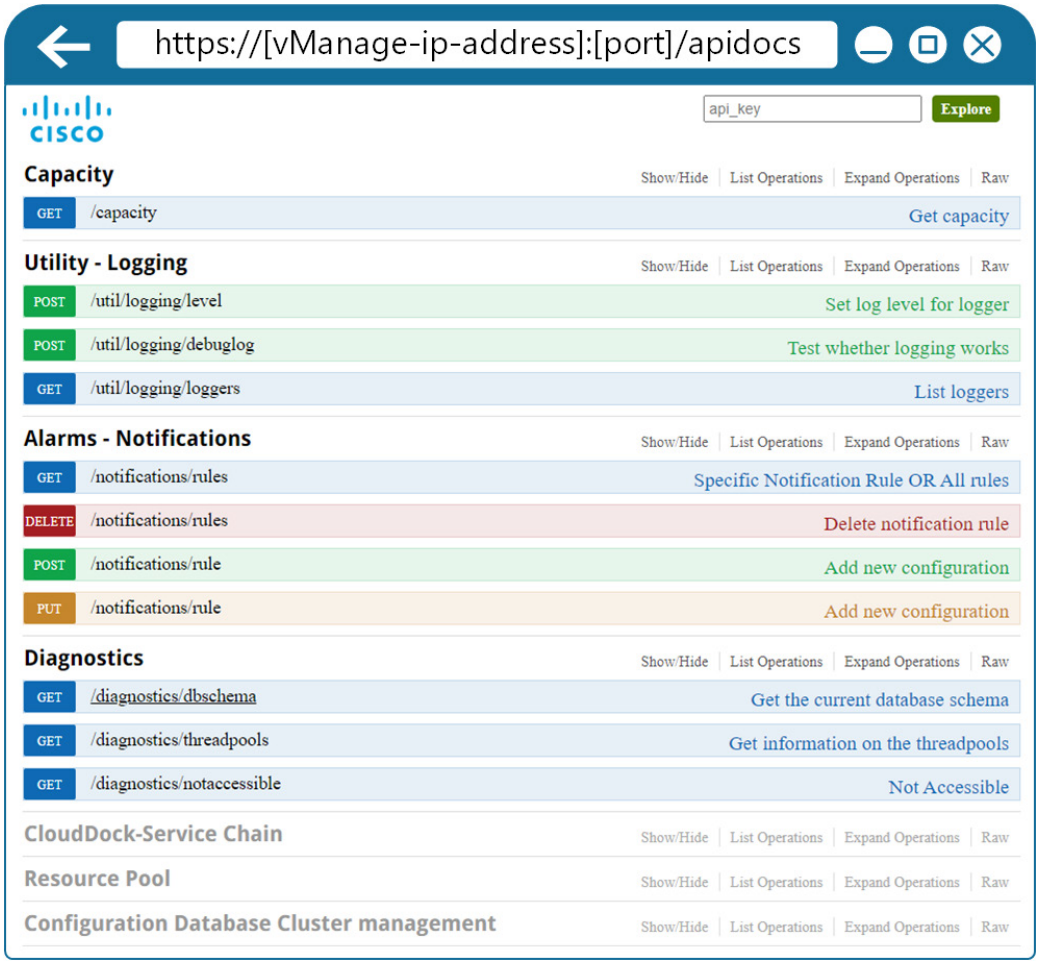


Figure 2. Cisco SD-WAN API Docs

The documentation is divided into a few major categories of API calls:

- Certificate Management
- Configuration
- Device Inventory
- Real-Time Monitoring
- Troubleshooting Tools

When you select a particular API class. The documentation shows its response class, required parameters, and the returned status codes.

Table 1 below shows some typical examples of vManage API calls:

Requested Info	API Call
Environment health status of device's	dataservice/device/hardware/environment?deviceId= system-ip-address hardware components
A list of all cisco sd-wan devices	dataservice/device
Status of the transport interfaces of a device	dataservice/device/interface?deviceId=system-ip-address&port-type=transport
Status of DTLS control connection	dataservice/device/control/connections?deviceId= system-ip-address
Interface statistics and packet drops	dataservice/device/interface?deviceId=system-ip-address
A list of OMP peers	dataservice/device/omp/peers?deviceId=system-ip-address
A list of BGP peers	dataservice/device/bgp/neighbors?deviceId=system-ip-address

Examples of vManage API calls

Cisco SD-WAN Python SDK

Cisco has provided a full-fledged Python-based SDK for Cisco vManage that has tools, libraries, and documentation to simplify the interactions with the REST API. It is intended for engineers interested in automating the administration and operation of the SD-WAN solution using Python without any GUI interaction.

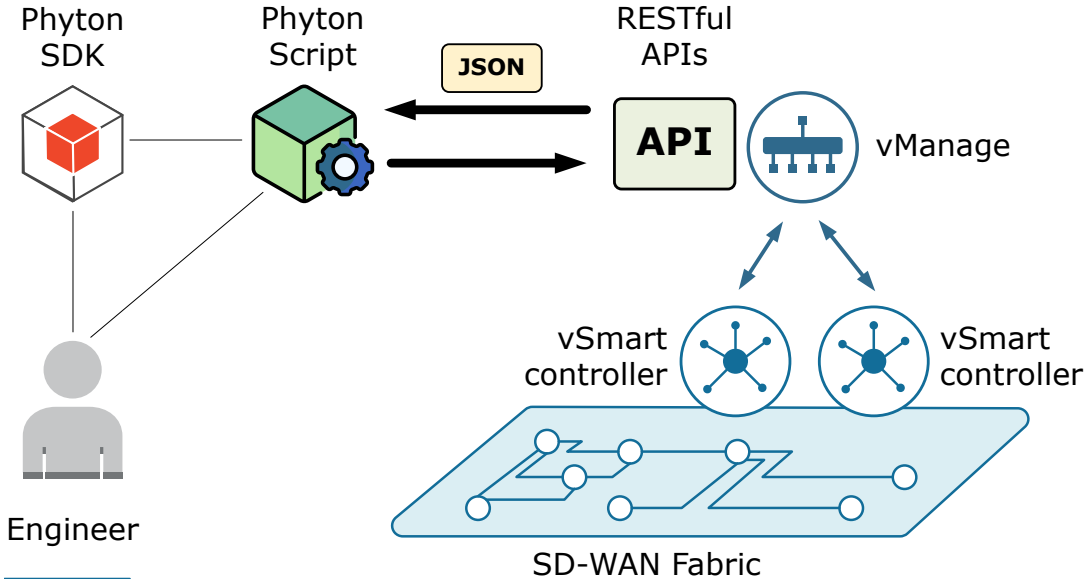


Figure 3. vManage Python SDK

There are many great things that you can do when using Cisco vManage programmatically. A few typical use-cases are:

- Software integrations with other platforms;
- Programatically keeping track of device status and acting upon change;
- Management of policies and device templates in an automated fashion;
- Automated backup and restore;
- CI/CD integrations;
- Automatic querying and aggregating of device and traffic statistics.

3. CISCO SD-WAN CONTROL PLANE

Underlay vs Overlay Routing

Why do we need the overlay?

Traditional network devices are hardware-centric and forward packets based on the destination IP address. Furthermore, each network node makes a separate independent decision on how, when, and to whom to forward each packet. This creates the following inefficiencies and drawbacks:

- Network segmentation and network slicing is not possible - duplicate IP address ranges cannot traverse a single IP network natively;
- Scaling is hard. Equal-cost multipathing (ECMP) over multiple types of WAN transports at scale is practically impossible.
- Design changes require hardware interactions
- Virtualization and Abstraction is not possible
- Multicast does not natively traverse public transport such as the Internet.
- And many more;

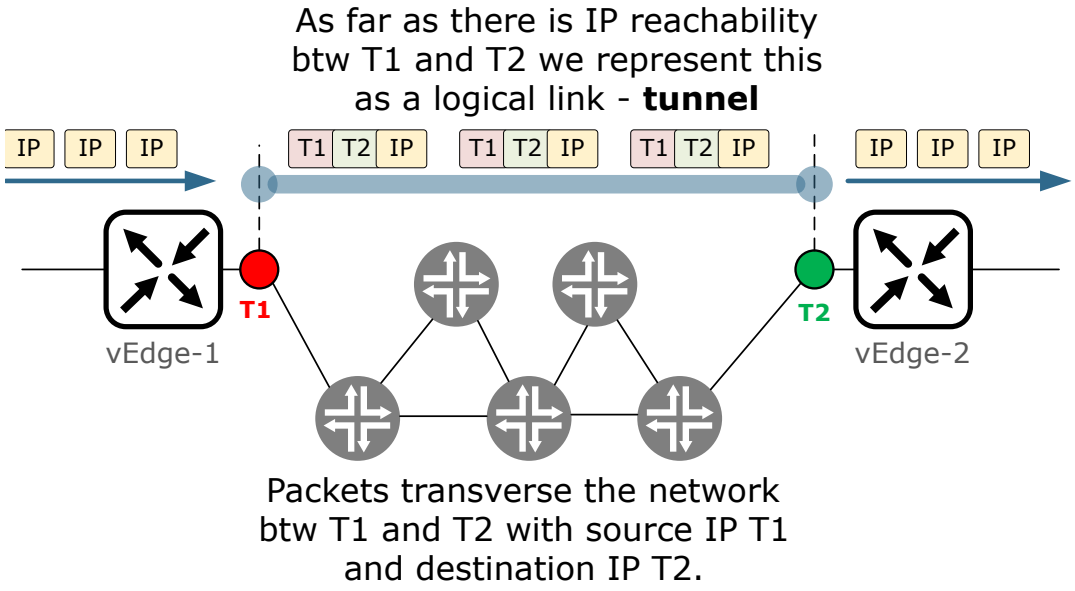


Figure 1. Overlay Tunnels

One of the main business propositions of Cisco SD-WAN is that it can use any given IP transport in an efficient, secure, and flexible manner. In order to do that, the solution abstracts the packet forwarding away from the network and application logic. This is done by building IPsec tunnels between the routers' WAN attachment points. The traffic that is going through the tunnels is encapsulated with a new IP header where the source/destination addresses are replaced with the T1/T2 addresses. This way, the intermediate network between T1 and T2 does not need to know anything about the original traffic. Most network engineers are very familiar with the tunneling techniques that exist. However, here are some examples of overlay tunnels - IPsec tunnels, Virtual Extensible LAN (VXLAN), Generic Encapsulation (NVGRE), Stateless Transport Tunning (SST), Network Virtualization Overlays 3 (NVO3), etc. Cisco's SD-WAN solution uses DTLS/TLS tunnels in the overlay.

Cisco SD-WAN Underlay vs Overlay

Cisco SD-WAN architecture is divided into two very distinct parts: the underlay network and the overlay fabric.

The underlay network represents the hardware infrastructure - all network devices that connect to the available WAN transports and local site networks. The router interfaces that connect the WAN transport networks are always configured in VPN0 (the Transport VPN). The attachment points that connect to the transports are called TLOCs (colored with red in figure 2). TLOCs play a very important role in abstracting the underlay network away from the overlay fabric and the applications. The main and only function of the underlay network is to provide IP reachability between TLOCs.

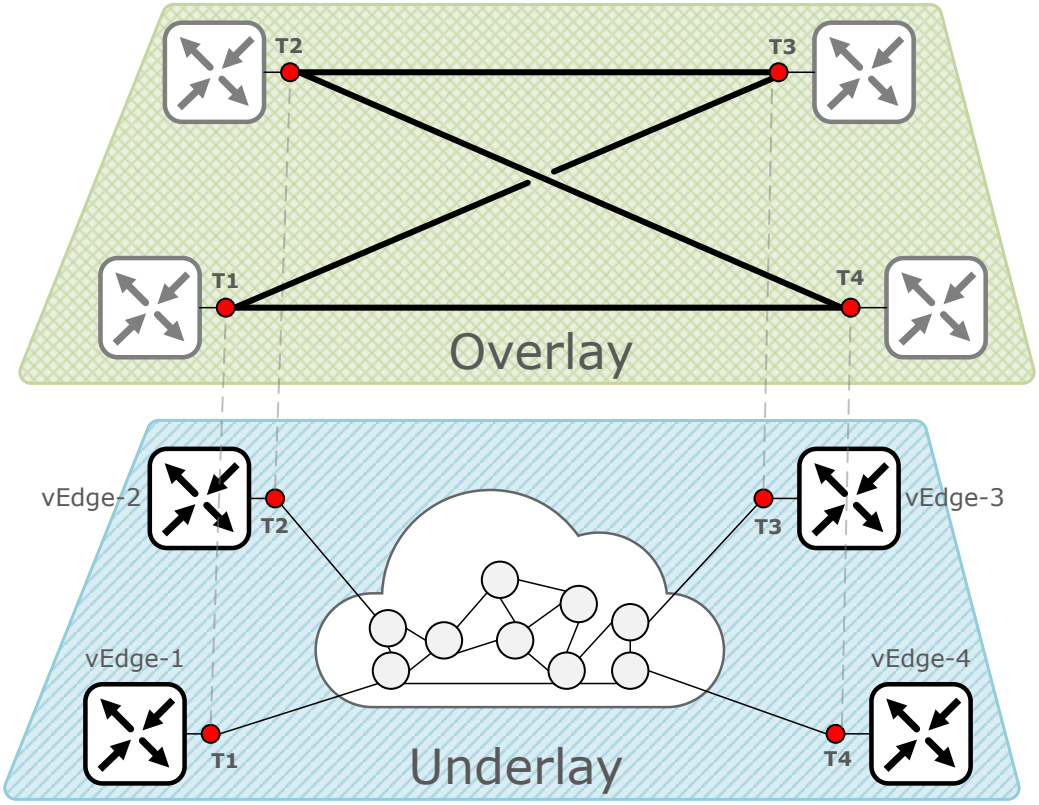


Figure 2. Underlay vs Overlay

Cisco's SD-WAN Overlay network is made of IPsec tunnels that traverse from site to site using the underlay network forming the so-called SD-WAN Fabric. Each overlay tunnel is formed between two TLOCs. The routing within the overlay is governed by the Overlay Management Protocol (OMP), a control-plane protocol very similar to BGP. The OMP protocol runs over secure DTLS or TLS connections between the WAN edge routers and the vSmart controllers. The process is very similar to the BGP operation, the vSmart controller acts as a BGP route reflector (RR), it receives, modifies, and re-advertises routes from the vEdge routers, but never participate in the data-plane (in the packet forwarding).

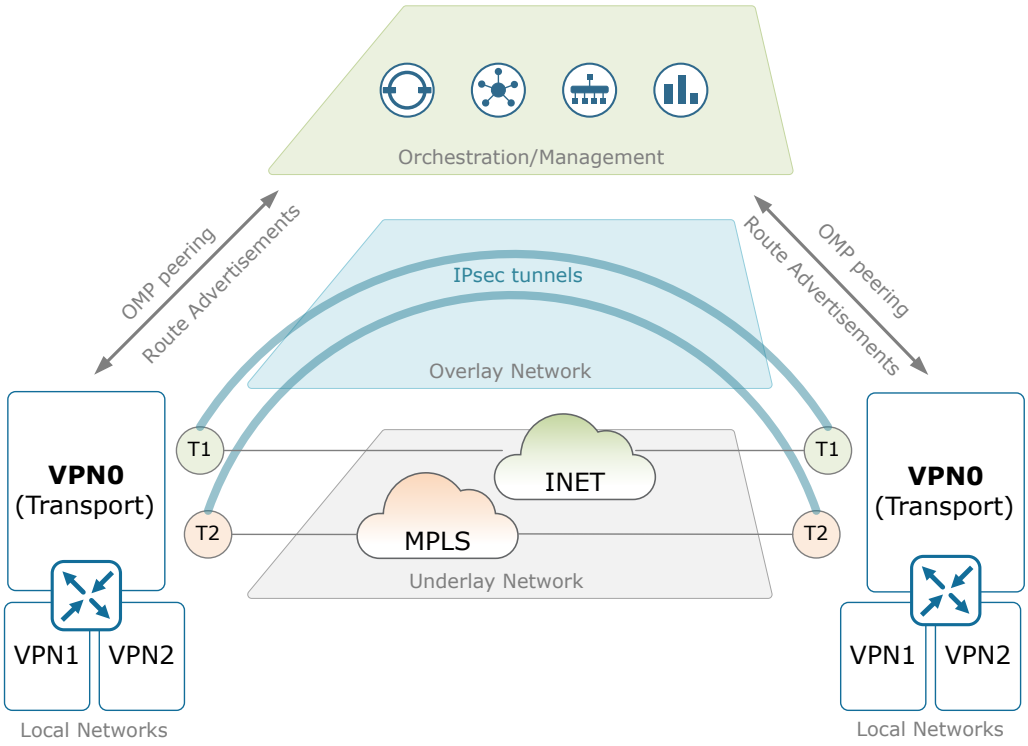


Figure 3. Cisco SD-WAN Underlay vs Overlay Routing

Network Segmentation

Abstracting the packet forwarding away from the network and application logic opens a world of possibilities. This allows for the use of VPNs that divide the overlay network into different network segments. Essentially, segmentation is done at the WAN edge routers, and the segmentation information is carried as a VPN label in the pakcets. However, the underlay network (Transport VPN0), that connects the WAN edge routers to the WAN transport, is completely unaware of the network segments (VPNs). Only the overlay knows about the VPNs; the underlay network follows the standard IP routing.

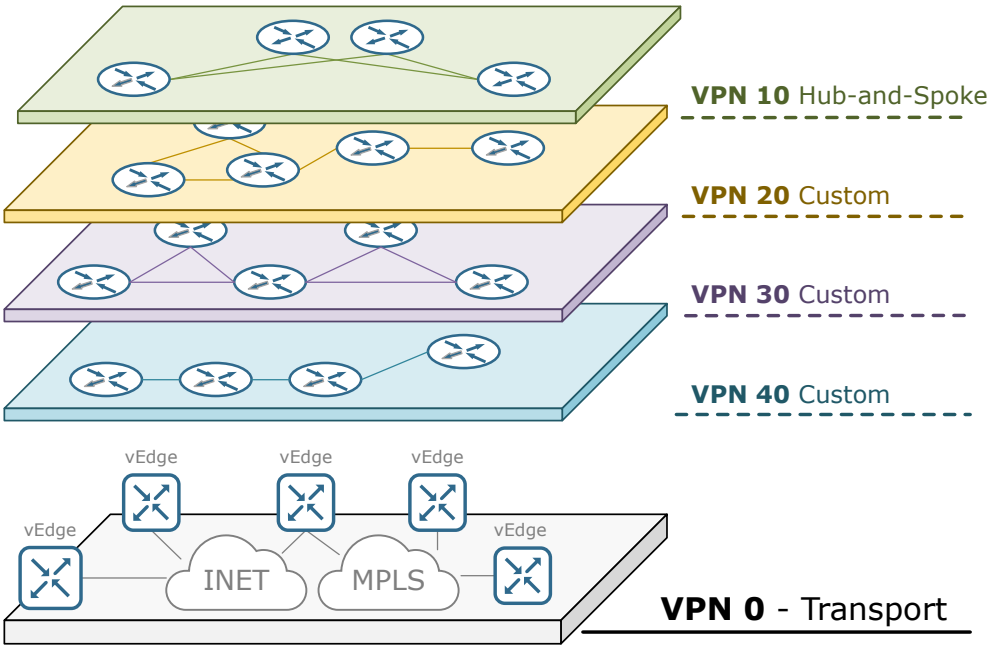


Figure 4. Different Overlay Topologies per VPN

Key Takeaways

Let's try to summarize the difference between Cisco SD-WAN's Underlay vs Overlay in one table:

Cisco SD-WAN	Underlay	Overlay
Description	The underlay network represents the hardware infrastructure - all network devices that connect to the available WAN transports and local site networks.	The overlay network the IPsec/GRE tunnels that are built between the underlay TLOCs.
Function	To provide IP reachability between TLOCs.	To provide network segmentation, security, and flexibility.
Packet Forwarding	Packets traverse over the WAN following the standard IP routing principles. Next-hop is an IP address.	Packets are forwarded between overlay nodes over IPsec tunnels. Next-hop is a TLOC of a remote peer.
Packet Control	Hardware oriented.	Software oriented.
Packet Delivery	Responsible for delivery of packets.	Abstracted away from the delivery of packets.
Control-Plane Protocol	Standard control-plane protocols such as OSPF, IS-IS, BGP, and static routing.	Cisco's Overlay Management Protocol (OMP)
Multipathing (ECMP)	Achieving Equal-cost Multi-pathing (ECMP) over multiple different types of WAN transports is associated with overhead and complexity. Very hard to achieve at scale.	Support for scalable multi-path forwarding over multiple virtual IPsec/GRE tunnels.
Deployment time	Deployment times are long. Design changes typically require hardware changes and manual activities.	Ability to rapidly deploy new functions at scale. Design changes in the overlay are done in a centralized fashion.
Multitenancy	Multitenancy could be achieved via VLANs/VRFs/NAT. Requires a custom and complex control plane to propagate the VRFs across the network. Large scale implementations are associated with configuration overhead and complexity.	Natively supports Multitanency and has the ability to manage overlapping IP addresses between multiple tenants.
Scalability	Less scalable due to legacy technology limitation.	Designed to provide great scalability, security, and flexibility.

OMP Overview

What is OMP?

Cisco SD-WAN Overlay Management Protocol (OMP) is the control plane protocol that runs between the WAN Edge devices and the Cisco vSmart controllers and also as a full mesh peering between the controllers themselves. Cisco SD-WAN applies the principles of Software-Defined Networking, which separates the control and data plane of the network. That means that control plane information is never exchanged between WAN edge routers. vEdges only send and receive information through the vSmart controllers as shown in figure 1.

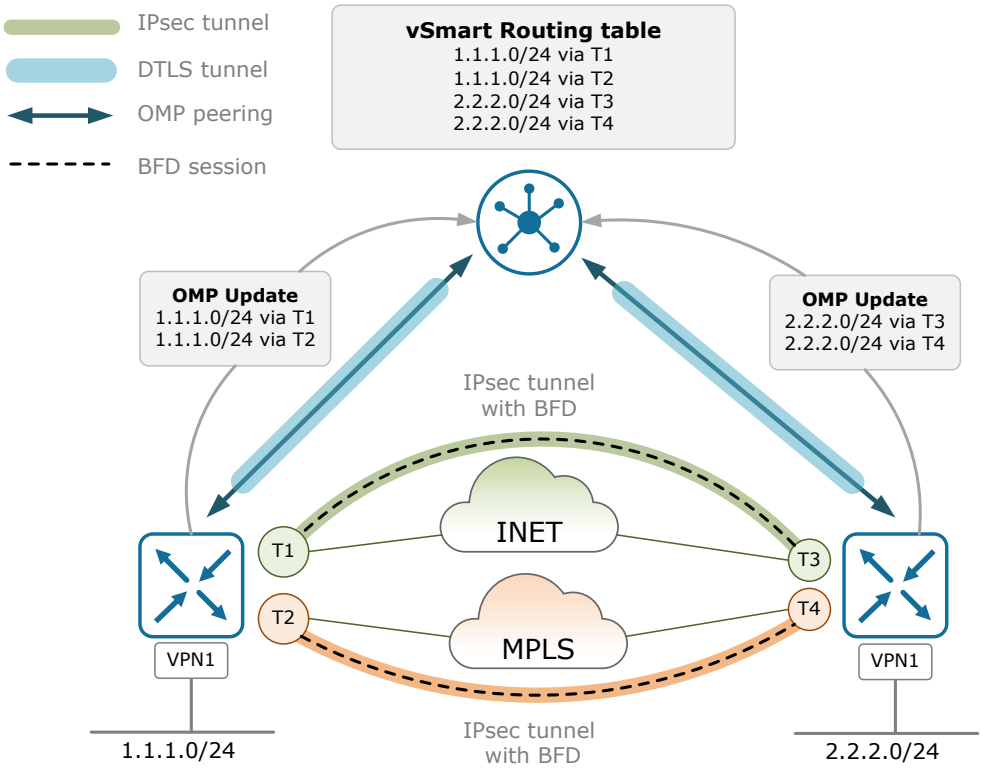


Figure 1. Cisco SD-WAN OMP Operation

The Cisco vSmart's role in the network is very similar to that of a BGP route reflector. The controller takes all routing and topology information received from every WAN edge device, calculates the best paths based on the configured policies, and then re-advertises the results to all other WAN Edge routers.

OMP Peering

OMP is enabled by default on all Cisco SD-WAN edge devices. When vEdges go through the Zero-Touch Provisioning process, they learn about the addresses of all available vSmart controllers and automatically initiate secure connections to them. By default, these connections are authenticated and encrypted via the Datagram Transport Layer Security (DTLS) protocol. Depending on the number of available transports, each vEdge router will try to establish a secure control connection via every TLOC. However, as shown in figure 2, the OMP peering uses the System-IPs, and only one peering session is established between one WAN Edge device and one vSmart controller even if there are multiple DTLS connections to the same controller.

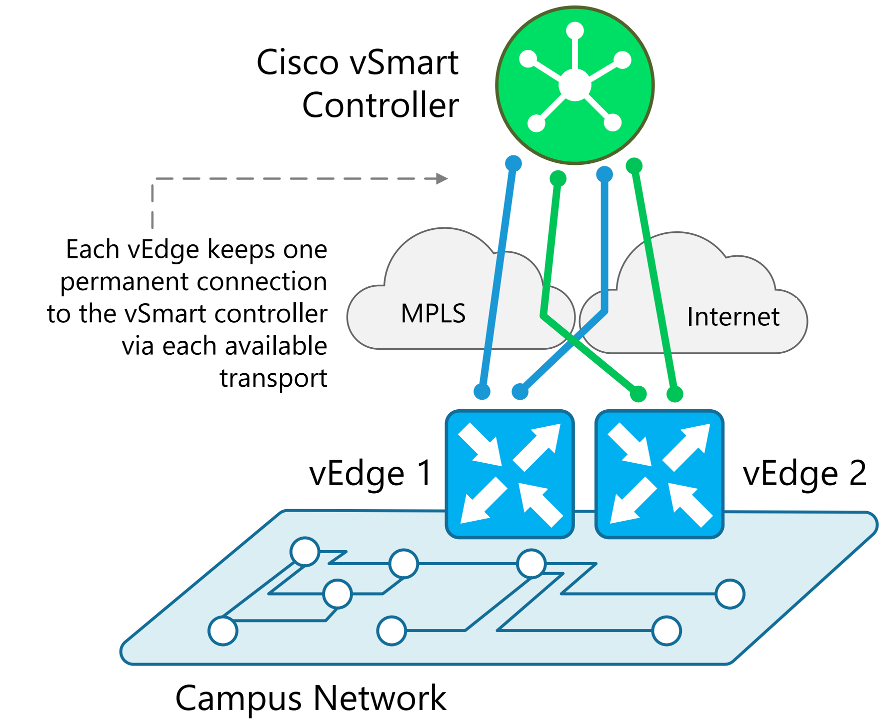


Figure 2. Cisco SD-WAN OMP Peering

You can see in the following output that the WAN edge device has two DTLS control connections initiated to the vSmart controller with IP address 1.1.0.3. One connection through the MPLS transport and another one through the Internet.

vEdge-1# **show control connections**

PEER TYPE	PEER PROT	PEER SYSTEM IP	SITE ID	PEER PUBLIC IP	PEER PUB PORT	LOCAL COLOR	STATE
vsmart	dtls	1.1.0.3	1	1.1.1.70	12346	mpls	up
vsmart	dtls	1.1.0.3	1	1.1.1.70	12346	public-internet	up
vbond	dtls -		0	1.1.1.60	12346	mpls	up
vbond	dtls -		0	1.1.1.60	12346	public-internet	up
vmanage	dtls	1.1.0.1	1	1.1.1.50	12346	mpls	up

However, if we check how many omp peering sessions to the controller there are, you can see that there is only one.

vEdge-1# **show omp peers**
R -> routes received
I -> routes installed
S -> routes sent

PEER	TYPE	DOMAIN ID	OVERLAY ID	SITE ID	STATE	UPTIME	R/I/S
1.1.0.3	vsmart	1	1	1	up	0:14:25:18	4/1/8

Another important thing to know is that these DTLS control plane tunnels are used by other protocols as well. For example, besides OMP, NETCONF and SNMP will also be transported via these secure connections. By utilizing these encrypted DTLS tunnels, we no longer need to be concerned about the native security of protocols like SNMP, NTP, etc.

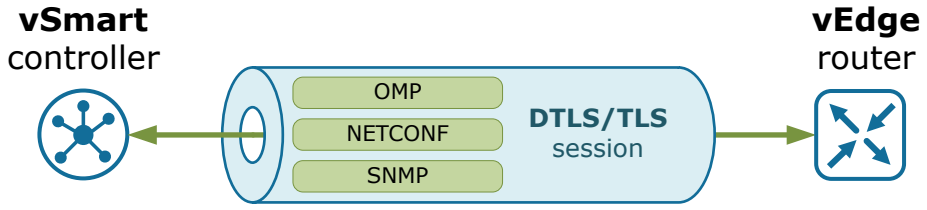


Figure 3. Cisco SD-WAN DTLS Connection

In a typical production deployment, there are at least two or three controllers for redundancy purposes. When we have multiple vSmarts, they also establish OMP peering between them in a full-mesh manner as shown in figure 4.

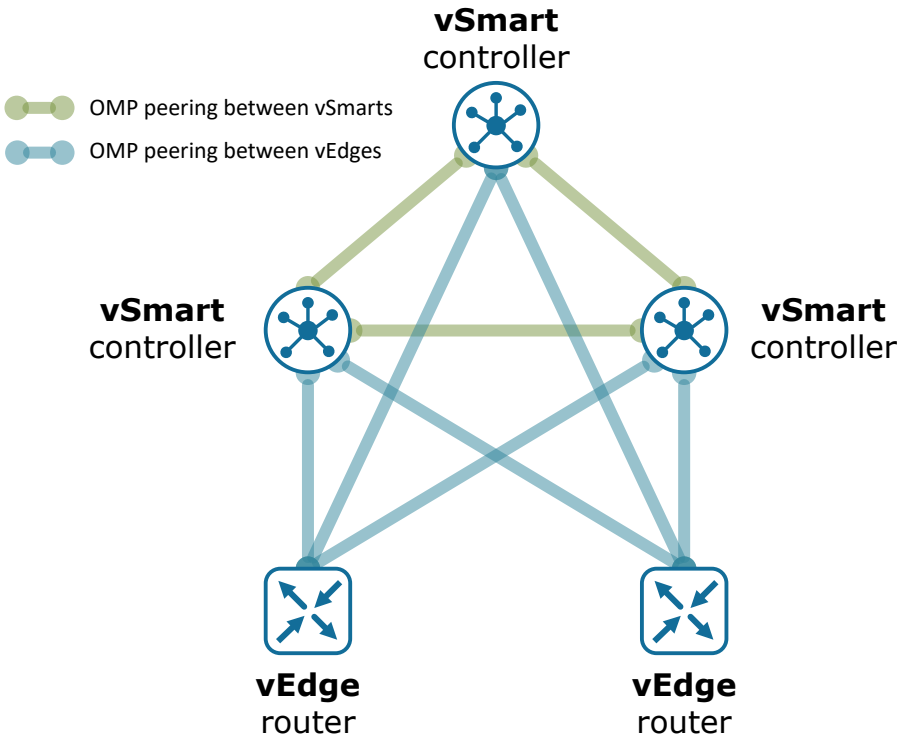


Figure 4. OMP Peering with multiple controllers

OMP Routes

As it is visualized in figure 5, vEdge routers advertise three types of routes via the Overlay Management Protocol (OMP) to the vSmart controllers:

- vRoutes (also called OMP routes or just routes) are prefixes learned from local networks of a WAN edge router. These can be locally connected prefixes or ones learned from a dynamic routing protocol such as OSPF and BGP. Once the vEdge device learns these prefixes, it redistributes them into OMP as vRoutes so they can be carried across the overlay fabric.
- TLOC routes advertise Transport Locators of the connected WAN transports, along with additional attributes such as public and private IP addresses, color, TLOC preference, site ID, weight, tags, and encryption keys.
- Service routes advertise embedded network services such as firewalls and IPS that are connected to the vEdge local-site network.

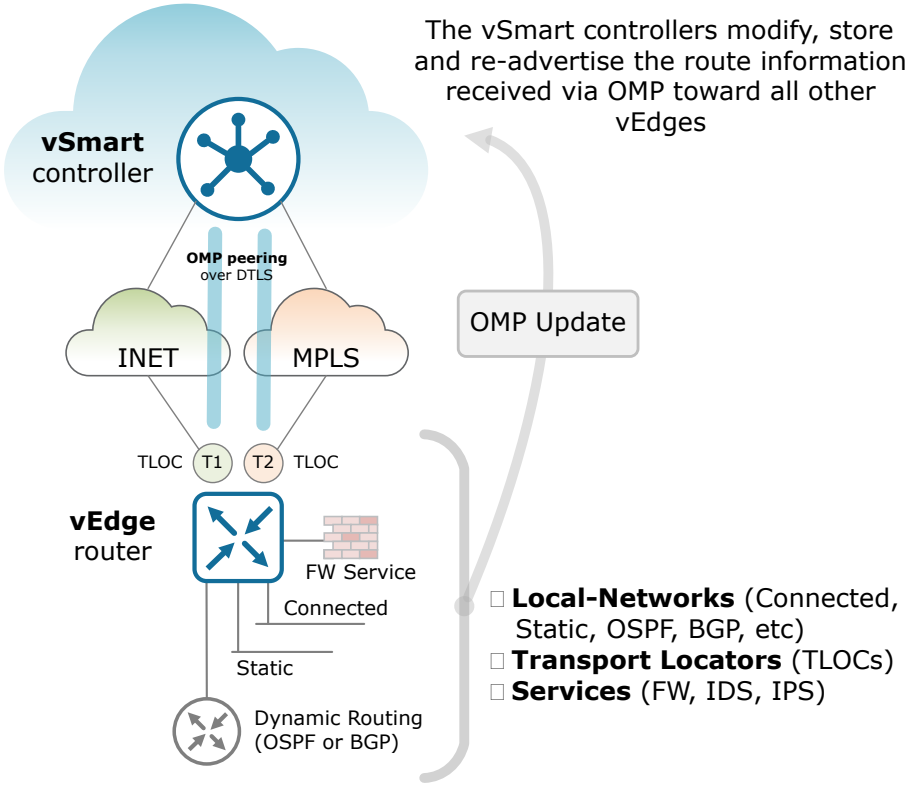


Figure 5. Cisco SD-WAN Overlay Management Protocol

vRoutes

The following output shows an example of a Cisco SD-WAN vRoute.

```
------------------------------------------------------------
omp route entries for vpn 1 route 172.16.50.0/24
------------------------------------------------------------
RECEIVED FROM:
peer              .1.0.3
path-id           2
label             1
status            C,I,R
loss-reason       not set
lost-to-peer      not set
lost-to-path-id   not set

Attributes:
originator        50.50.50.50
type              installed
tloc              50.50.50.50, mpls, ipsec
ultimate-tloc     not set
domain-id         not set
overlay-id        1
site-id           50
preference        not set
tag               not set
origin-proto      connected
origin-metric     0
as-path           not set
unknown-attr-len  not set
```

Let's look at every attribute in the OMP route in more detail. Each one can be used as a match criteria when creating policies and also can be modified in order to influence the routing decisions across the overlay fabric.

- **Originator:** The Originator attribute is pretty much self-explanatory. It identifies where the route was originally learned from.
- **TLOC:** The Transport Location (TLOC) is the next hop tunnel endpoint of the route. It is similar to the BGP_NEXT_HOP attribute. In order for a vRoute to be considered valid, it must have a next-hop TLOC that is known and reachable, where reachable means that there must be at least one IPsec tunnel to that TLOC with a BFD session in UP state.
- **Site ID:** The Site ID value represents the site of origin of the vRoute and is primarily used for loop-prevention. It is very similar to the BGP autonomous system number (ASN). The value can also be used for influencing routing decisions and policy orchestration. All sites must have unique Site-IDs and in sites where there are multiple WAN edge devices, they must have the same site ID for loop prevention.
- **Preference:** The Preference value is designed to be used for influencing the best-path selection process for a given prefix. It is pretty much the same as LOCAL_PREF in BGP - a higher preference is preferred over a lower one.
- **Tag:** This is an optional, transitive attribute that operates similarly to the route tags in traditional routing protocols. It can be matched and acted upon via policy.
- **Origin:** This value represents the originating protocol of the prefix. It can be BGP, OSPF, EIGRP, Connected, or Static. Origin is used in the OMP best-path selection for vRoutes and also can be influenced by a policy.
- **Origin-Metric:** This value represents the originating protocol-metric of the prefix.
- **VPN:** This value shows what VPN this route was advertised from. VPN tags allow for logical separation of networks and the use of overlapping subnets, assuming that they live in different VPNs.

TLOCs

Transport Locators (TLOCs) represent the attachment points where a vEdge router connects to the available WAN transports. A TLOC is uniquely identified by a tuple of three values - System-IP, color, and encapsulation type. However, TLOC routes have a number of other attributes that are advertised to the vSmart controllers, such as private and public IP addresses, port numbers, and encryption keys. Each WAN edge device advertises its Transport Locators via OMP TLOC routes to all Cisco vSmart controllers. The controllers then redistribute the TLOC routes to all other vEdge routers. When a vEdge receives a new TLOC route, it attempts to form an IPsec tunnel and establish a BFD session over each available WAN transport. However, WAN edge devices do not form overlay tunnels to devices with the same site-id.

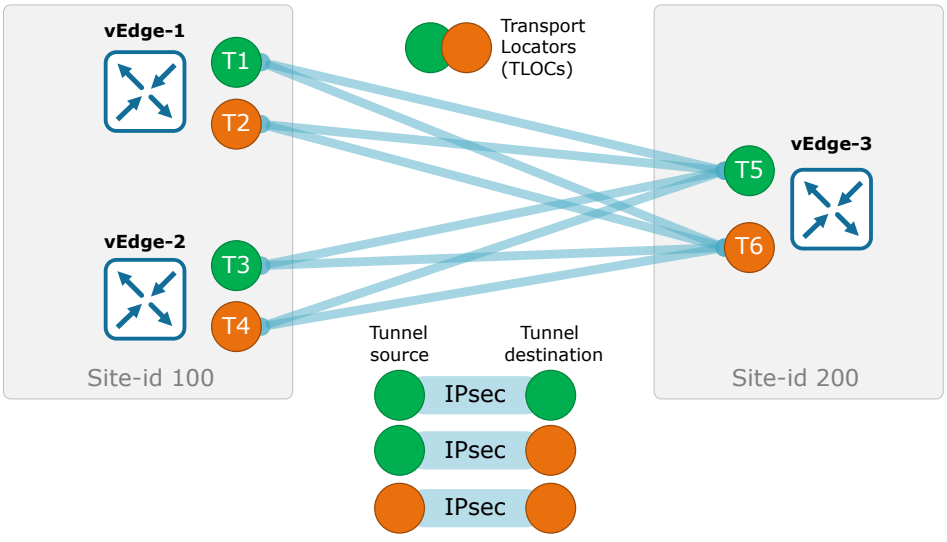


Figure 6. Cisco SD-WAN Tranport Locators (TLOCs)

A TLOC route contains the following pieces of information:

- **TLOC private address:** This is the private IP address of the WAN edge's interface attached to the WAN transport.
- **TLOC public address:** This value contains the publicly routable IP address of the WAN transport interface if it is connected behind NAT. WAN edge devices use STUN (RFC 5389) protocol that allows both ends to find out their public address, the type of NAT they are behind, and associated port numbers. This is a very important part of supporting data plane connectivity across a NAT boundary. If both the public and private addresses are equal, the vEdge is considered to not be behind a NAT.
- **Color:** Colors are logical abstractions used to identify WAN transports that are attached to WAN Edge devices. They are statically defined keywords and are globally significant and identify an individual transport as either public or private. Possible options are 3g, biz-internet, blue, bronze, custom1, custom2, custom3, default, gold, green, lte, metro-ethernet, mpls, private1, private2, private3, private4, private5, private6, public-internet, red, and silver.
- **Encapsulation type:** The encapsulation type could be IPsec or GRE. To successfully form a data plane tunnel to another TLOC, both sides must use the same encryption type.
- **Preference:** The TLOC preference value makes one TLOC more preferred over another when comparing the same OMP route. A higher value is better.
- **Site ID:** This value identifies the originator of the TLOC route. It is used for loop prevention and it controls how data plane tunnels are built. WAN edge devices do not establish tunnels to other WAN edge routers in the same site (having the same site-id)
- **Tag:** TLOC tags are used in the same way as OMP and route tags.
- **Weight:** This value is pretty much the same as Weight in BGP. It is locally significant and is used for path selection manipulation. Higher is better.

```
------------------------------------------------------
tloc entries for 50.50.50.50
                mpls
                ipsec
------------------------------------------------------
RECEIVED FROM:
peer            1.1.0.3
status          C,I,R
loss-reason     not set
lost-to-peer    not set
lost-to-path-id not set

Attributes:
Attributes:
attribute-type      installed
encap-key           not set
encap-proto         0
encap-spi           256
encap-auth          sha1-hmac,ah-sha1-hmac
encap-encrypt       aes256
public-ip           135.13.56.182
public-port         12346
private-ip          10.50.1.1
private-port        12346
```

```
public-ip            ::
public-port          0
private-ip           ::
private-port         0
bfd-status           up
domain-id            not set
site-id              50
overlay-id           not set
preference           0
tag                  not set
stale                not set
weight               1
version              2
gen-id               0x80000011
carrier              default
restrict             0
groups               [ 0 ]
border               not set
unknown-attr-len     not set
```

Service routes

As the name implies, Service routes represent services like firewalls and IPS that are connected to a WAN edge device or to the local-network attached to the WAN edge device. It is important to note that these devices that provide network services for the overlay fabric must be Layer 2 adjacent to the WAN edge device (meaning that there cannot be any intermediate hops between the WAN Edge router and the device performing the service).

OMP Best-Path Selection

OMP Route Advertisements

In the Cisco SD-WAN solution, WAN Edge routers advertise their local networks to the Cisco vSmart controllers using the Overlay Management Protocol (OMP). In a typical production deployment, most local networks are attached to two or more vEdge devices for redundancy and most networks get advertised from multiple devices. Additionally, each subnet is advertised as reachable via each Transport Locator (TLOC) that the WAN edge router has.

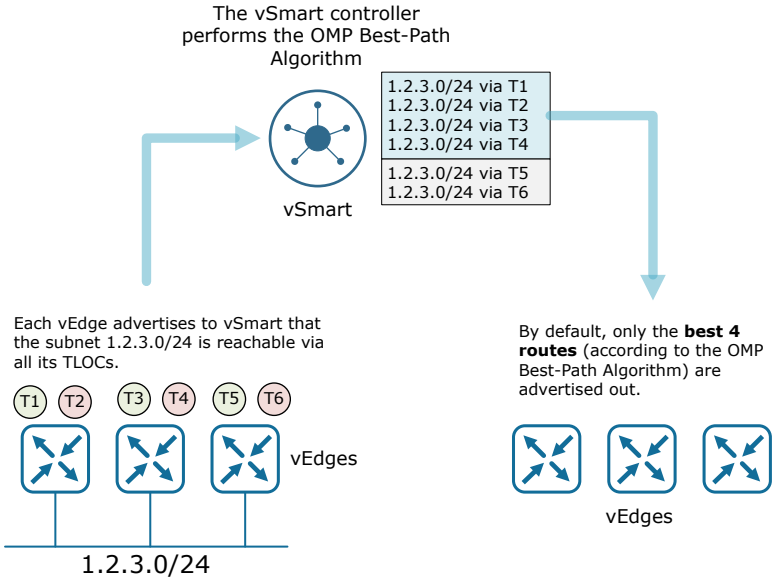


Figure 1. OMP Routing Advertisements

If we look at the example shown in figure 1, three vEdge routers are connected to subnet 1.2.3.0/24. The first one advertises to the vSmart controller that 1.2.3.0/24 is reachable via TLOC T1 and also that subnet 1.2.3.0/24 is reachable via TLOC T2. In the same manner, the other two WAN edge routers advertise that subnet 1.2.3.0/24 is reachable via TLOCs 3,4,5, and 6. In the end, the vSmart controller has six OMP routes for prefix 1.2.3.0/24. By default, vSmart is configured to advertise only four paths for a given prefix. Therefore, it must compare all available routes for this prefix and select the best four that will be sent out to all WAN edge routers. This is done using the Overlay Management Protocol Best-Path Algorithm.

Note that the number of routes per prefix that vSmart advertises can be changed using the following configuration:

```
send-path-limit (1-16)
```

It is also possible to configure the controller to send backup paths using the following configuration:

```
send-backup-paths
```

OMP Best-Path Algorithm

vSmart controllers and vEdge routers perform the Best-Path Selection when they have multiple routes for the same prefix. Figure 2 shows the Best-Path algorithm that Cisco SD-WAN devices go through.

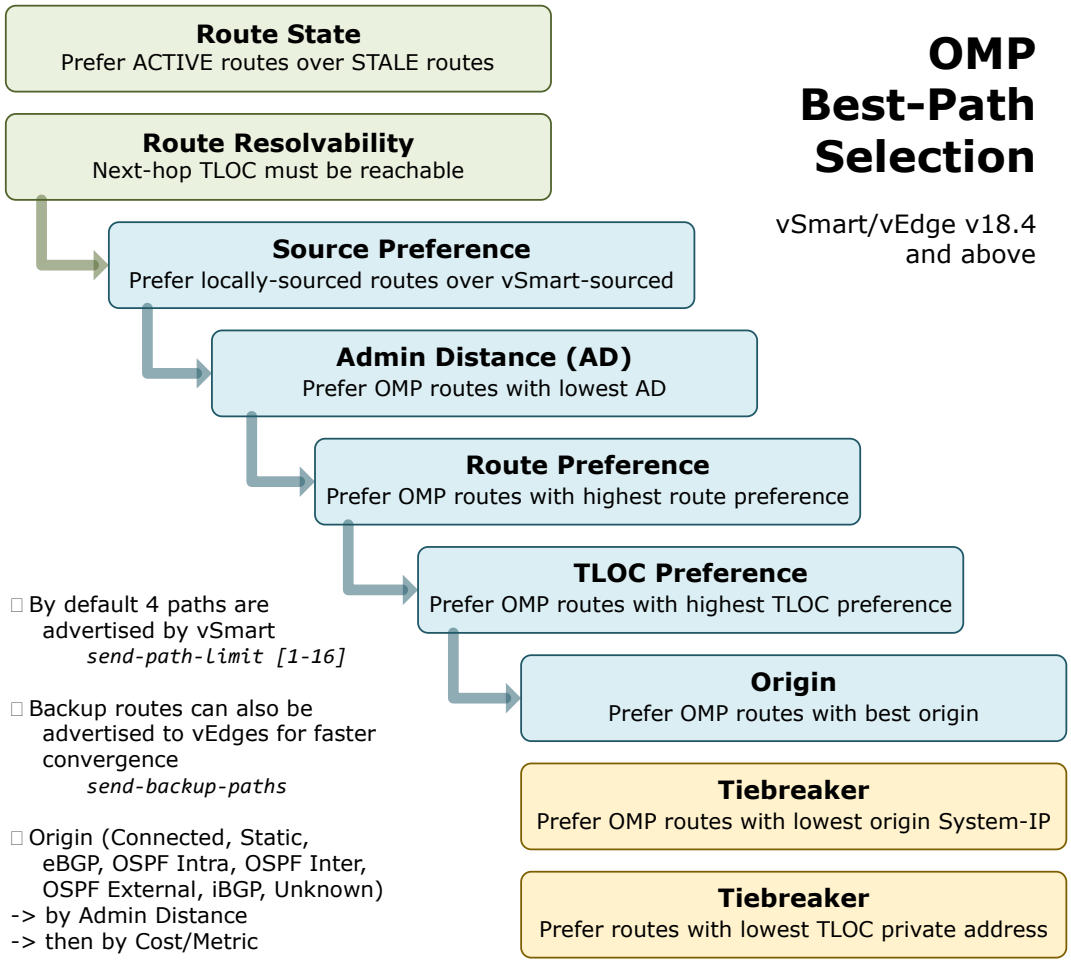


Figure 2. OMP Best-Path Selection Algorithm

Let's look at each step of the OMP Best-Path Selection in more detail:

1. Prefer ACTIVE routes over STALE routes. A route is ACTIVE when there is an OMP session in UP state with the peer that sent out the route. A route is STALE when the OMP session with the peer that sent out the route is in GRACEFUL RESTART mode;
2. Select routes that are Valid. Ignore invalid routes. A route must have a next-hop TLOC that is known and reachable.
3. Prefer vEdge-sourced routes over vSmart-sourced routes. This basically means that the WAN edge routers would always prefer routes that they originate over ones that come from the vSmart controller. From the perspective of the vSmart controller, it would always prefer routes that come from vEdges over ones that come from other controllers.
4. Prefer routes with lower administrative distance (AD);
5. Prefer routes with a higher route preference value. By default, all omp routes have 0 preference. This is typically the most often used value when we need to do traffic engineering.
6. Prefer routes with a higher TLOC preference value.
7. Compare the origin type, and select the first match in the following order:
    1. Connected
    2. Static
    3. EBGP
    4. OSFP intra-area
    5. OSPF inter-area
    6. OSPF external
    7. EIGRP internal
    8. EIGRP external
    9. IBGP
    10. Unknown
8. Compare the origin metric - If the origin type of the routes is the same, select the routes that have the lower origin metric.
9. Tiebreaker - If the origin types are equal, select the routes that have the lowest router-id (System-IP).
10. Tiebreaker - If the router IDs are the same, prefer the routes with the lowest private TLOC IP address.

ECMP - To be considered equal, omp routes must be valid and equal-cost up to step 8 (green and blue steps in figure 2). When there are more equal-cost routes than the send-path-limit value, the controller sorts the best ones based on the tiebreakers in descending order and advertises as many as the send-path-limit. This is visualized in figure 1.

Note that Cisco vEdge routers install a route in their forwarding table (FIB) only if the TLOC to which it points is active. Active TLOCs are ones that have a BFD session in UP state associated with them. When for whatever reason a BFD session becomes down(inactive), the Cisco vSmart/vEdge devices remove all routes that point to that TLOC from their forwarding table.

Best-Path Selection Examples

Let's look at some basic but important examples:

- A vSmart controller receives a route to 1.2.3.0/24 from a Cisco WAN Edge router with an origin code of eBGP. The controller also receives the same route from another vSmart Controller, also with an origin code of eBGP. Assuming all other properties are equal, the best-path algorithm would choose the route that came from the Cisco vEdge device.
- A Cisco vSmart Controller learns the same route, 10.10.10.0/24, from two Cisco vEdge devices on the same site. If all other parameters are the same, both routes are chosen and advertised to other peers. By default, up to four equal-cost routes are selected and advertised.
- A Cisco vSmart controller receives eight OMP routes for prefix 172.16.1.0/24. The send-path-limit value is the default one - 4. Six of them are chosen as best based on the OMP best path algorithm. They have the flag C set as you can see on the output below. The reason the other two have not been chosen as best can be seen in the loss-reason, lost-to-peer, and lost-to-path-id columns.

So there are six equal-cost routes (flagged with C) to 172.16.1.0/24, but the send-path-limit value is set to 4. Therefore, the controller sorts them based on the lowest originator System-IP and if there is a tie, based on the lowest TLOC private IP address, and advertises out the best four.

OMP Graceful Restart

While studying how Cisco SD-WAN works, have you ever wondered what happens with the overlay fabric when the SD-WAN Control Plane becomes unavailable? Well, there is a feature called OMP Graceful Restart that allows the data plane to continue functioning and forwarding traffic even if the control plane suddenly goes down or becomes unavailable. WAN edge devices do this by using the last known routing information that they received from the vSmart controllers. At the same time, vEdges actively try to re-establish a control-plane connection to the vSmart controllers. When the controllers are back up and reachable, DTLS control connections are re-established, and the vEdge routers then receive updated and refreshed network information from the vSmart controllers.

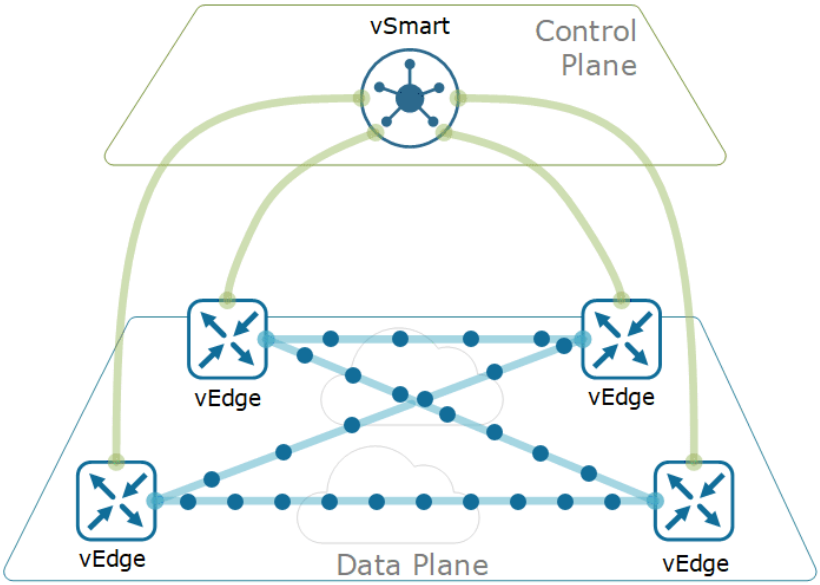


Figure 3. OMP Graceful Restart

Cisco vEdge and vSmart devices cache the OMP information that they learn from peers. The cached information includes OMP, TLOC, and SERVICE routes, IPsec SA parameters, and the centralized data policies in place. When a WAN edge device loses its OMP peering to the vSmart controller, the device continues forwarding data traffic using the cached OMP information. The Edge device also periodically checks whether the vSmart controller has come up. When it does come back up and the control plane peering is re-established, the device flashes its local cache and refresh the control plane information from the vSmart controller. This same technique is valid in the opposite scenario when a vSmart controller no longer detects the presence of Cisco vEdge devices. It then uses its local cache until the WAN edge device becomes reachable again.

Manipulating the Best-Path Selection Process

Similar to the well-known process of manipulating the BGP best-path selection process with route-maps, in Cisco SD-WAN we have the ability to manipulate the route selection locally on WAN edge devices. This is typically down using a Local device template. We have done that many times in traditional networking. The more interesting ability that SD-WAN allows us to do is to manipulate the routing information that goes in and out of the controller's routing table. This allows for a per-prefix network-wide routing manipulation, similarly to modifying route properties on a BGP Route-Reflector.

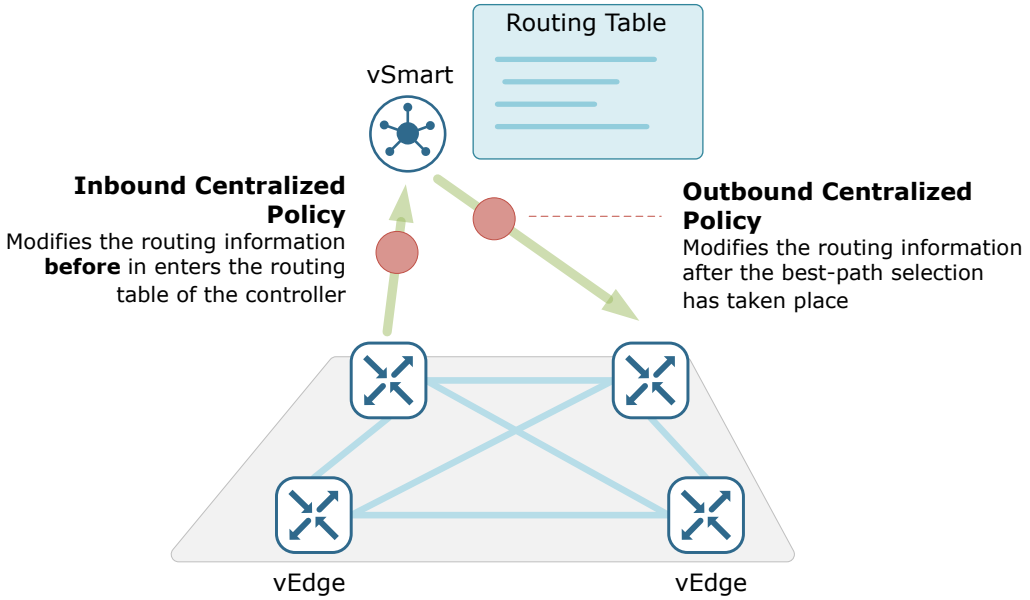


Figure 4. Manipulating the Cisco SD-WAN best-path selection process

There are two configuration options that allow for that:

Using Inbound Centralized Policy - With an inbound centralized policy, we can change the origin code or the TLOC preference of a particular prefix before it goes into the routing table of the vSmart controller. This will then influence the best-path selection and lead to a different output from the best-path algorithm. Remember that the best routes are then advertised downstream to all WAN edge routers. Therefore, any manipulation of the controller's routing table will change the control-plane information across the whole overlay fabric.

Using Outbound Centralized Policy - With an outbound centralized policy, we typically modify the routing information that is sent to a particular set of WAN edge devices in order to influence their own best-path selection algorithm.

4. CISCO SD-WAN DATA PLANE

What is a TLOC?

A TLOC is a Transport Locator that represents an attachment point where a Cisco WAN Edge device connects to a WAN transport. A TLOC is uniquely identified by a tuple of three values - (System-IP address, Color, Encapsulation).

- System-IP: The System-IP is the unique identifier of the WAN edge device across the SD-WAN fabric. It is similar to the Router-ID in traditional routing protocols such as BGP. It does not need to be routable or reachable across the fabric.
- Transport Color: The color is an abstraction used to identify different WAN transports such as MPLS, Internet, LTE, 5G, etc. In scenarios where transport types are duplicated (for example two different Internet providers) and should be treated differently from each other, the colors could be arbitrary, such as Green, Blue, Silver, Gold, etc.
- Encapsulation Type: This value specifies the type of encapsulation this TLOC uses - IP-sec or GRE. To successfully form a data plane tunnel to another TLOC, both sides must use the same encryption type.

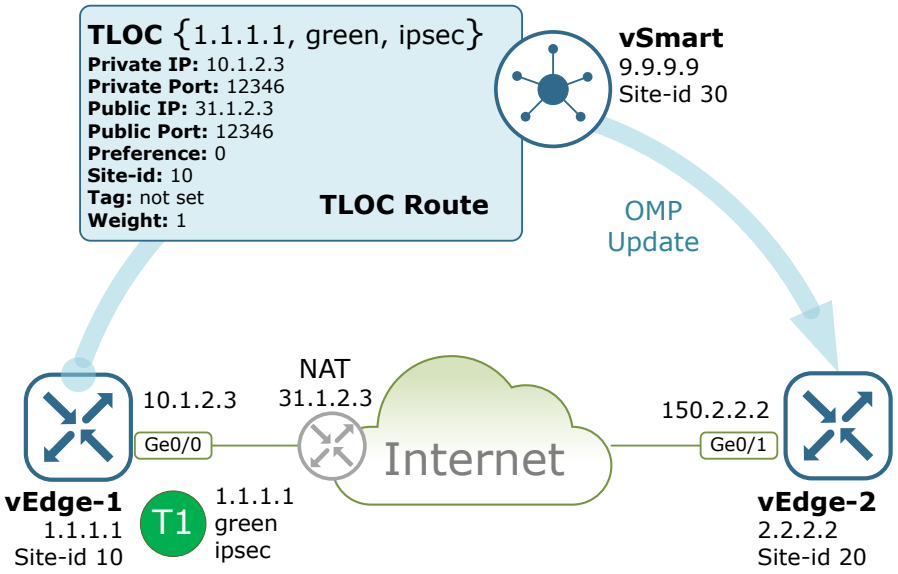


Figure 1. Cisco SD-WAN Forming an IPsec tunnel - Step 1

Let's stop here and look at the example shown in figure 1. WAN edge router 1 has an interface Ge0/0 with IP address 10.1.2.3 that connects to the Internet. As we all know, this IP is private and not routable over the Internet, so on the provider side, this private address is translated to the public address 31.1.2.3 through one-to-one NAT. On the other side, vEdge-2 has interface Ge0/1 with a publicly routable IP address 150.2.2.2.

Now let's say that we are tasked to manually configure an overlay tunnel between WAN edge 1 and WAN edge 2. If the devices are regular Cisco IOS routers and we are configuring a GRE tunnel with IPsec on top, we are going to specify a tunnel interface with tunnel-source IP and tunnel-destination IP on both ends taking the NAT into account. Also, we are going to specify the IPsec authentication and encryption parameters, the IPsec session keys, tunnel keys, and many additional settings depending on the situation. Now imagine that vEdge-1 and vEdge-2 must establish the same tunnel but in an automatic fashion and on-demand. Well, they will need to exchange all these parameters with each other. That is what the concept of Transport Locators (TLOCs) is created for.

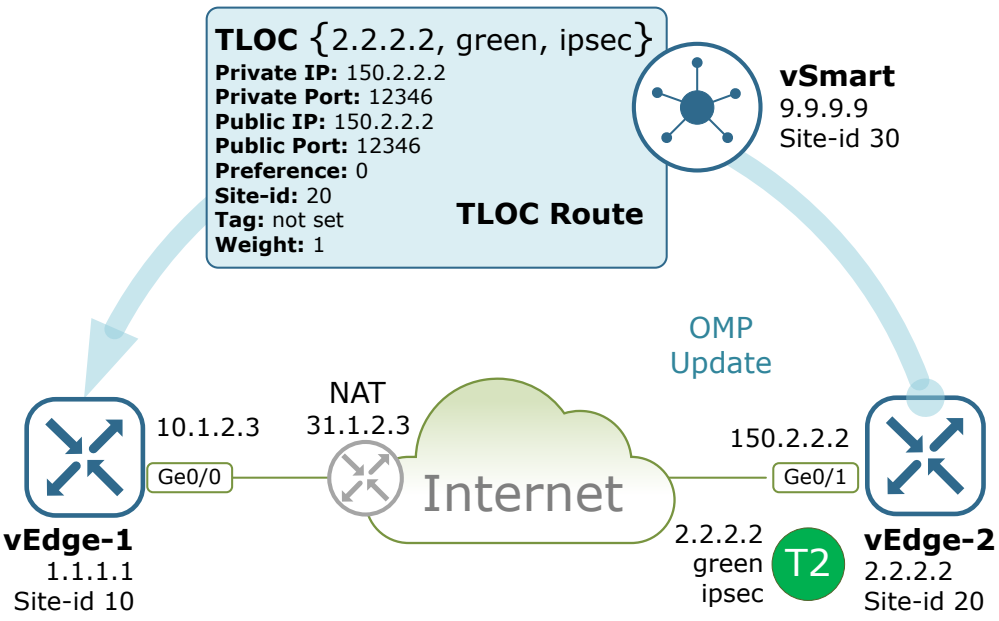


Figure 2. Cisco SD-WAN Forming an IPsec tunnel - Step 2

Having all we have said about TLOCs in mind, let's now follow the process of forming an overlay tunnel between WAN edge-1 and WAN edge-2. When vEdge-1 is first connected to the Internet via interface Ge0/0, it gets an IP address/Mask/Default Gateway via DHCP and it gets the transport color and encapsulation type from the controller or from a manual configuration. It then tries to establish all control plane connections to vBond/vManage/vSmart and in the process, it detects its publicly routable NATed address. Once it is able to communicate to the control plane via this interface, it creates a unique TLOC identified by the system-IP 1.1.1.1 (not the interface IP), the WAN transport color - green, and the encapsulation type - IPSec. This TLOC is then advertised to vSmart along with many other attributes.

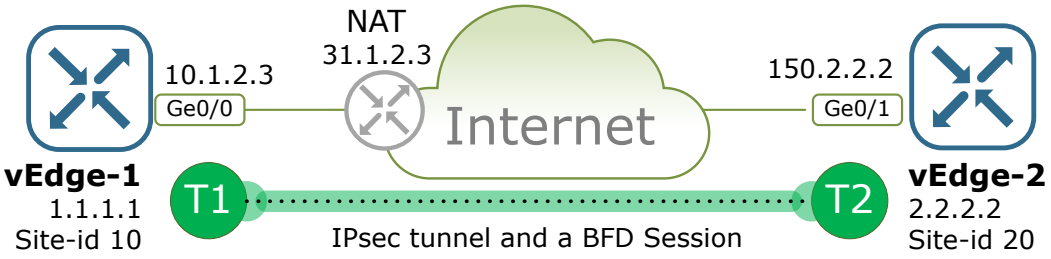


Figure 4. Cisco SD-WAN Forming an IPsec tunnel - Step 4

Multiple TLOCs

At this point, it is very likely that you might have thought - "Ah I see, a TLOC is just a tunnel endpoint and one TLOC is equal to one tunnel". If we have a WAN edge router that has two connections to two different WAN providers, for example, two Internet circuits, it would have two TLOCs. One that represents Internet-1 and one that represents Internet-2 as it is visualized on the example shown in figure 5. Well, this is the point where it gets a little bit more complicated.

By default, WAN Edge routers try to form an overlay tunnel to every TLOC over each available WAN transport, including TLOCs that belong to other colors if there is IP reachability between the two transport networks. In the example shown in figure 5, there is any-to-any reachability between the two clouds because they are basically two Internet providers. In this scenario, four IPsec tunnels will form T1-T3, T1-T4, T2-T3, T2-T4. There is one important exception, WAN edge devices do not form overlay tunnels to other devices in the same site ie having the same site-id.

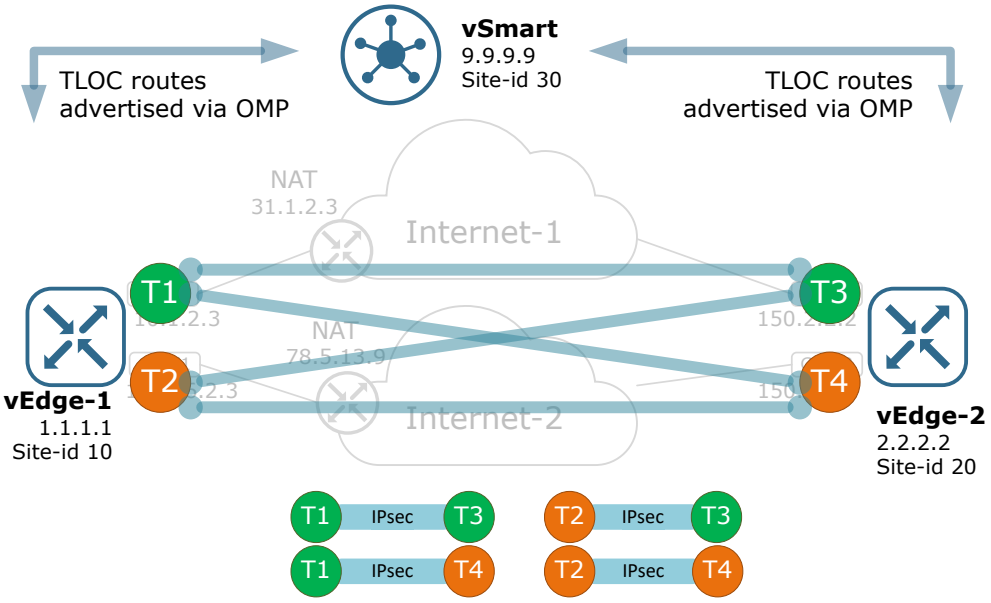


Figure 5. Forming the overlay with over TLOCs

This is where the Cisco SD-WAN overlay magic happens. If you advertise all TLOCs to all WAN edge routers, a full-mesh overlay fabric will be formed (assuming there is IP reachability between the underlay transports). However, if you want to have a custom overlay topology, you just control which particular TLOCs are advertised to which particular WAN edge routers. The logic is as follows - if vEdge-X receives TLOCs Y and Z, it will attempt to form IPsec tunnels from each own TLOC. If vEdge-X does not receive TLOCs Y and Z, it won't ever attempt to establish overlay tunnels to these TLOCs.

Overlay routing and TLOCs

Ok, but what does that mean for the overlay routing? Transport Locators are also a key part of the overlay OMP routing. Each omp route points to a TLOC next-hop and not to the particular overlay tunnel used to get there. By doing that, Cisco SD-WAN abstracts the overlay routing away from underlay routing, and WAN edge devices do not have to keep any underlay information in the overlay routing tables. For an OMP route to be considered as valid, the vEdge device must have at least one IPsec tunnel with a BFD session in UP state to the next-hop TLOC.

TLOC Color and Carrier

What is TLOC Color?

TLOC Color is a logical abstraction used to identify specific WAN transport that connects to a WAN Edge device. The color is a statically defined keyword that distinguishes a particular WAN transport as either public or private and is globally significant across the Cisco SD-WAN fabric. If you think about it, there is no automatic way for a vEdge router to understand which interface is connected to which transport cloud. If we look at the example in figure 1, vEdge-1 has three interfaces connected to three different providers. From the perspective of vEdge-1, the only way to distinguish which interface is connected to which cloud is through the concept of colors that would be externally defined by the controller or locally via CLI.

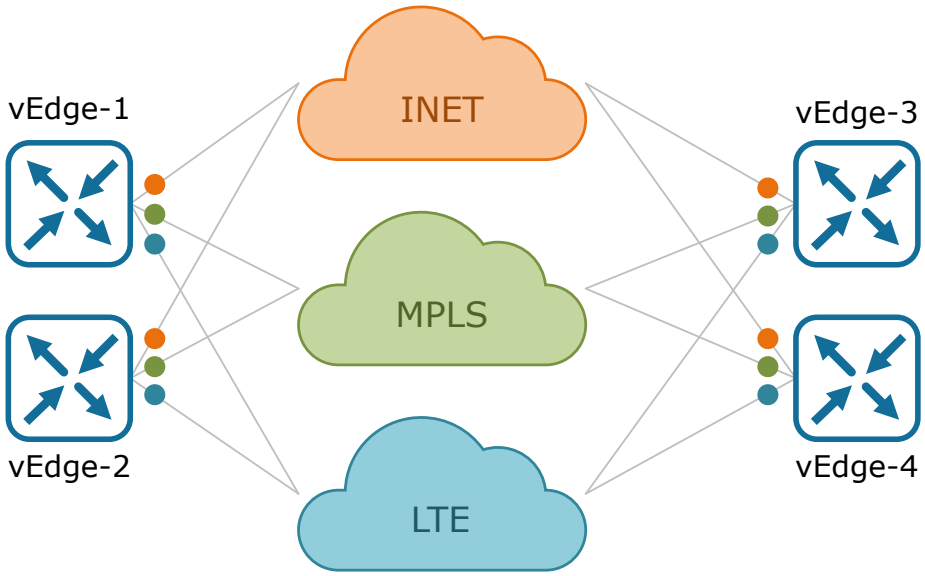


Figure 1. What is TLOC Color?

As of now, there are 22 pre-defined color keywords that are summarized in Table 1 below. They are divided into two main categories - public and private colors. The public colors are designed to distinguish connections to public networks such as the Internet where typically the attachment interface has an RFC1918 address that is later translated to a publicly routable address via NAT. On the other hand, private colors are intended for use on connections to clouds where NAT is not utilized. On WAN Edge routers, each Transport Locator is associated with a private-public IP address pair. The TLOC color dictates whether the private or public IP address will be used when attempting to form a data plane tunnel to a remote TLOC.

Public Colors	Private Colors
public-internet	mpls
biz-internet	metro-ethernet
3g	private1
lte	private2
blue	private3
green	private4
red	private5
bronze	private6
silver	
gold	
custom1	
custom2	
custom3	

Table 1. Cisco SD-WAN TLOC Colors

Communication Between Colors

During the authentication process with the vBond orchestrator, WAN edge devices learn whether they sit behind a NAT device and what is their NATed address and port. This is done using the STUN protocol and the process is explained in further detail in our lesson about TLOCs and NAT. In the end, each TLOC contains a pair of private/public addresses and ports. If there is no NAT, both the private and public addresses are the same, if there is a NAT device along the path, the private address represents the native interface IP and the public address represents the post-NAT address. When two Cisco SD-WAN devices attempt to form an overlay tunnel between each other, they look at the colors at both ends in order to decide which IP address to use.

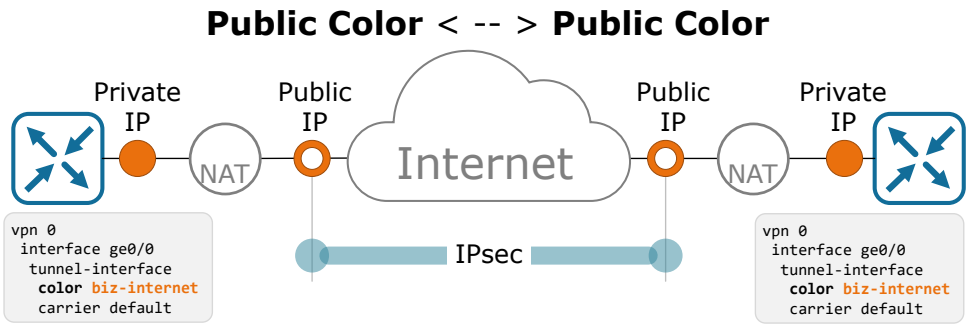


Figure 2. Overlay tunnel between public colors

Even if only one of the colors is public, the WAN edge devices will also attempt to form the data plane tunnel using the public IP addresses.

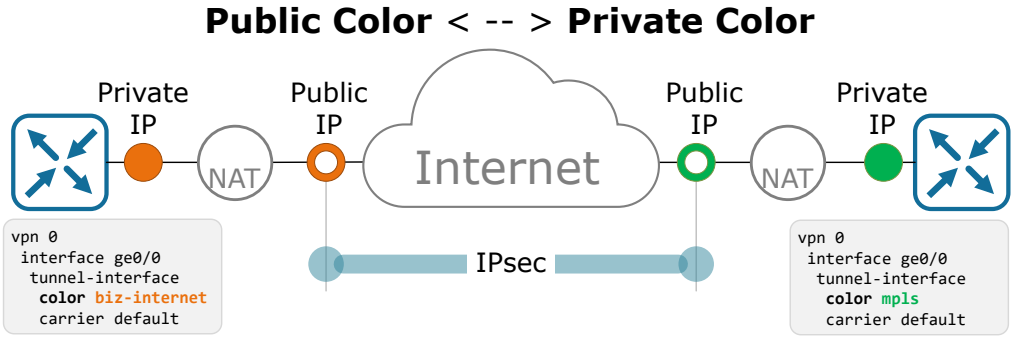


Figure 3. Overlay tunnel between public and private colors

However, If the TLOC color at both ends is a Private one, the WAN edge devices attempt to form the data plane tunnel using their private IP addresses.

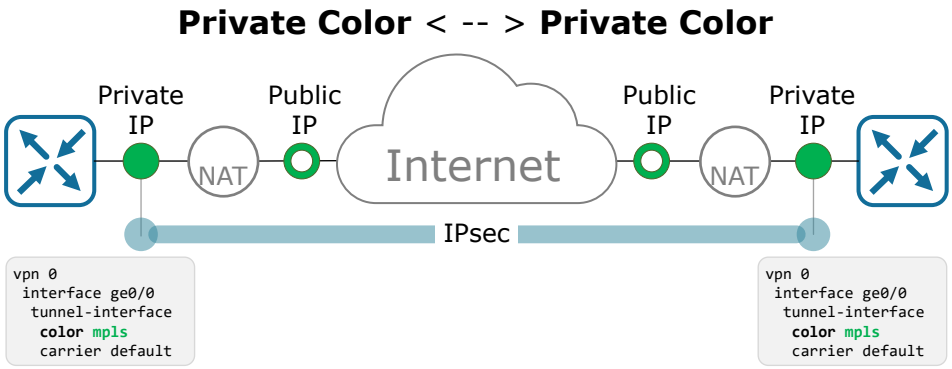


Figure 4. Overlay tunnel between private colors

This is particularly important in cases where WAN edge devices communicate directly with their native address over a private cloud such as MPLS but at the same time, they access the control plane through the same cloud via Network Address Translation. That is why data plane tunnels between TLOCs marked with private colors are formed using the private IP addresses as is demonstrated in figure 5.

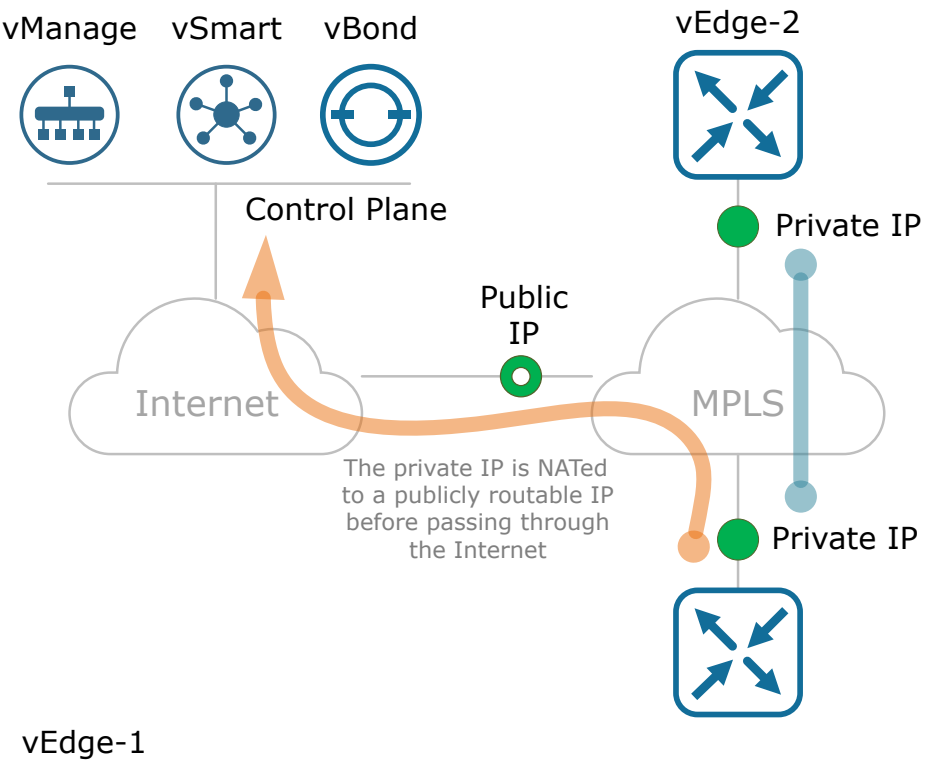


Figure 5. Why private clouds use private IPs?

TLOC Carrier

However, specific scenarios might occur where using the public IP addresses between private colors is the desired behavior. An example would be having two MPLS clouds that are interconnected using NAT. For such cases, there is a particular TLOC attribute called carrier that changes this behavior - if the carrier setting is the same in the local and remote TLOCs, the WAN edge device attempts to form a tunnel using the private IP address, and if the carrier setting is different, then the WAN edge device attempts to form a tunnel using the public IP address. The diagram below visualizes this:

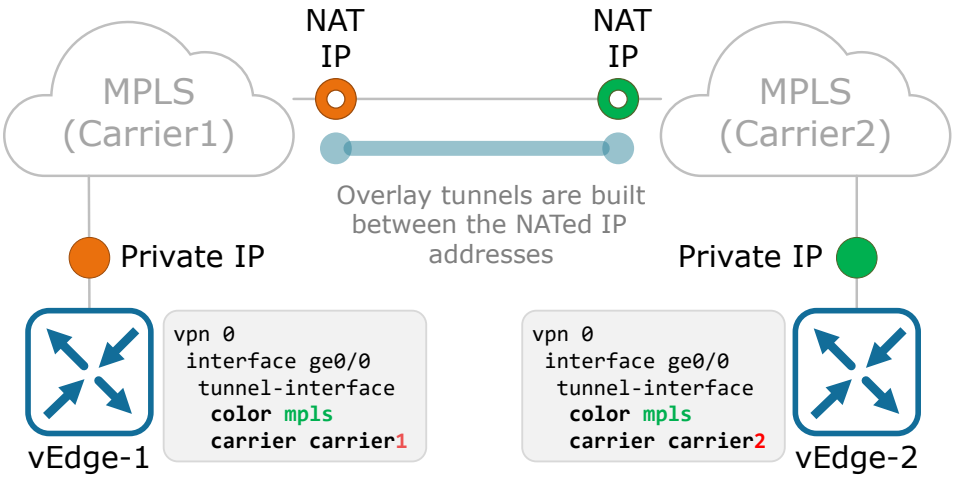


Figure 6. Overlay tunnels when using the Carrier settings

TLOC Color Restrict

By default, WAN edge routers try to form overlay tunnels to every received TLOC from a different site using every available color. This is usually the desired outcome in scenarios where we have two Internet connections from two different providers. Although we typically mark them with different colors in order to treat them separately as shown in figure 7, we would like to have a full mesh of tunnels because there is IP reachability between the clouds.

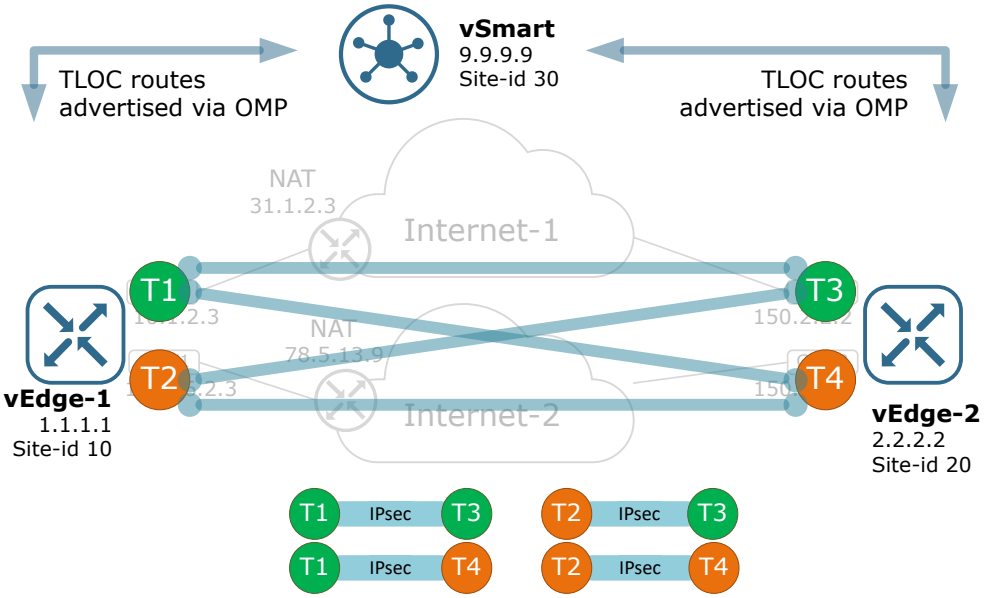


Figure 7. Default overlay fabric without Restrict keyword

However, this behavior might not be desirable in scenarios where we have one private transport alongside an Internet cloud, as it could lead to inefficient routing—such as WAN edge routers trying to build tunnels through the MPLS cloud to Internet TLOCs. Even though the IP reachability between the clouds may exist, the tunnels might be established over paths that are inefficient or unintended. This behavior can be changed with the restrict keyword or by using tunnel groups.

When a TLOC is marked as restricted, a WAN edge route router will attempt to establish a data plane tunnel to a remote TLOC only via WAN connections marked with the same color. This behavior is demonstrated in figure 8. vEdge-1 will never try to establish an IPsec tunnel from T1 to T4 because TLOC1 and TLOC4 are not marked with the same color.

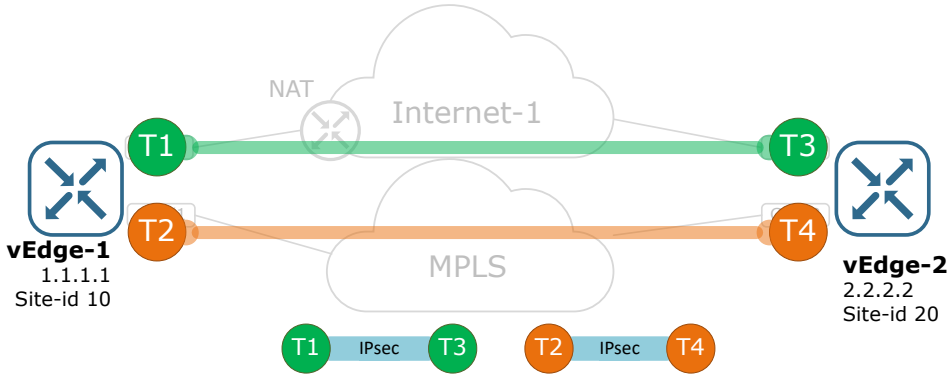


Figure 8. Overlay fabric using Restrict keyword

Another option to achieve the same goal of restricting the data plane connectivity between the same colors is by using tunnel groups. Only tunnels with matching tunnel groups will form a data plane connection (regardless of the color).

Tunnel Groups

TLOC Color vs Tunnel Group

By default in Cisco SD-WAN, vEdges will try to build a full-mesh overlay by establishing tunnels to all other TLOCs, regardless of their color. This behavior was explained in detail in our lesson for TLOC colors. For scenarios, where a full-mesh is not the desired overlay topology, there is an option called restrict, that allows only tunnels to TLOCs marked with the same color. Typically, this feature is configured on transports marked with private colors because a private cloud usually does not have reachability to public ones such as the Internet. However, the TLOC color-restrict option is not flexible enough because of the following limitation - only one interface can be marked with a particular color per WAN edge router.

IMPORTANT TOPIC A WAN edge router can't have multiple interfaces marked with the same color, because it breaks the uniqueness of the TLOC route! A TLOC is uniquely represented by a three-tuple (System-IP, Color, Encap). The system IP makes the route unique to a particular WAN edge device that has this system-IP address and the color makes the route unique to a particular interface on this exact WAN edge router.

If we look at the example shown in figure 1, vEdge-2 has one connection to the MPLS cloud that is marked with the mpls color. Therefore, different private color has to be assigned to the second interface (metro-ethernet).

Use Case 1: Grouping different interfaces to the same transport

If we want to have two overlay tunnels to vEdge-2 over the same MPLS transport, we can not use the restrict option on the mpls color. But then, if the private colors do not have the restrict option configured, they will try to establish tunnels to all other public colors that exist.

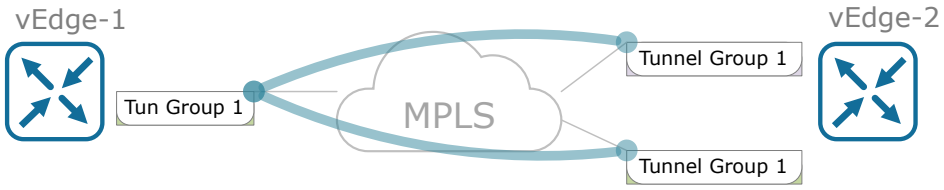


Figure 1. Scaling traffic to the same transport (no restrict option used)

The tunnel-group feature is designed to give more flexibility and granular control over the overlay tunnel establishments irrespective of the TLOC color. It works by assigning a tunnel group ID under a tunnel. Once the group-ID is configured under the TLOC, it obeys the following rules:

- TLOCs can only establish tunnels with remote TLOCs with the same tunnel-group IDs irrespective of the TLOC color.
- TLOCs with any tunnel-group ID will also form tunnels with TLOCs that have no tunnel-group IDs assigned.
- If the restrict-option is configured in conjunction with the tunnel-group option, then TLOCs will only form an overlay tunnel to remote TLOCs having the same tunnel-group ID and TLOC color

So if we go back to the example shown in figure 1, all interfaces attached to the MPLS cloud can be configured with the same tunnel-group 1 without the restrict feature. In this way, vEdge-1 will form an overlay tunnel to both interfaces of vEdge-2 but at the same time, tunnels to other public colors/tunnel-groups will not be attempted.

Use Case 2: Grouping different colors

Another typical use case that is illustrated in figure 2 is when a remote site (vEdge-3) uses different colors compared to the rest of the sites. In a typical real-world deployment, you would like to configure the private colors to only form tunnels over the private MPLS cloud and the same for the public colors of the Internet. This exact setup is only possible using tunnel-groups. By assigning all private colors to one tunnel-group (for example 2) and assigning all public colors with different tunnel group (for example 1), we will prevent the forming of overlay tunnel between the public and private transports while still allowing different private colors to form tunnels (which would not be possible if we use the restrict-option).

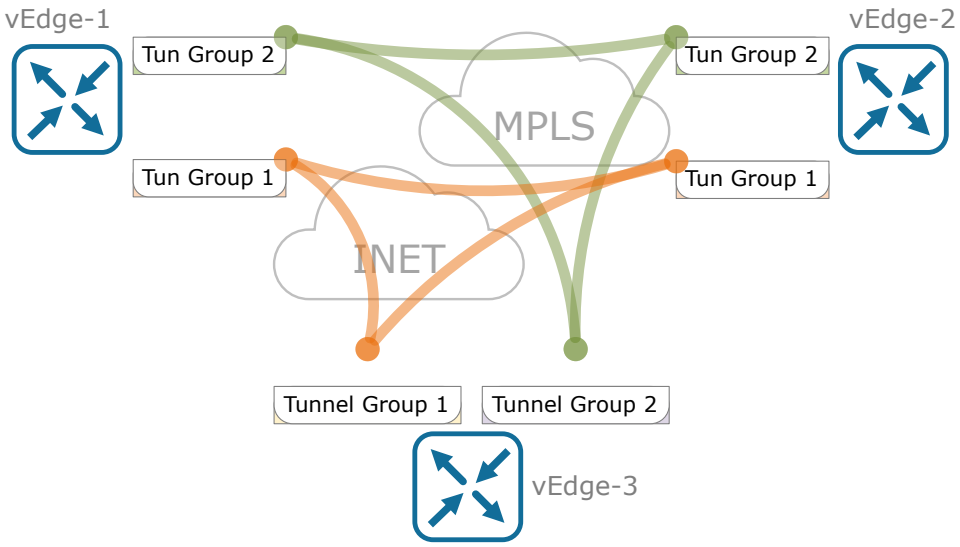


Figure 2. Multiple colors combined in two tunnel groups (no restrict option used)

Use Case 3: Grouping different meshes

Another typical use-case would be if we like to achieve groupings of meshed tunnels as it is illustrated in figure 3. All interfaces in the left tunnel-mesh are configured with group-id 10 and all interfaces in the right tunnel-mesh are assigned a group-id of 20. However, the key point of this example is that the hub routers don't have tunnel-group IDs configured on their interfaces, so they will form overlay tunnels with all other tunnel-group IDs.

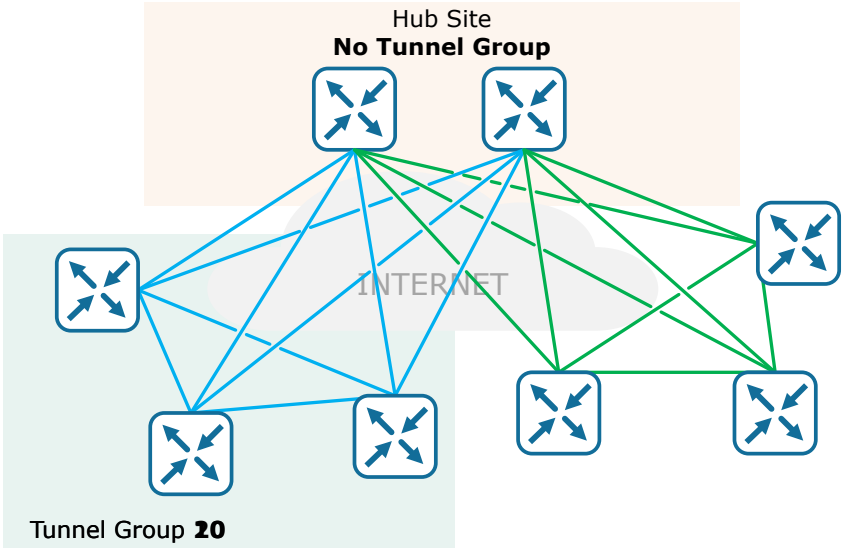


Figure 3. Grouping different meshes (no restrict option used)

Configuring Tunnel-Groups

Configuring this feature is very straight-forward. You just enter in the tunnel configuration mode of a particular interface and enter a group value.

The tunnel group is advertised as an attribute in the TLOC route, as demonstrated in Example 3-10. The possible values for tunnel groups are between 0 and 4294967295.

TLOCs and NAT

NAT Detection

Cisco SD-WAN solution is designed to run over any kind of WAN transport that is available to the WAN edge devices including all different public networks such as Broadband, 4G/5G, LTE, Business Internet, and so on. This implies that the overlay fabric should be able to form through all flavors of Network Address Translations that these public networks utilize. In practice, any Cisco SD-WAN device may be unknowingly sitting behind one or more NAT devices. In order to discover the public IP addresses/ports allocated by NAT, Cisco SD-WAN devices use the Session Traversal Utilities for NAT (STUN) protocol defined in RFC5389.

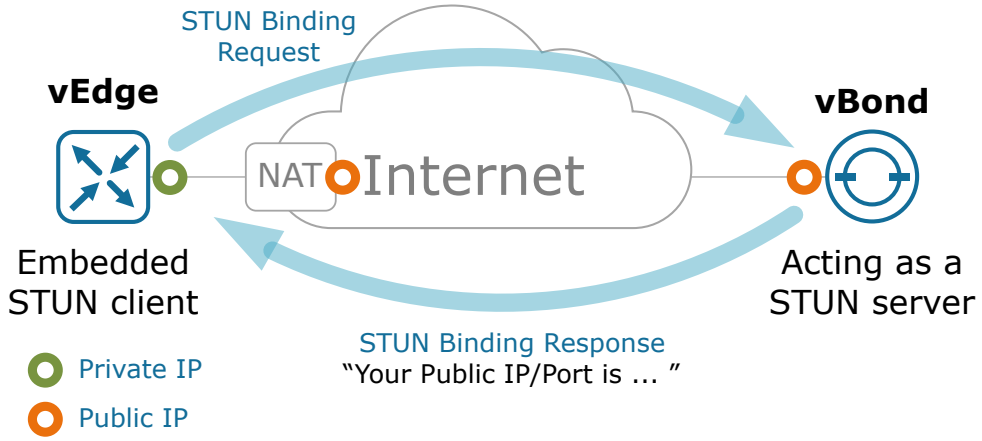


Figure 1. Session Traversal Utilities for NAT (STUN) Operations

STUN is a client-server protocol that uses a request/response transaction in which a client sends a request to a server, and the server returns a response. As the request (called STUN Binding Request) passes through a NAT, the NAT will modify the source IP address/port of the packet. Therefore, the STUN server will receive the request with the public IP address/port created by the closest NAT device. The STUN server then copies the public address into an XOR-MAPPED- ADDRESS attribute in the STUN Binding response and sends it back to the client. Going back through the NAT, the public address/port in the IP header will be un-NATted back to the private ones, but the public address copy in the body of the STUN response will remain untouched. In this way, the client can learn its IP address allocated by the outermost NAT with respect to the STUN server.

As it is shown in Figure 1, all Cisco SD-WAN devices have an embedded STUN client and the vBond orchestrator acts as a STUN Server. When the initial control communication to vBond takes place, the SD-WAN device performs the STUN operations and discovers its public IP address and port. Once determined, this information is then advertised as part of the TLOC routes to the vSmart controllers and then re-advertised to all other SD-WAN devices.

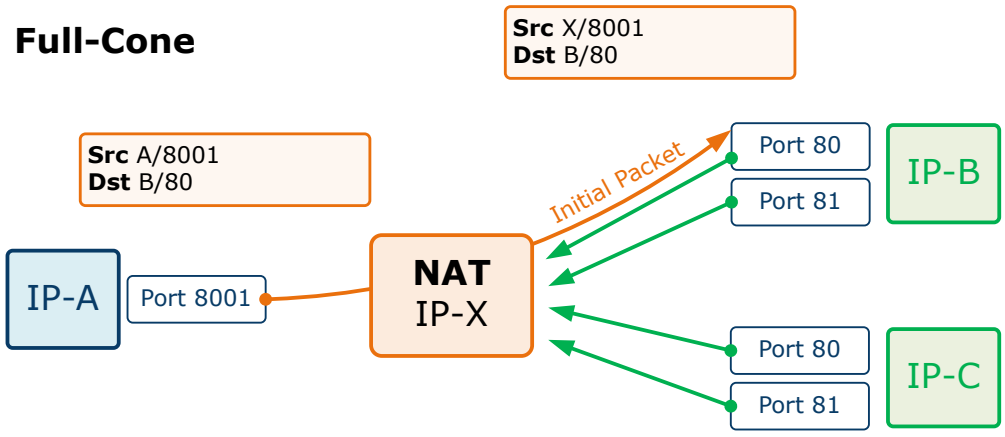
NAT Types

In a typical production SD-WAN deployment, we would probably have many remote sites connected via many different Internet connections to a centralized data center or a regional hub. In most regions in the world, Internet providers will always use some type of private-public address translation due to a shortage of public IPv4 addresses. Let's look at the NAT classifications according to the STUN protocol and how they can affect whether sites can form connections and communicate directly with each other or not.

Full-Cone NAT

A full-cone is one where all packets from the same internal IP address are mapped to the same NAT IP address. This type of address translation is also known as One-to-One.

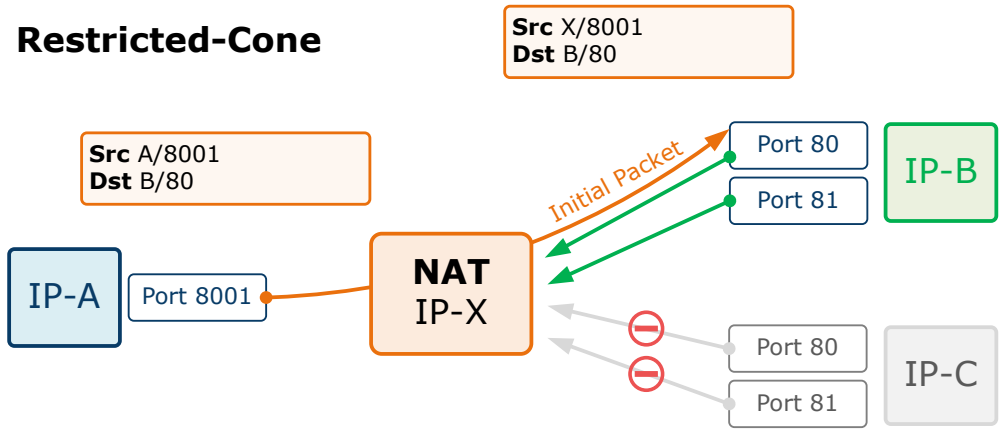
Additionally, external hosts can send packets to the internal host, by sending packets to the mapped NAT IP address.



Full-Cone NAT

Restricted-Cone NAT

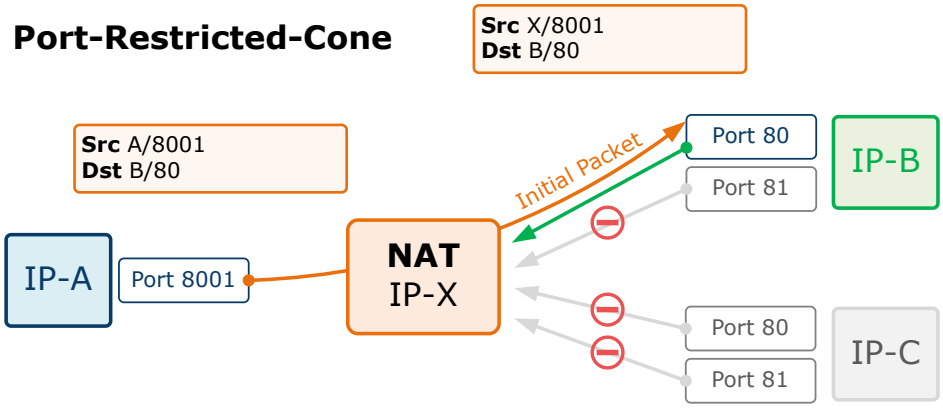
A Restricted-Cone network address translation is also known as Address-Restricted-Cone. It is a network translation technique where all packets from the same internal IP address are mapped to the same NAT IP address. The difference to a Full-Cone is that an external host can send packets to the internal host only if the internal host had previously sent a packet to the IP address of the external destination. It is important to note that once the NAT mapping state is created, the external destination can communicate back to the internal host on any port.



Restricted-Cone NAT

Port-Restricted-Cone NAT

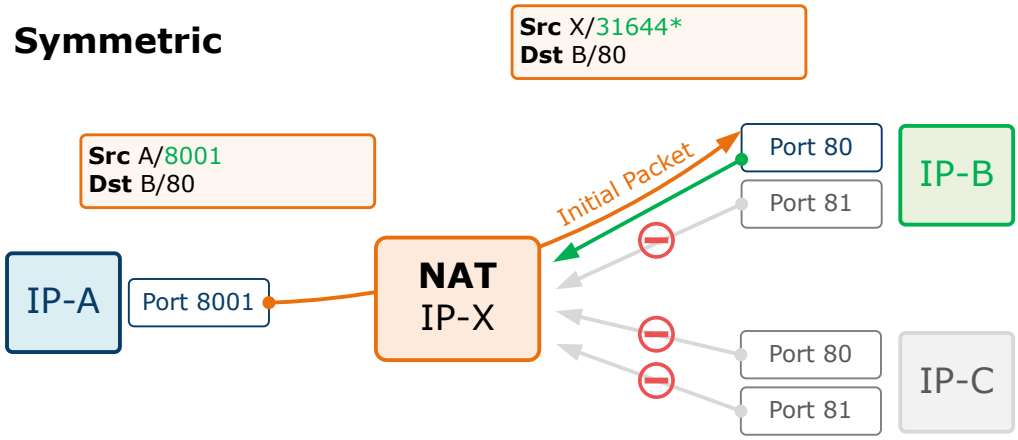
A Port-Restricted-Cone is similar to the Restricted-Cone address translation, but the restriction includes also port numbers. The difference is that an external destination can send back packets to the internal host only if the internal host had previously sent a packet to this destination on this exact port number. In a typical Cisco IOS/IOS-XE or Cisco ASA configuration, this feature is known as Port Address Translation (PAT).



Port-Restricted-Cone NAT

Symmetric

Symmetric NAT is also known as Port Address Translation (PAT) and is the most restrictive of all other types. It is a network translation technique where all requests from the same internal IP address and port to a specific destination IP address and port, are mapped to a unique NAT IP address and NAT port. Furthermore, only the external destination that received a packet can send packets back to the internal host. In a typical Cisco IOS/IOS-XE or Cisco ASA configuration, this feature is known as Port Address Translation (PAT) with port-randomization.



Symmetric-NAT

Best-Practices

Although Cisco SD-WAN supports several types of Network Address Translations, to create a full mesh overlay fabric, at least one side of the WAN Edge tunnels is recommended to be able to initiate a connection inbound to the second WAN Edge. This means that at least one side of the tunnel is recommended to have a public IP address or to be behind a Full-Cone (1-to-1). It is also strongly recommended to configure full-cone, or one-to-one address translation at the data centers or regional hub sites so that, regardless of what NAT type is running at the remote sites (restricted-cone, port-restricted cone, or symmetric), they can send traffic to the hubs without issues.

vEdge-1	vEdge-2	IPsec tunnel can form	GRE tunnel can form
No-NAT (Public IP)	No-NAT (Public IP)	YES	YES
No-NAT (Public IP)	Symmetric	YES	NO
Full Cone (One-to-one)	Full Cone (One-to-one)	YES	YES
Full Cone (One-to-one)	Restricted-Cone	YES	NO
Full Cone (One-to-one)	Symmetric	YES	NO
Restricted-Cone	Restricted-Cone	YES	NO
Symmetric	Restricted-Cone	NO	NO
Symmetric	Symmetric	NO	NO

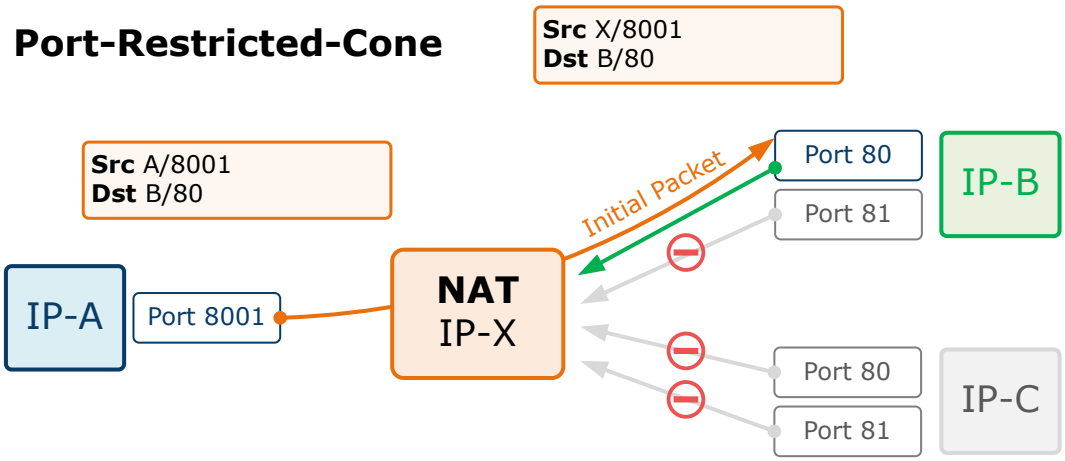
NAT combinations between WAN Edge routers

Symmetric address translation configured at the transport attached to one vEdge requires a full-cone or a public IP on the other vEdge to establish a direct IPsec tunnel between them. Sites that cannot connect directly should be set up in a hub-and-spoke topology so they can reach each other through a regional hub site or data center.

IMPORTANT NOTE that for overlay tunnels configured to use GRE encapsulation instead of IPsec, only public IP addressing or one-to-one address translation is supported. Any type of Network Address Translation with port overloading is not supported since GRE packets lack an L4 header.

TLOC Routes

Once every WAN edge router discovers its private-public translated address and port, it advertises them to the vSmart controller via OMP using the OMP TLOC routes. The vSmart controller then re-advertises this information across the overlay fabric.



An example of two WAN edge router connected through Port-Restricted-Cone

Lastly, let's see an example of two WAN edge devices connected through a Port-Restricted-Cone. As you can verify in the combination table, they are able to form an IPsec encapsulation tunnel between themselves but if we change the encapsulation type to GRE - the data plane tunnel does not come up. Let's quickly verify that.

Data Plane Encryption

Most overlay solutions these days encrypt and authenticate data plane traffic using IPsec and Cisco SD-WAN is no different. Although there is one major difference that Cisco SD-WAN utilizes in order to scale better and more efficiently. Most traditional IPsec environments use Internet Key Exchange (IKE) to handle the key exchange between IPsec peers. However, IKE creates scalability issues in full-meshed environments with thousands of spokes because each spoke is required to manage n^2 key exchanges and n-1 different keys.

Cisco SD-WAN was designed to overcome these scaling limitations by not utilizing IKE at all but instead implement the key exchange within the control plane as shown in Figure 1. This is possible because vEdges identity is established during the provisioning process with the vBond orchestrator.

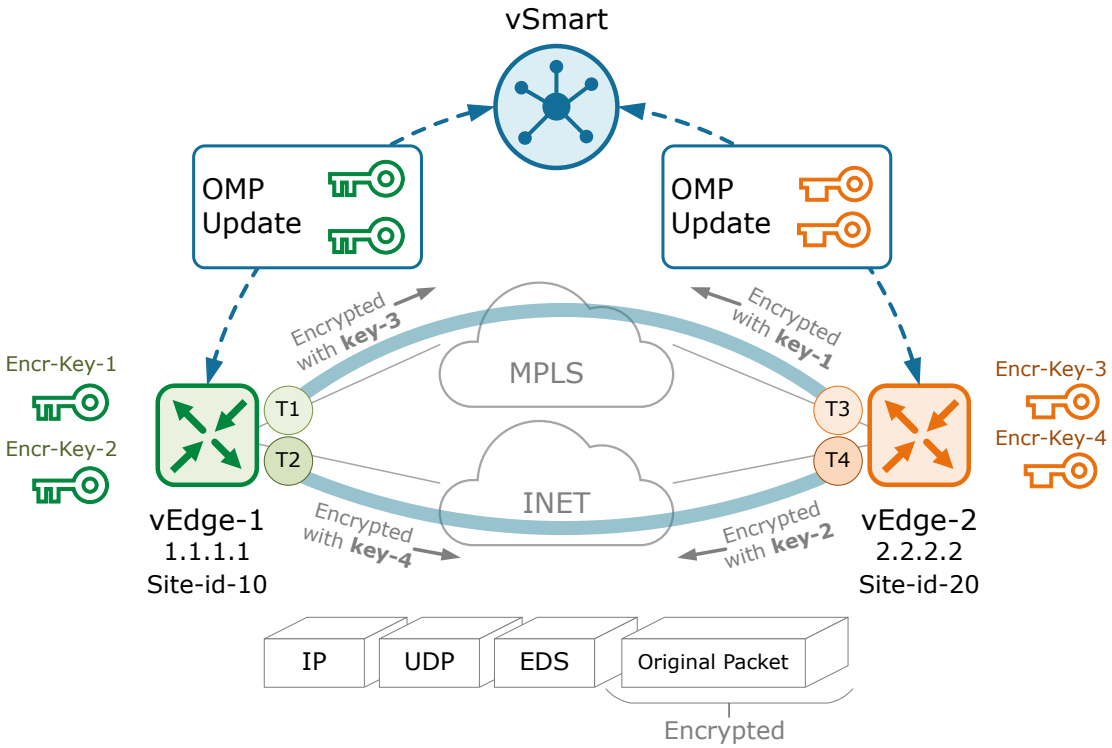
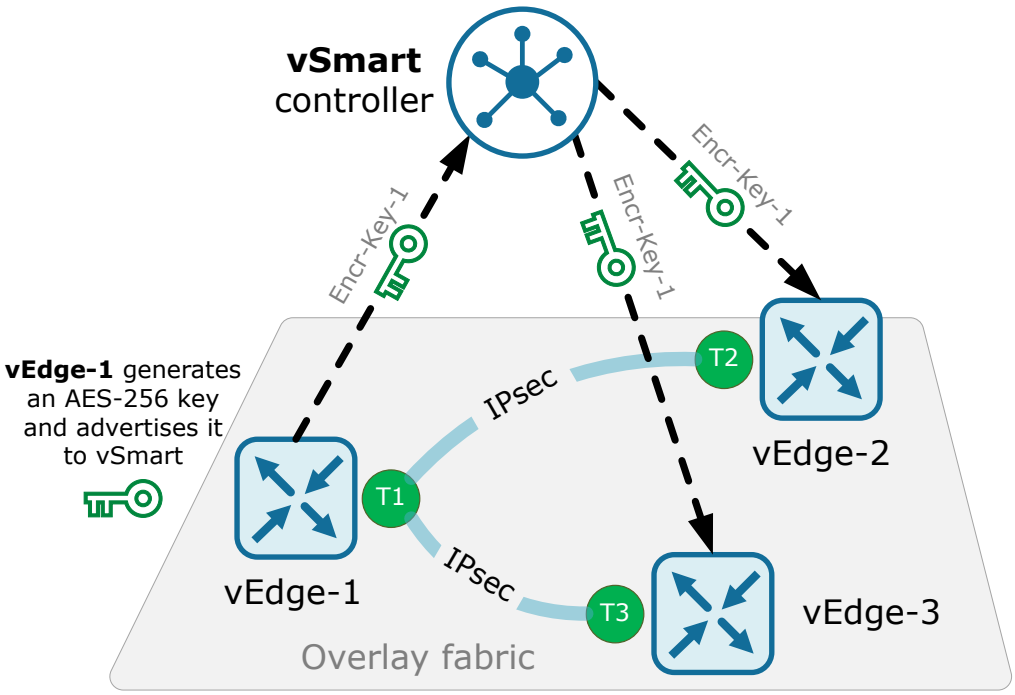


Figure 1. Data Plane Encryption

The main idea is that WAN edge routers can leverage the existing encrypted control connections to the vSmart controller and advertise their keys to the controller via OMP. The controller then redistributes them as OMP updates to all other peers so the exchange is completely done through the SD-WAN control plane.

Let's look at the example shown in Figure 2. vEdge-1 generates an AES-256-bit key for each connected WAN transport. In the example, there is only one transport so there is only one generated key - encr-key-1. However, if we have three WAN providers, three symmetric keys will be generated. Once the encr-key-1 is generated, vEdge-1 advertises it in an OMP update to vSmart, along with the corresponding TLOC T1. This route advertisement is then re-advertised to the rest of the overlay fabric. vEdge-2 and vEdge-3 will then use this information to build their IPsec tunnels to vEdge-1 and encrypt the data plane traffic with the received AES-256 key.



Cisco SD-WAN Key Exchange

Essentially, this keys exchange model removes the burden of individual negotiations between WAN edge devices that using IKE would have brought. In addition to that, each key lifetime is 24 hours and each WAN edge router will regenerate its keys every 12 hours in order to provide enhanced encryption and authentication. This means that two keys (old and new) are present at any one time. The renegotiation of keys does not affect existing traffic, as it happens in parallel with the existing ones and the old key is still held for another 12 hours so any traffic is accepted using either one.

If we summarize everything we have said up to this point - the Cisco SD-WAN solution exchanges keys between WAN Edges and vSmart controllers and uses symmetric keys in an asymmetric fashion. This means the following:

The same key is used for encryption and decryption of data plane traffic.

WAN edge routers use their remote peer's key to encrypt the data rather than their own when sending traffic over the tunnel.

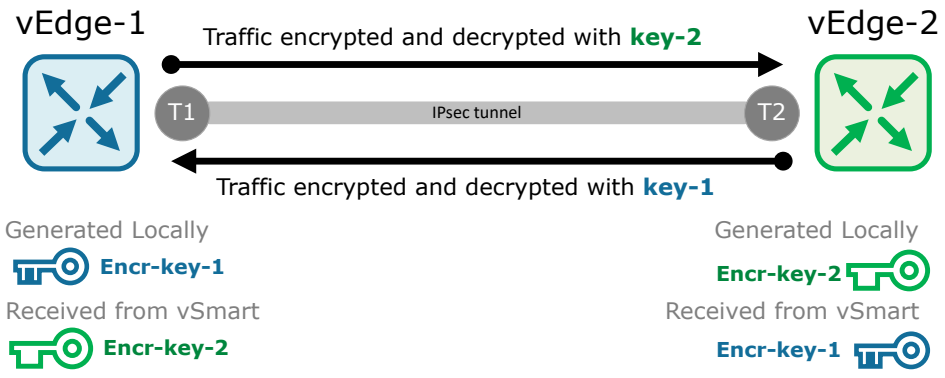


Figure 3. Traffic encryption with Symmetric Keys

Let's look at the example shown in figure 3 where two WAN edge devices are going to communicate over a secure overlay tunnel. Encryption and decryption will occur using the following process:

- vEdge-1 generates an AES-256 key called encr-key-1 and vEdge-2 generates one called encr-key-2.
- Both routers advertise these via OMP to the controller and it distributes them across the overlay.
- When vEdge-1 sends data to vEdge-2, it will encrypt the data using vEdge-2's key.
- When vEdge-2 receives the data, it will use its key for the decryption of that data.
- When vEdge-2 sends data to vEdge-1, it will encrypt the data using vEdge-1's key.
- When vEdge-1 receives the data, it will use its key for the decryption of that data.

Additional security with Pairwise

The IPsec Pairwise keys feature provides additional security by ensuring that multiple vEdge devices across the fabric do not use the same session key for encryption and decryption. The feature functions by generating a pair of IPsec session keys (one encryption and one decryption key) for each pair of local - remote TLOCs. This is visualized in the example shown in figure 4 where there is a fabric with three WAN edge devices. The security improvement comes from the fact that encryption and decryption between vEdge-1 and vEdge-2 will use a session key that is unique to that pair of TLOCs. The green tunnel between WAN Edge 1 and WAN Edge 3 will also use a different session key pair.

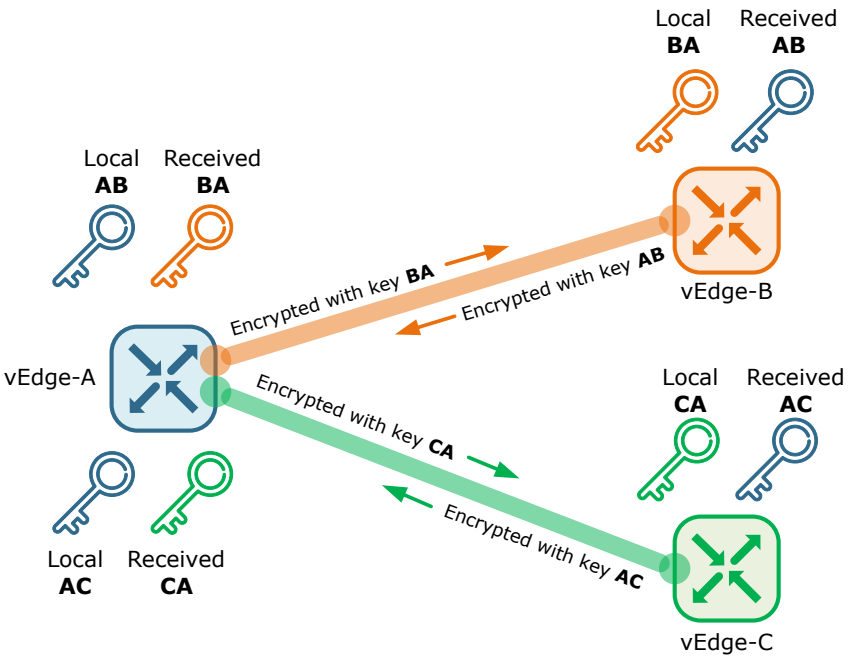


Figure 4. Encryption with Pairwise Keys

Therefore, the process of encryption and decryption of data when using the IPsec Pairwise keys feature will be as follow:

- Each WAN Edge will generate a key for each pair of local-remote TLOC. The session key will then be advertised to the vSmart via OMP.
- The vSmart controller will redistribute the key to the respective peers.
- When WAN edge A sends data to WAN edge B, the IPsec session key AB will be used. In the reverse scenario, WAN Edge B will use the IPsec session key BA.
- When vEdge-A sends data to vEdge-C, the key AC will be used. In the reverse direction, vEdge-C will send traffic using CA.

Another very important thing to note is that the IPsec Pairwise feature is backward compatible with devices that don't support pairwise keys. The feature is disabled by default on the Cisco SD-WAN device and can be enabled via templates.

VPN Segmentation

Traffic isolation is a key part of the security strategy of any company these days. Network segmentation has many use cases within the business such as:

- A company wants to keep traffic from different business verticals separate.
- An organization wants to keep guest users separate from authenticated users.
- An enterprise wants to grant access to partners and suppliers only to some portions of the network.
- A company needs to enforce regulatory compliance such as ISO27001 or the Payment Card Industry (PCI) security standards.

In traditional networking, the most rudimentary forms of network segmentation are VLANs (virtual LANs) at layer 2, and VRFs (virtual routing and forwarding) at layer 3. However, these technologies are limited in scope because they are implemented in a locally significant fashion (on a single device per interface) or require complex control plane interactions to work (for example MPLS-based BGP L3VPNs).

Cisco SD-WAN uses a simpler but more scalable approach to network segmentation using two very efficient techniques:

- Enforcing segmentation at the edges on WAN edge routers.
- Carrying the segmentation information in the data plane packets by inserting VPN labels that identify the network segment.

Cisco SD-WAN Pre-defined VPNs

By default, the Cisco SD-WAN solution has two pre-defined VPNs that cannot be deleted or modified: Transport VPN 0 and Management VPN 512.

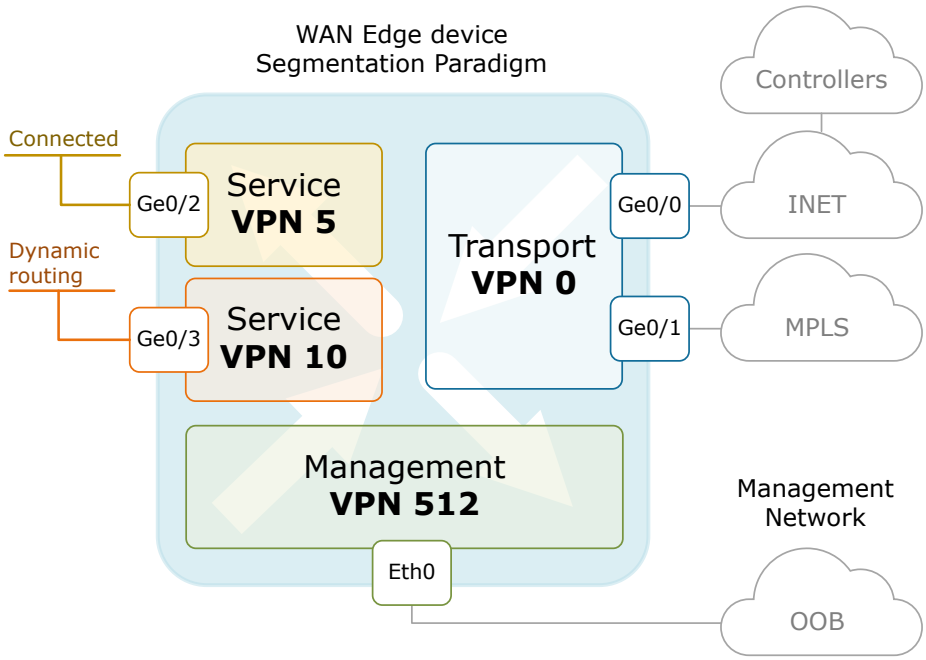


Figure 1. Cisco WAN Edge Segmentation

Transport VPN 0

VPN 0 is the pre-defined Transport VPN of the Cisco SD-WAN solution. It cannot be deleted nor modified. The purpose of this VPN is to enforce a separation between the WAN transport networks (the underlay) and network services (the overlay). Therefore, all WAN links such as Internet and MPLS circuits are kept in VPN 0 as is visualized in figure 2.

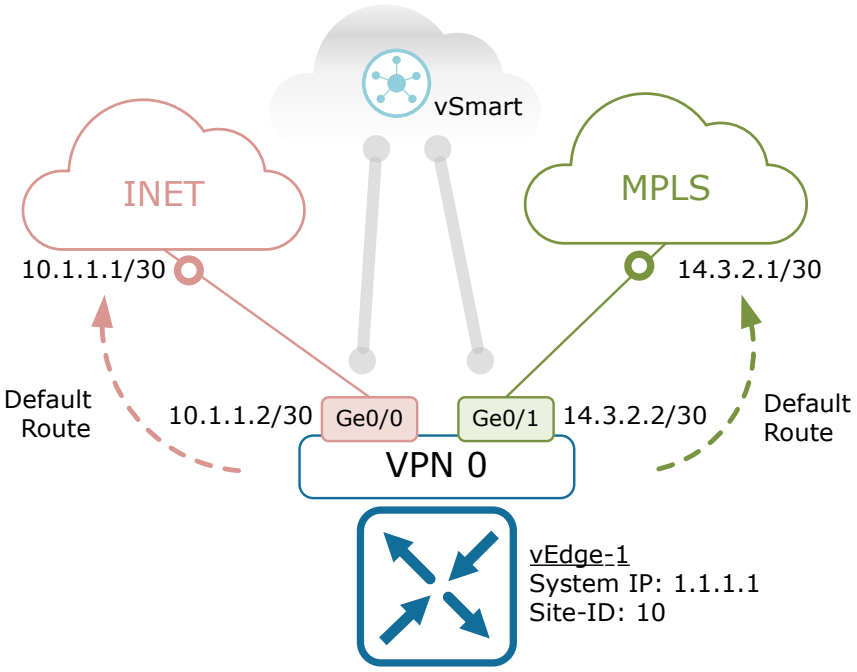


Figure 2. Cisco SD-WAN Transport VPN 0

Additionally, WAN edge routers must have at least one interface configured in VPN0 in order to establish the control plane tunnels to the vSmart controllers. Each interface that is connected to the WAN must have an IP address, color, and encapsulation type configured. These parameters are then advertised to the controllers via OMP as part of the TLOC route advertisements. Typically, a default route is defined via each WAN interface. Note that in Transport VPN0, multiple default routes can exist. In the underlay routing, the actual default route that is chosen depends on the overlay tunnel that is going to be used. When the overlay routing decides to use a particular IPsec tunnel, the underlay routing uses the default route that has a next-hop IP address in the same subnet as the tunnel source IP address.

Let's elaborate on that with the example shown in figure 2. When vEdge-1 forwards packets through IPsec tunnels sourced from interface ge0/0, in the underlay it will use the default route with next-hop 10.1.1.1, because the next-hop address is in the same subnet as the IP address of ge0/0 (10.1.1.2/30).

Management VPN 512

By default in Cisco SD-WAN, VPN 512 is configured for out-of-band management. It is enabled and ready-to-go out of the box.

Service VPNs

When we want to create an isolated network domain and isolate the data traffic in this domain from the other user networks onsite, we create a new service VPN on the vEdge routers. This VPN is specified by a number different from 0 (Transport) and 512 (Management). Once the network segment is created, you associate interfaces, enable routing protocols, and other network services such as VRRP and QoS within this VPN. One important thing to point out is that interfaces associated with user segments must not be connected to WAN transports.

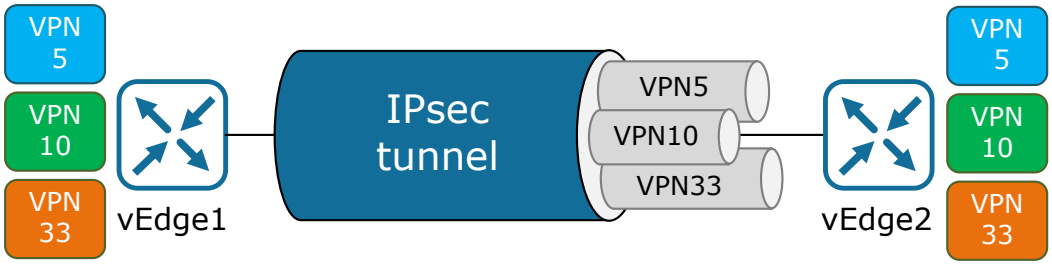


Figure 3. Cisco SD-WAN VPNs

Secure Segmentation

Each interface on a Cisco WAN edge device must be configured in a particular VPN. All prefixes learned via interfaces or routing instances in a VPN are kept in a separate routing table. When this network information is advertised to the vSmart controller, each prefix is associated with a VPN-ID. In addition, the vSmart controller maintains the VPN context of each prefix.

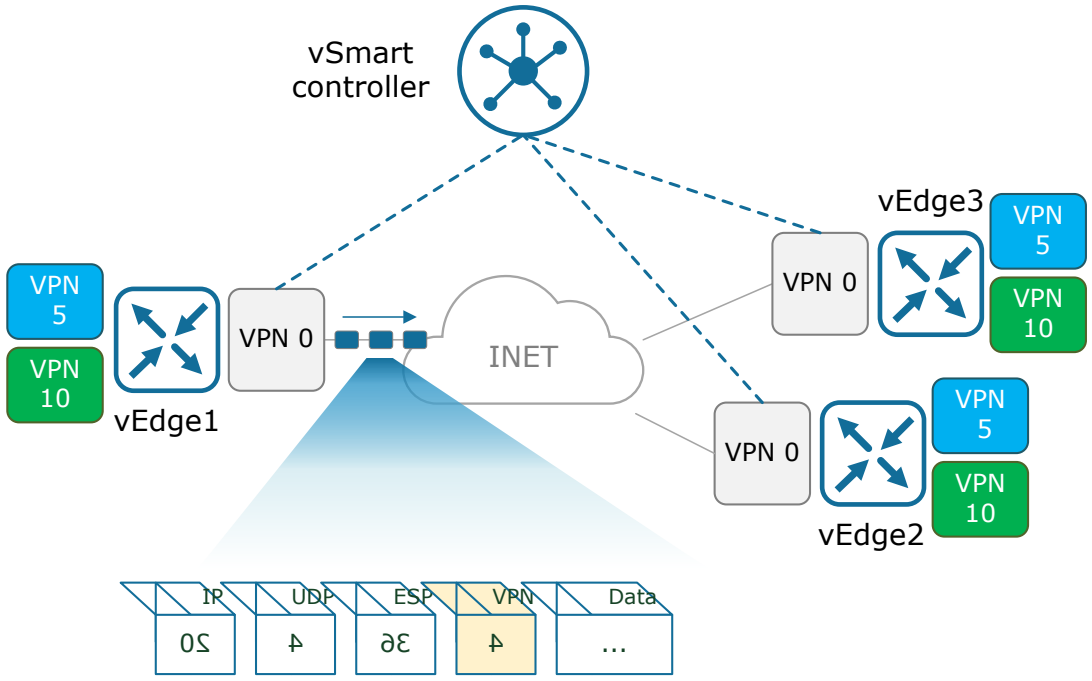


Figure 4. Cisco SD-WAN VPN Lables

This mechanism is very similar to traditional layer 3 isolation with VRFs (Virtual Routing and Forwarding). Therefore, separate route tables provide network segmentation on a single device. However, the question is how to populate this isolated routing information across the overlay domain. At the packet forwarding level, WAN edge routers will insert a new VPN label field in each IP packet. This label will identify the network segments that the packets belong to. The process is visualized in figure 4. When we configure a new VPN on a WAN edge router, it will associates a label to it. The WAN Edge device will then advertise this label along with the VPN-ID to the vSmart controller via OMP. The controller itself will then redistribute this VPN-ID mapping to the other WAN Edges device in the network. The remote WAN Edges in the network will then use these labels to send traffic for the appropriate VPNs, similarly to the label switching in MPLS.

Arbitrary VPN Topologies

Because the control plane is completely separated from the data forwarding plane, Cisco SD-WAN allows us to define different topologies per VPN. Typical topology types are full-mesh, partial-mesh, hub-and-spoke, and point-to-point. By default, if a specific topology is not defined, all VPNs will build a full mesh. Custom segment topologies can be created by using centralized control policies that filter TLOCs or modify next-hop TLOC attributes for specific OMP routes.

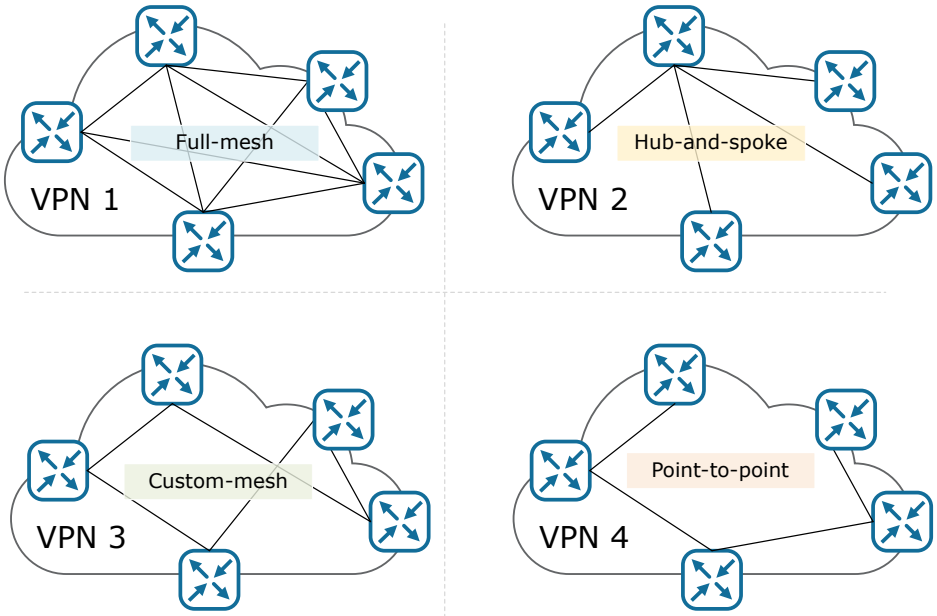


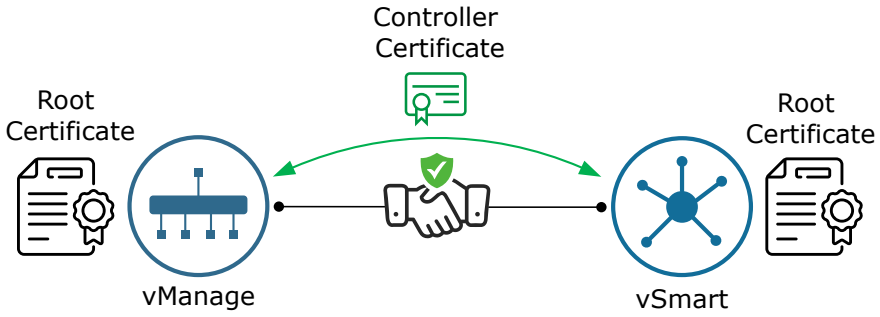
Figure 5. Cisco SD-WAN Arbitrary Topologies

At this point, you might be wondering why custom topologies are important. Well, they are important because some applications may benefit from going via the shortest possible path, e.g VoIP works best in full-mesh topologies. On the other hand, some segments of the network might benefit from controlled connectivity topology such as hub-and-spoke.

5. CISCO SD-WAN DEPLOYMENT

Controllers Identity and Whitelisting

Cisco SD-WAN is designed with the highest level of security in mind. Part of the security stack of the solution is the authentication and authorization of components. Basically, before establishing a control-plane connection, each component must authenticate to each other using their signed certificate, which is validated against the specified chain of trust. The idea is visualized in figure 1 below.



Figuer 1. Cisco SD-WAN Establishing Trust

Controllers Identity

Cisco SD-WAN Controllers can not be brought into operation unless their identity is validated by an established chain of trust. This identity validation process is intended to ensure that only trusted devices can join the SD-WAN solution while still retaining flexibility. Each controller must have a root certificate installed and a controller certificate installed and signed by a trusted CA (Certification Authority). Depending on the deployment method, the chain of trust may be defined in the following manner:

- In cloud and managed deployments, the root certificate chains might be pre-loaded or automatically installed.
- In on-prem deployments, the root certificate is retrieved from an Enterprise CA that can be OpenSSL, Cisco ISE, AWS Certificate Manager, or another private CA option.

For the controller certificates, a CSR (Certificate Signing Request) is generated for every controller through the vManage GUI. Then each certificate request is submitted either automatically or manually to the certification authority. After the certification is signed, it is downloaded and installed on the respective controller. There are several different methods to obtain and install a controller certificate:

1. Automated certificate deployment using Cisco PKI: This is the recommended method according to Cisco's documentation. A network administrator generates a CSR for each SD-WAN controller. The CSR is then automatically sent to Cisco's Public Key Infrastructure. When the CSRs are signed, vManage automatically downloads each signed certificate and installs it on the respective SD-WAN controller. Note that the root certificate that defines the chain of trust is pre-loaded by default on each SD-WAN controller. This method requires vManage version 19.1 or higher.

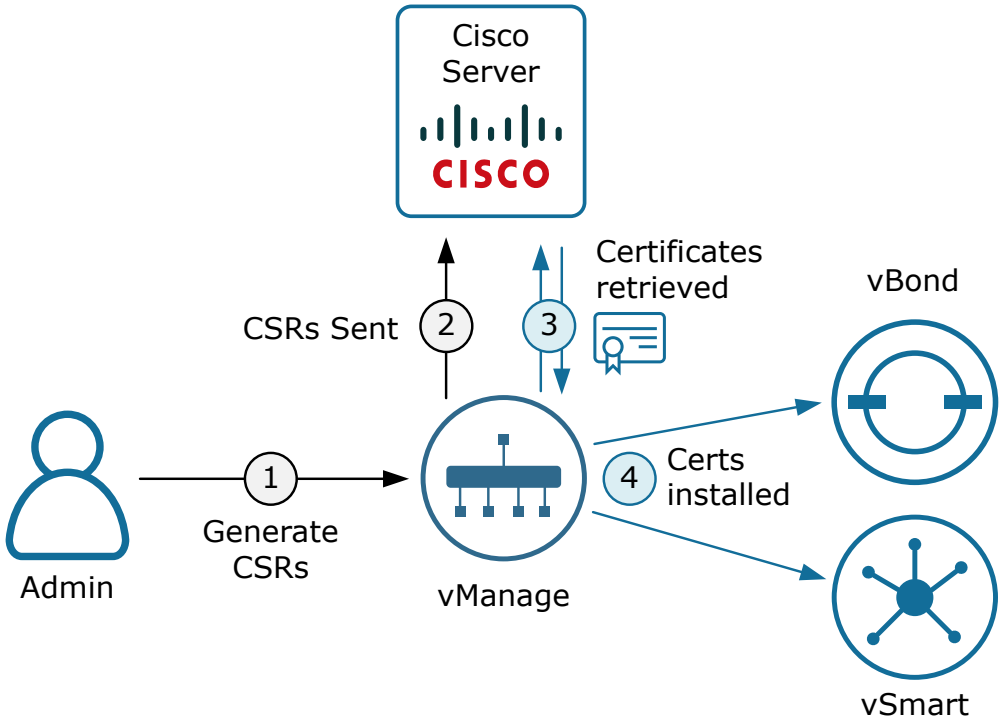


Figure 2. Automated Certificate signing using Cisco PKI

2. Automated certificate deployment using 3rd-party CA: A network administrator generates a CSR for each SD-WAN controller. The CSR is then automatically sent to the Symantec/ Digicert PKI. The difference here is that a Cisco TAC case needs to be submitted in order to complete the certificate signing process. Once the certs are signed, vManage automatically downloads each one and deploys it on the respective SD-WAN controller. Note that again the root certificate that defines the chain of trust is pre-loaded by default on each device.

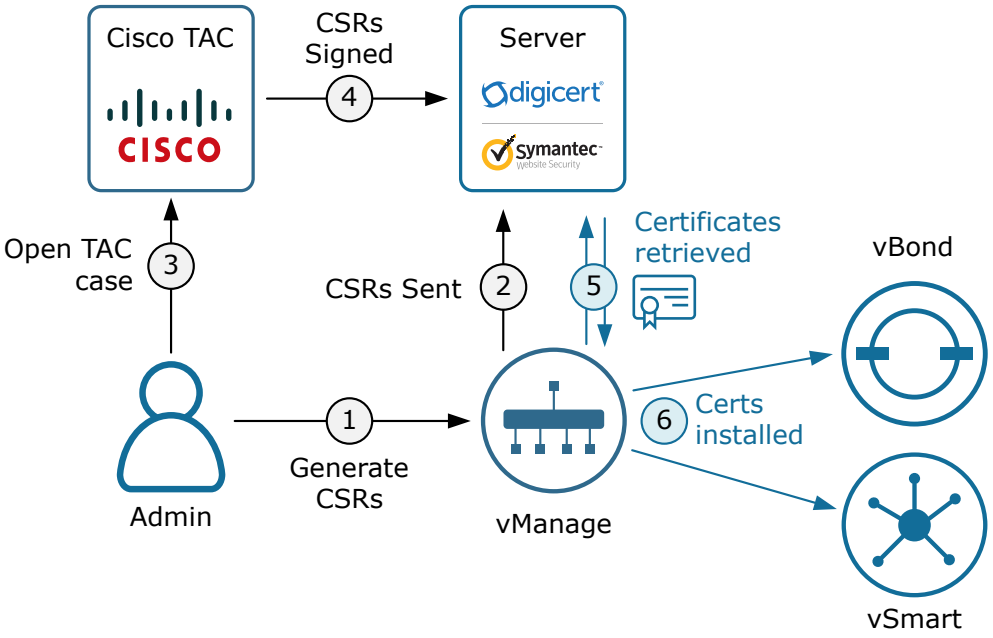


Figure 3. Automated Certificate signing using 3rd-party CA

3. Manual Cisco PKI certificate signing: This option is similar to the automated certificate deployment using Cisco PKI with the only difference that the CSRs are downloaded locally and manually submitted for signing. After the certs are signed, they are downloaded locally and upload manually to vManage. Once that is completed, vManage deploys each cert on the respective controller. Note that again the root certificate that defines the chain of trust is pre-loaded by default on each device.

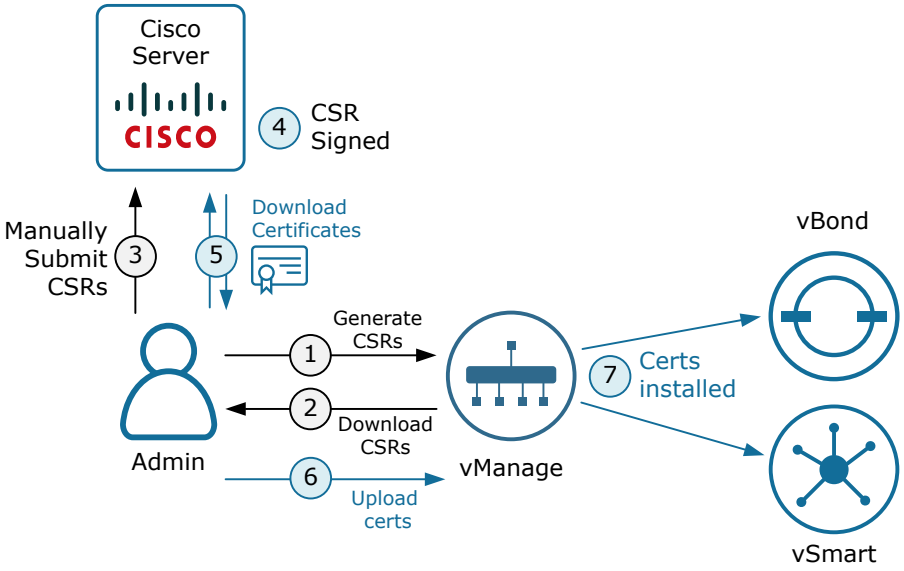


Figure 4. Manual Certificate signing using Cisco PKI

4. Manual third-party certificate signing through Symantec/Digicert: This option is similar to the automated certificate deployment using 3rd-party CA with the only difference that the CSRs are downloaded locally and manually submitted for signing. A Cisco TAC case needs to be opened in order to complete the certificate signing process. After the certs are signed, they are downloaded locally and upload manually to vManage. Once that is completed, vManage deploys each cert on the respective controller. Note that again the root certificate that defines the chain of trust is pre-loaded by default on each device.

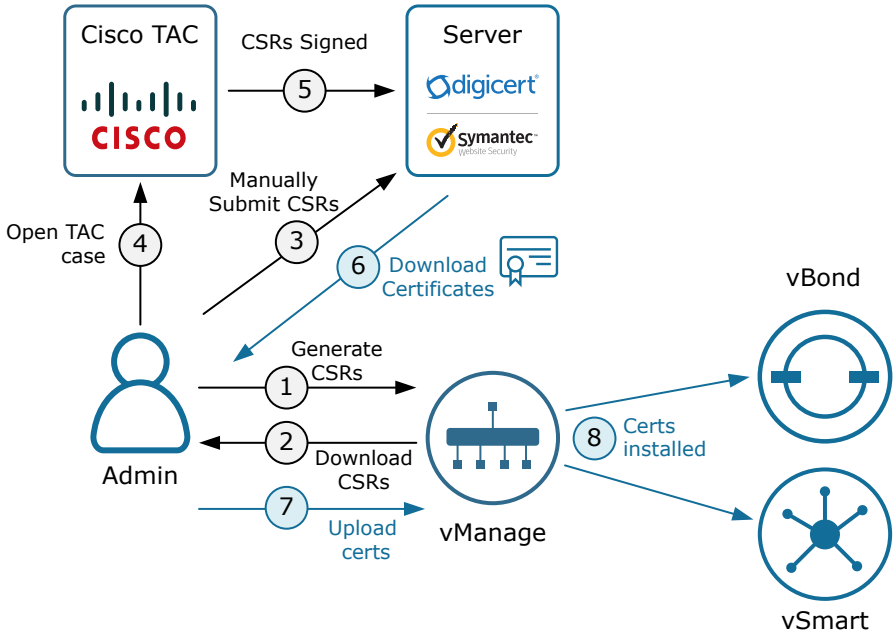


Figure 5. Manual Certificate signing using 3rd-party CA

5. Enterprise Root Certificate Authority (CA): Of course, Cisco SD-WAN provides the flexibility to enterprises to use their private PKI infrastructure and sign controller certs with their own CA. This method requires the most manual steps comparing to the other available options. The first one is to install the Enterprise CA root certificate on vManage, vManage then automatically distributes this root cert to the other configured controllers. This process is visualized with the black lines in figure 6. A network administrator then generates the CSRs and makes a request for each controller to the Enterprise Root CA. This process is visualized with the blue lines in the figure below. Once the certificates are signed, they are manually uploaded to vManage. vManage will then deploy each certificate on the respective controller. This is visualized with the green lines.

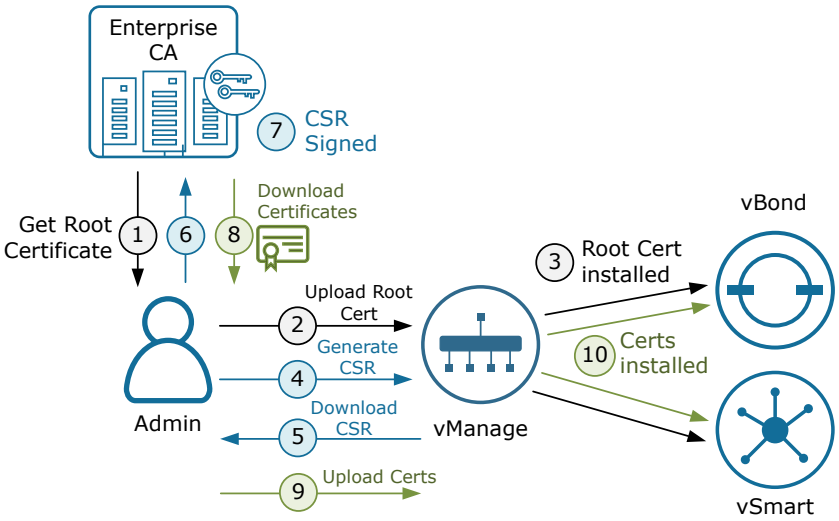


Figure 6. Manual Certificate signing using Enterprise CA

Controller Whitelisting

Cisco SD-WAN employs a whitelisting approach when it comes to adding new controllers into the solution domain. When a network administrator configures a new control device through the vManage GUI, it is automatically added to an authorized list that includes certificate serial numbers of all controllers. This list is then distributed by vManage to all other control-plane devices within the domain. Before establishing control-plane tunnels with each other, the controllers always check whether the remote node's parameters against the authorized whitelist. This technique prevents rouge unauthorized controllers from joining the solution and pushing unapproved configs. However, note that vBond is not checked against the authorized list, but all devices are explicitly configured with the vBond IP address so it basically achieves the same goal.

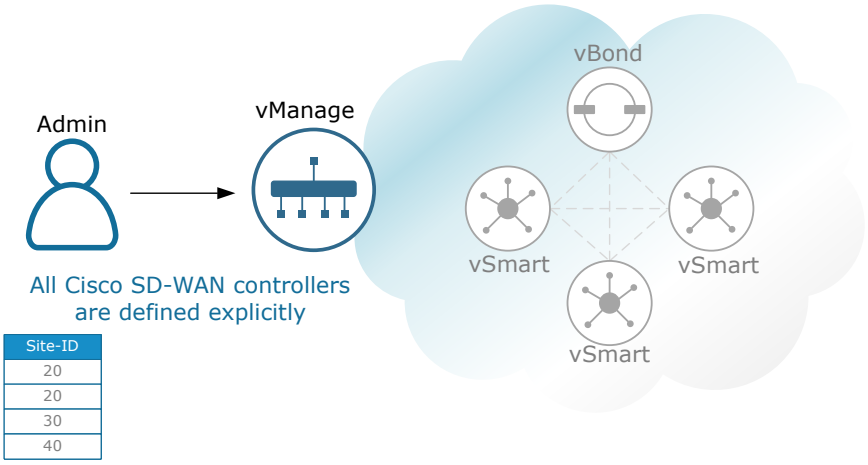


Figure 7. Controllers Whitelisting

WAN Edge Deployment

In this lesson, we are going to go through the different WAN edge onboarding options that Cisco SD-WAN provides.

vEdge devices could be physical appliances or virtual instances. Both types can be onboarded using an automated deployment process, such as Zero Touch Provisioning (ZTP) for Viptela devices and Cisco Plug-and-Play for IOS XE devices. However, there are two available options in case that automated deployment is not possible - manual deployment using the CLI or bootstrap configuration that can be loaded via USB stick.

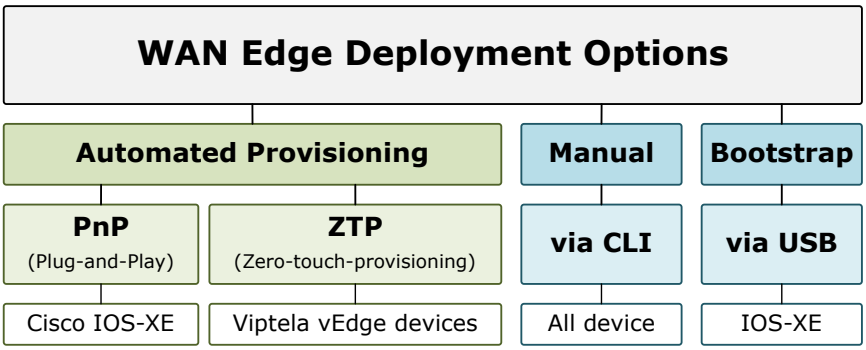


Figure 1. vEdge Deployment Options

It is important to make sure that the following statements are true before a WAN edge device can be onboarded:

All Cisco SD-WAN Controllers (vManage, vBond, and vSmart) should be deployed and operational with valid certificates installed.

The Edge device should have IP reachability to all controllers.

Automated Deployments

The automated WAN edge deployment is the recommended method for adding new nodes to the Cisco SD-WAN fabric. It is enabled by default on all vEdge devices and provides a true zero-touch experience. In essence, this automated onboarding just discovers the vBond IP address dynamically using one of the following processes:

On all Cisco IOS-XE devices, the process is called Cisco Plug-and-Play (PnP). It basically resolves the hostname devicehelper.cisco.com and asks what is the vBond IP for my organization-name.

On all Viptela vEdge appliances, the process is called Zero-Touch provisioning (ZTP). It resolves the hostname vtp.viptela.com and gets the vBond IP for the given organization-name.

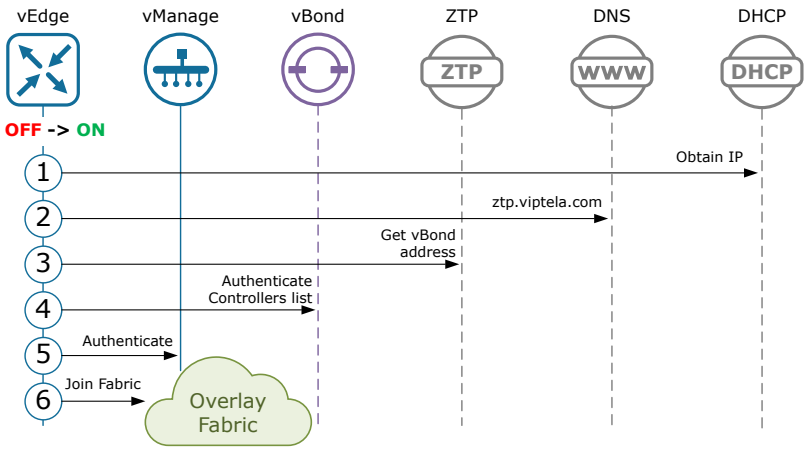


Figure 2. ZTP / PnP Process Sequence

A high-level overview of the steps involved during the Zero-touch Provisioning (ZTP) / Cisco Plug-and-Play (PnP) deployment process is listed below:

0. The WAN edge device is powered up.
1. The vEdge attempts to assign an IP address to its transport interface in VPN0. If it receives IP settings (address/mask/gateway/DNS) via DHCP or Auto-IP, it continues to step 2, otherwise, the automatic deployment does not continue
2. The router tries to resolve the URL ztp.viptela.com (for Viptela vEdge devices) or device-helper.cisco.com (for Cisco IOS-XE device).
3. The device contacts the PnP/ZTP server. The server verifies the vEdge router and sends back the IP address of the respective vBond Orchestrator for this Organization-name.
4. The vEdge establishes a transient connection to the vBond orchestrator. Note that at this point in the automated deployment process, the WAN edge router does not have a system-IP configured, so the connection is established with a NULL system IP address. The Edge authenticates to vBond with a chassis number and serial number. The vBond then sends back the IP address/port of the other SD-WAN controllers as visualized in figure 3.
5. The WAN edge node then establishes a connection to vManage and gets its System-IP address. Then it repeats the process but this time with the correct System-IP (not NULL):
    1. It re-establishes a connection to the vBond using its system IP.
    2. It re-establishes a connection to vManage using its system IP.
    3. vManage then pushes the full configuration to the WAN edge routers.
6. The router establishes a connection to vSmart and joins the overlay fabric.

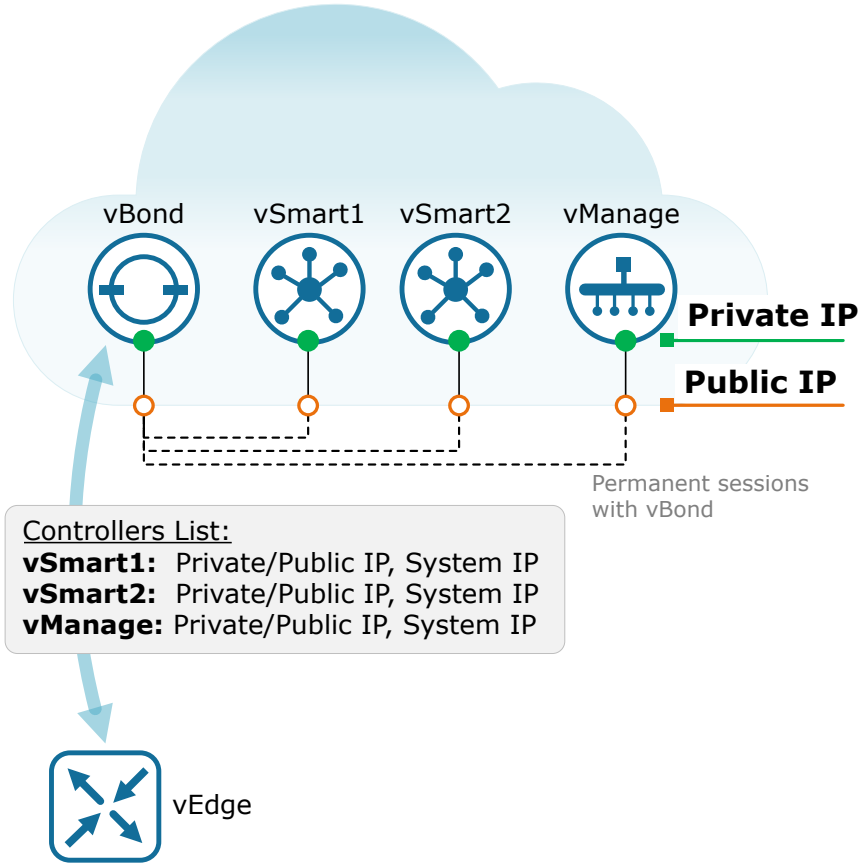


Figure 3. vBond sends back the SD-WAN controllers list

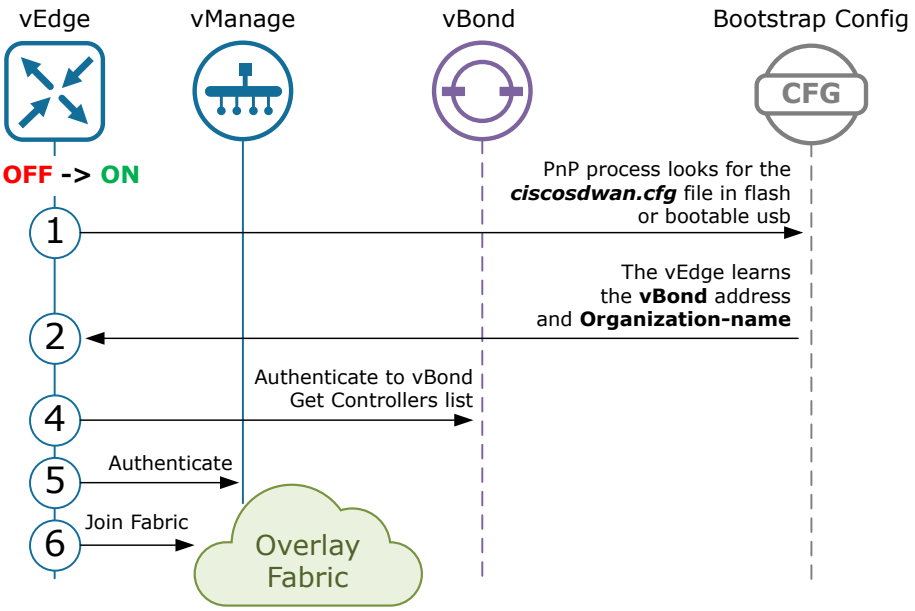


Figure 4. Bootstrap Onboarding Process Sequence

The bootstrap onboarding process sequence is listed below:

- At bootup, a WAN Edge router searches its boot flash memory for a configuration file with a specific filename based on the platform. If the file is not present, the PnP process continues to search for the file on all connected USB sticks. If it manages to find the file, it loads the config, otherwise, the process does not continue further
- If the config file is successfully loaded, the WAN Edge router learns the vBond IP address and organization name and establishes a secure connection to the vBond orchestrator. The Bond sends back the controllers-list.
- The WAN Edge router then establishes secure connections to vManage and vSmart, downloads its configuration using NETCONF over SSH (TCP 830) from vManage, and joins the SD-WAN overlay fabric.

Manual Deployment

The manual onboarding option is something that Network Engineers are pretty familiar with. The WAN edge device is basically configured via the console port or using the KVM/ESXi virtual console connection if the device is a virtual one.

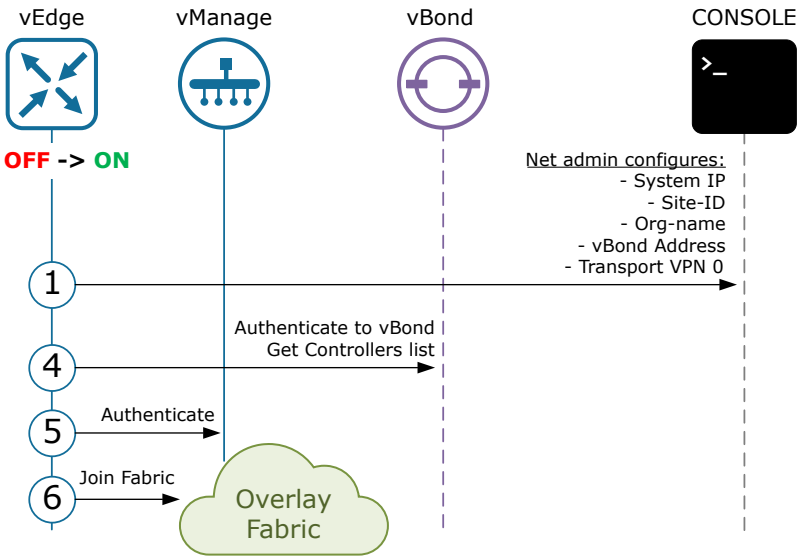


Figure 5. Manual Onboarding Process Sequence

The minimum configuration that is required to successfully onboard a WAN edge router is as follow:

- System-IP, Site-id, Organization-name, vBond IP address.
- VPN 0 interface with IP address/mask, default route, and tunnel interface.

WAN Edge Authorized List

If you have been going through the lesson very closely, you should have noted that the automatic authentication of vEdges can only occur if the vBond/vSmart knows the serial and chassis numbers of the WAN edge routers. The SD-WAN controllers learn this information through a document called vEdge authorized list. This provisioning file can be downloaded from the Cisco Software Central > Plug and Play Connect portal and then uploaded to vManage, which then, sends the list to all vSmart and vBond controllers.

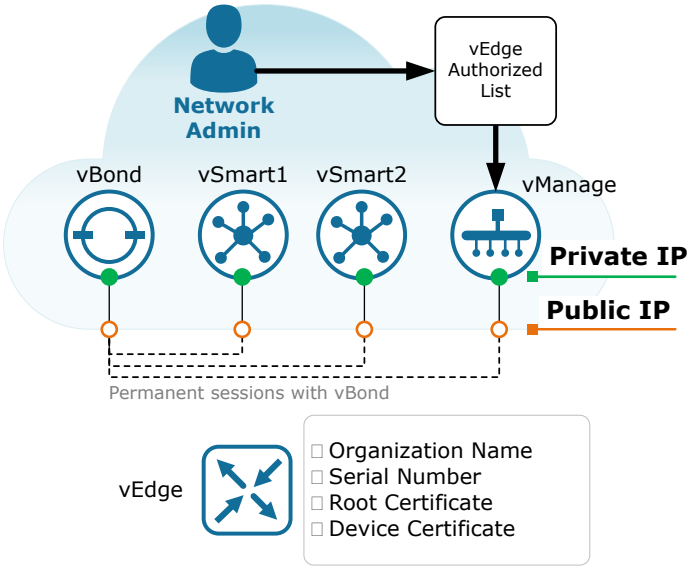


Figure 6. vEdge authorized serial numbers list

This process will be covered in more detail in the lab lessons about vEdge onboarding.

WAN edge deployment behind a Firewall

At the beginning of this lesson, I have written that it is mandatory to have IP reachability from the vEdge to all controllers in order to onboard a device. In reality, the connectivity can be restricted only to the following protocols/ports in case that the solution uses the default DTLS encapsulation.

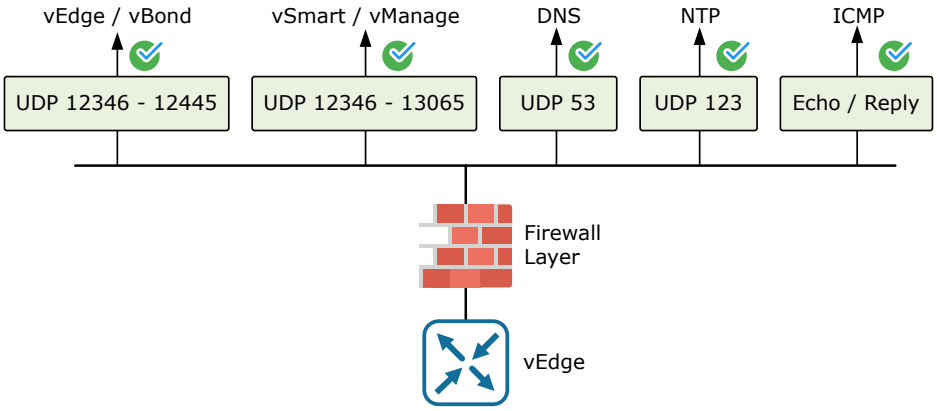


Figure 7. WAN Edge Required FW Ports

Note that, for the certificate authentication to succeed, the time between the WAN Edge routers and the SD-WAN controllers should be synced. That is why NTP should be allowed through the firewall.

Last Resort Circuit

The Business Need

Cisco SD-WAN's device portfolio includes WAN edge routers that support WAN connections over 3G/4G LTE. This is a great option in remote areas where Internet circuits are expensive or not available

However, 3G/4G LTE is not a service provider leased line and is not designed for communicating a large amount of data at a constant rate 24/7. In many parts of the world, an LTE SIM card comes with a data limit that only allows for a certain volume of data to be sent over the LTE line per month. After the data limit is exhausted, either the radio link speed is greatly decreased or there are additional charges for provisioning additional data.

Therefore, in many real-world deployments, where we have a remote site connected to two WAN transports, one of which is LTE, we would generally like to use the LTE radio link only in case the other transport goes down.

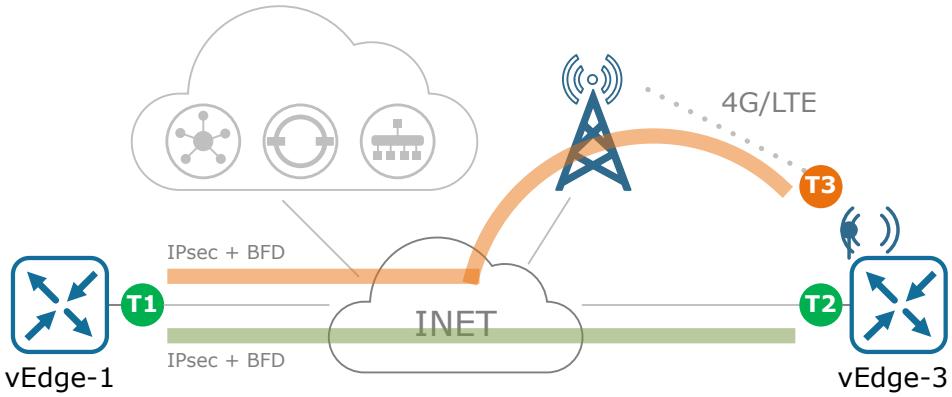


Figure 1. Why do we need a Last Resort Circuit

One way of offloading the traffic from the LTE link is by configuring a higher TLOC preference and higher WEIGHT to the primary WAN transport. This will make sure that in normal circumstances, most of the traffic will pass through the 'primary' tunnel. However, this is not an optimal solution, because even the IPsec tunnel to the LTE TLOC is generating constant traffic. There is a BFD session that exchanges keepalives every second (as shown in figure 1) and there are DTLS control connections via which the vEgde constantly pings the controllers (as shown in figure 2)

These control/overlay connections will still consume a lot of data, even though application traffic does not go over this TLOC.

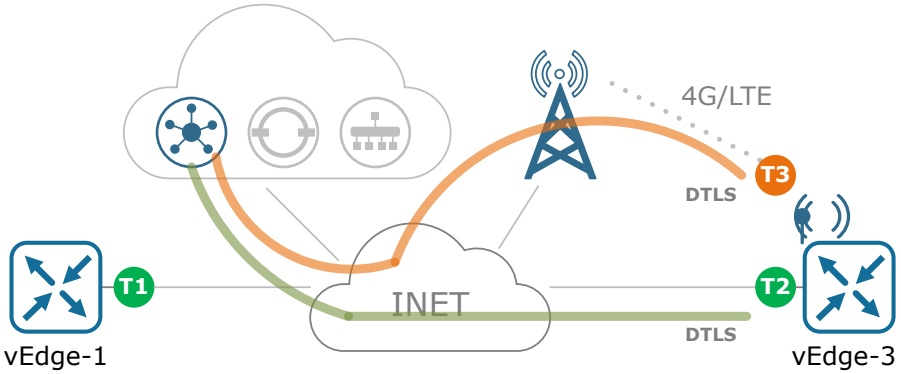


Figure 2. Control Connections over 4G/LTE

Let's verify that on vEdge-3 using the CLI. You can see that there is a BFD session that is UP and the TX interval time is 1 second. Therefore, each second there will be at least two BFD probes to this TLOC (one originated by vEdge-1 and one by vEdge-3). But what if there are multiple WAN edge routers and there are many BFD sessions? Depending on the 4G LTE plan, this may not be very efficient and consume a lot of data unnecessarily.

We can also verify that there are control connections over this orange TLOC. Therefore the WAN edge router is constantly pinging the controllers to make sure they are reachable. This may consume additional data as well.

A better solution - Last Resort Circuit

A better solution to this problem would be to form an IPsec tunnel over this 4G TLOC only in case that the primary WAN transport goes down. Well, Cisco SD-WAN provides such an option in the solution. It is called Last Resort Circuit and is very straightforward and easy to set up.

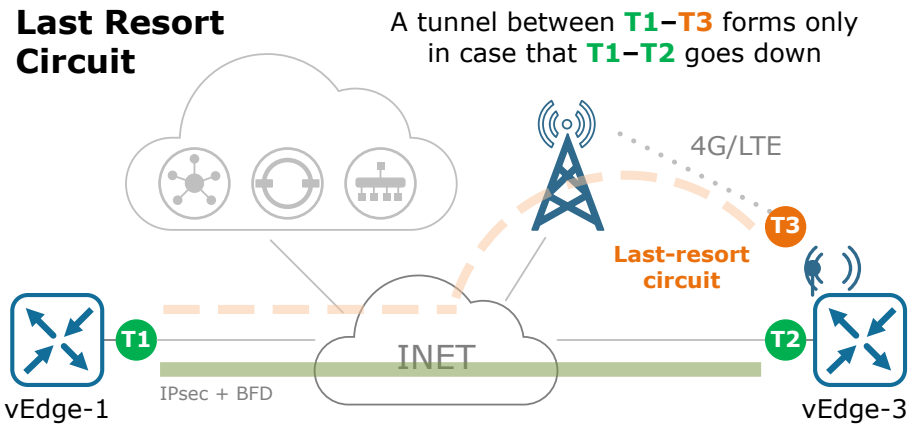


Figure 3. Cisco SD-WAN Last Resort Circuit

The idea is visualized in figures 3 and 4. We would like to advertise the LTE TLOC to the vEdges but only form a tunnel when the primary IPsec tunnel goes down.

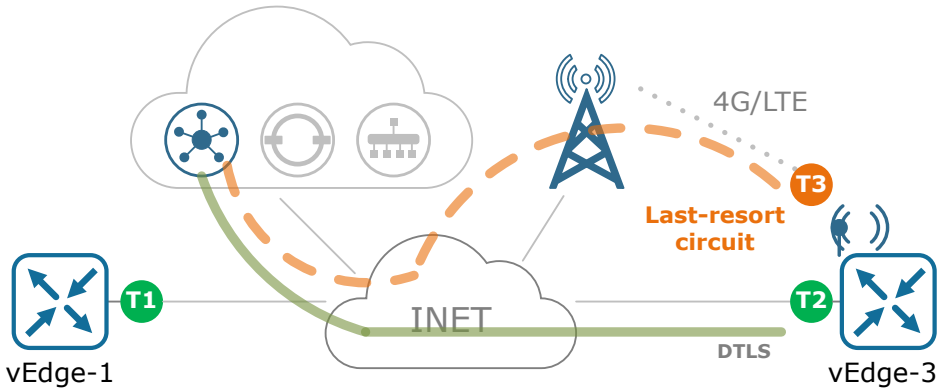


Figure 4. Cisco SD-WAN Last Resort Circuit Control Connections

The same logic applies to the control connections as well. We would like to form a control connection and OMP peering over the LTE TLOC only in case of primary link failure.

Last Resort Circuit Configuration

Let's first check the initial configuration of both TLOCs of vEdge-3. There is nothing out of the ordinary.

That is how simple it is to set up the Last Resort Circuit feature in Cisco SD-WAN. Now let's verify that the feature will work when the primary transport is down.

Verification

To verify that the feature is working, we are going to shut down the primary tunnel and see whether an IPsec overlay will form over the 4G.

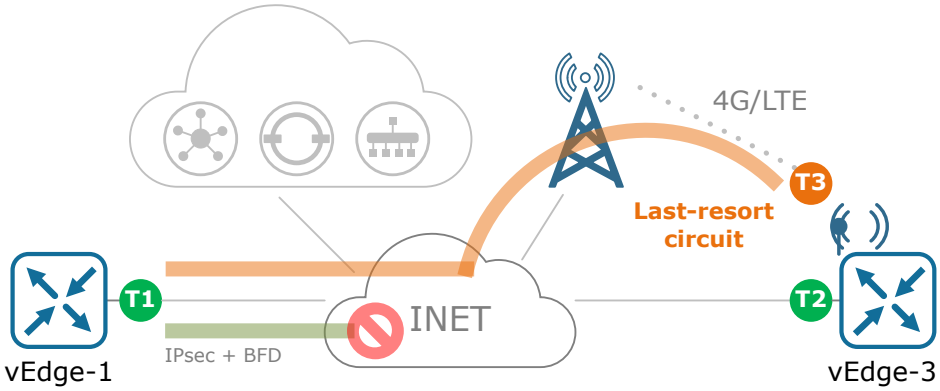


Figure 5. How does the Last Resort Circuit work

You can see that CIsco SD-WAN Last Resort Circuit is a very useful and flexible feature that can be easily deployed at remote sites that use data-constrained WAN transports.

TLOC Extension

In real-world deployments, there will always be cases where the WAN edge routers cannot be directly connected to each available WAN transport as is shown in figure 1. In the example on the left, only one WAN edge device is connected to a single transport. This significantly reduces the overall availability and creates inefficiencies in the overlay fabric. TLOC-extension feature is designed to overcome this problem by extending the WAN transports to both SD-WAN routers without requiring direct attachment to both service provider clouds.

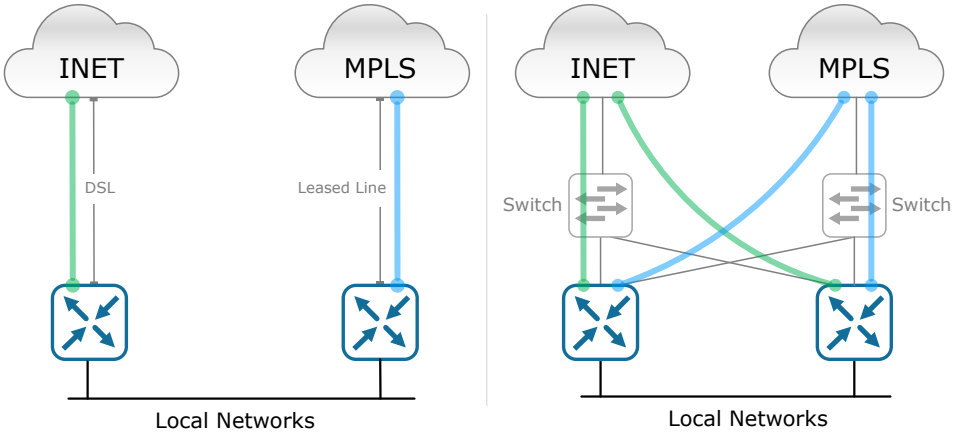


Figure 1. WAN Edges with a Single WAN Trasnport

Of course, the SD-WAN routers can connect to each WAN through front-facing switches as shown on the right but this is not a recommended approach because it adds additional costs and results in having another device to operate.

What is TLOC Extension?

TLOC extension is a feature that allows a WAN Edge router to communicate over the WAN transport connected to the adjacent WAN Edge router through a TLOC-extension interface. In the example shown in figure 2, vEdge-1 is directly connected to the Internet and also uses the TLOC extension feature to connect to the MPLS transport via vEdge-2. In the end, vEdge-1 has an overlay tunnel built over both WAN clouds even though it is directly attached only to one of the WAN transports. On the opposite side, vEdge-2 is attached to the MPLS cloud and uses the TLOC extension feature to connect to the Internet via WAN Edge 1.

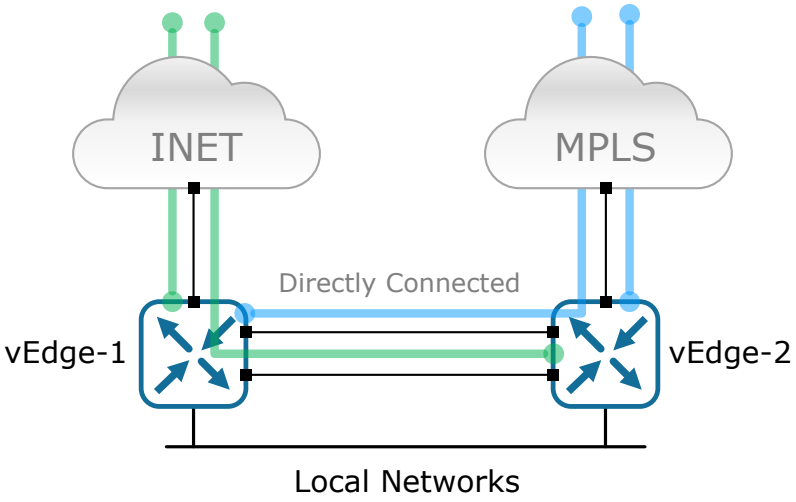


Figure 2. TLOC Extension with Directly Connected vEdges

The feature is set up in a per-interface manner and provides transparent connectivity from one interface (called a TLOC extension interface) to a particular WAN transport. If we take vEdge-1 for example, it is unaware that the blue tunnel to the MPLS cloud passes through another WAN edge device because vEdge-2 extends their directly connected interface transparently to the MPLS transport.

TLOC Extension Types

Cisco SD-WAN allows for multiple ways of implementing TLOC extensions on WAN edge routers. The most typical and straightforward way of extending the transports is by using directly connected interfaces between the vEdges. However, depending on the local site design, the SD-WAN routers can be connected through Layer 2 switches as is illustrated in figure 3. L2 TLOC-extensions describe extensions between vEdges that are Layer2-adjacent are situated in the same broadcast domain/the same subnet.

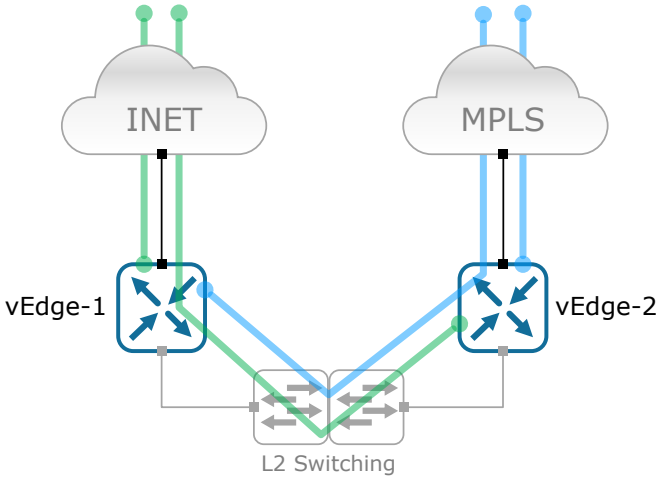


Figure 3. TLOC Extension via L2

The SD-WAN routers can also be connected at Layer 3 via any sort of IP routing. L3 TLOC-extensions describe extensions between two vEdges that are separated by an L3 device and are situated in different IP subnets. L3 TLOC extensions are implemented using GRE tunnels.

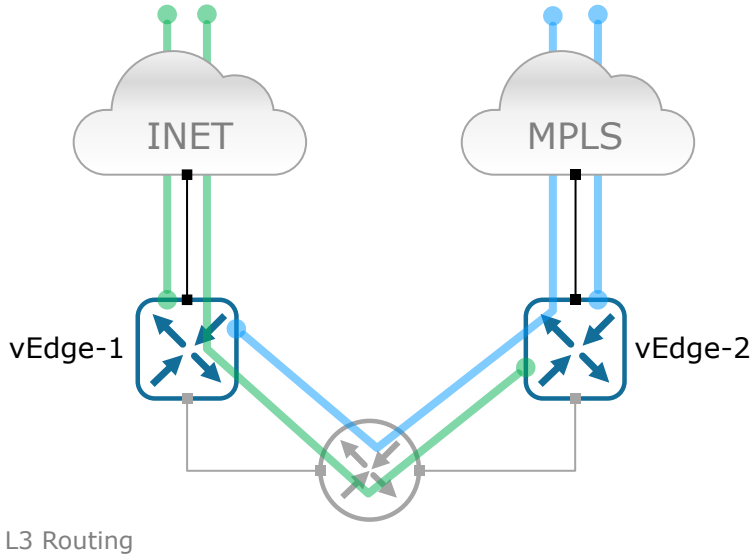


Figure 4. TLOC Extension via L3

I'd also like to point out that TLOC-extensions can be configured using physical interfaces such as Ge0/0, Eth0, etc., and also using L3 subinterfaces.

Notable Limitations

At this point, it is a good time to point out some notable limitation of this feature:

- As you might already know, TLOCs are only supported on routing interfaces. Well, the same applies to TLOC-extension interfaces as well. They are only supported on L3 routed interfaces. Switchports and SVIs cannot be configured as WAN/Overlay ports and can only be used within the service VPNs.
- LTE can not be used as a TLOC extension interface between WAN Edge routers.
- L3 TLOC-extension isn't supported on Viptela vEdge routers but only on SD-WAN devices running IOS-XE.
- The feature would not work on transport ports that are bound to the loopback tunnels.

Configuring TLOC Extensions

Routing Considerations

A TLOC extension interface is always configured in VPN 0 and has an IP address assigned. Then the WAN interface to which it is bound is specified. For example, the vEdge-1's extension interface is ge0/4 and is bound to the Internet through ge0/0 and vEdge-2's extension interface is ge0/5 and is bound to the MPLS cloud through ge0/1. Then static default routes are configured in VPN0 on each WAN Edge router, pointing to the adjacent vEdge's IP as a next-hop.

However, some IP reachability considerations must be done in order for the overlay tunnels and BFD sessions to come up with remote peers over the TLOC extension interfaces. For example, to reach the MPLS cloud, vEdge-1 must be configured with a default route pointing to Ge0/5 of vEdge-2. Another very important part is the reverse IP reachability. To make sure that packets can be routed back to the TLOC extension interface, WAN Edge 2 must advertise subnet B to the MPLS provider. This can be done using any dynamic routing protocol that the service provider is willing to run. In a typical production deployment, some sort of inbound route policy is applied to deny all incoming dynamic routes from the provider since there is a static route pointing to the MPLS cloud.

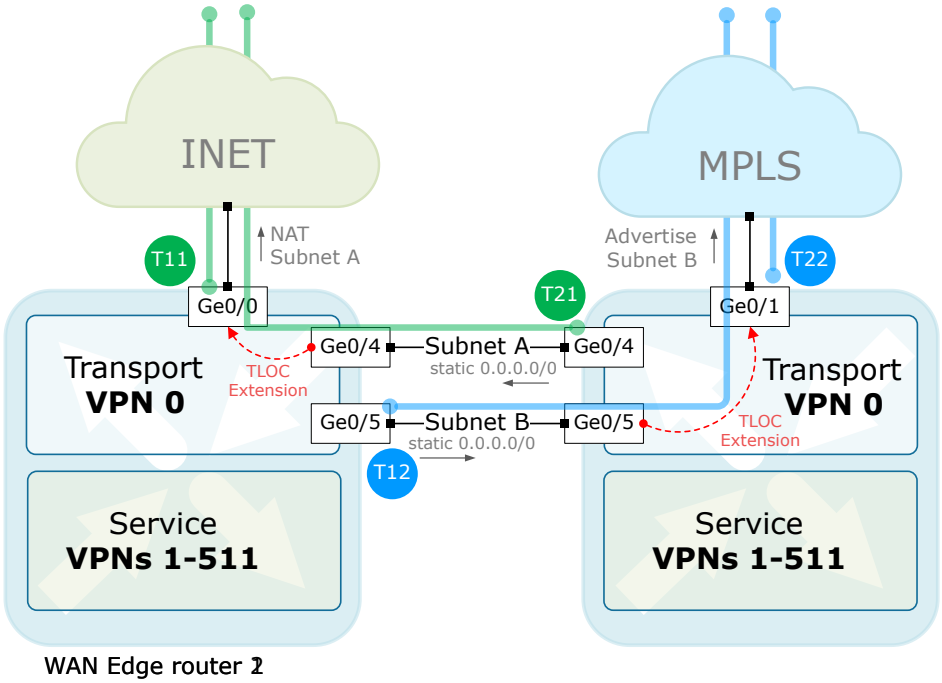


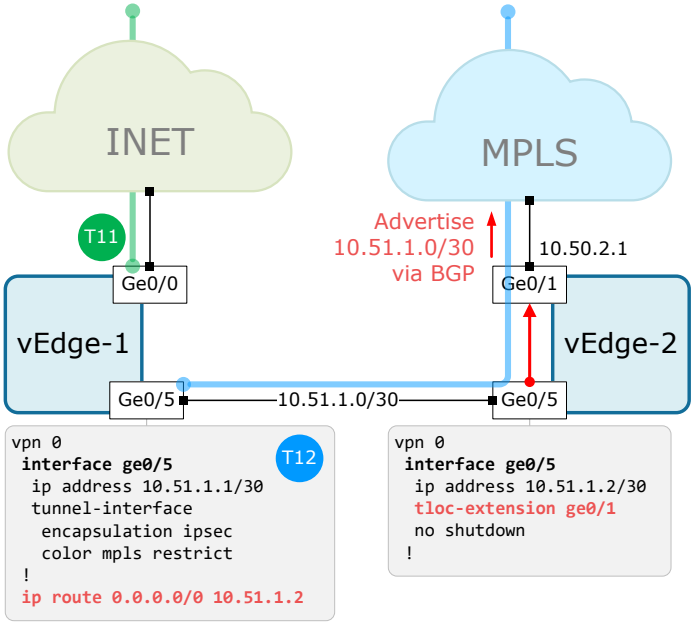
Figure 5.TLOC Extension via L3

In the case of WAN edge router 2, to reach the Internet cloud, it must be configured with a default route pointing to vEdge-1's ge0/4 IP address. In a typical production deployment, subnet-A would be from the RFC1918 address space and NAT must be utilized on WAN edge-1 towards the INET cloud in order to ensure that traffic can be routed back from the Internet to vEdge-2 over the extension interface.

Let's go through a configuration example and see how this feature is implemented.

Extending vEdge-1 to the MPLS

Let's first extend router-1 to the MPLS cloud. For this example, we are going to use the private subnet 10.51.1.0/30 between Ge0/5 interfaces of both routers. This prefix 10.51.1.0/30 must also be advertised to the MPLS provider in order to have IP reachability in the other direction. For this purpose, we are going to utilize BGP.



Configuring vEdge-1

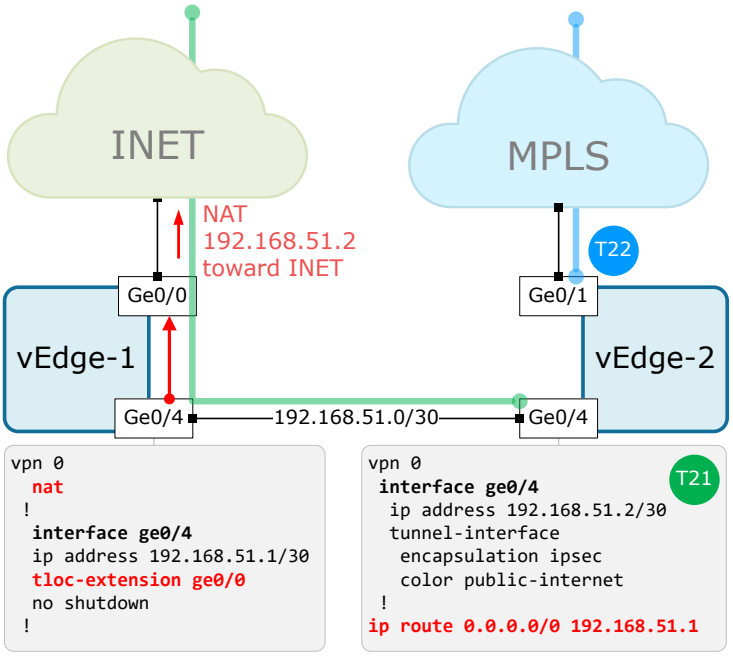
It is important to understand that in order to extend vEdge-1 to the MPLS cloud, the actual TLOC extension happens on the adjacent router. From the perspective of router-1, Ge0/5 is connected to the MPLS provider in the same way as it would have been with a direct attachment link. Therefore, there is nothing special in the configuration on the vEdge-1 as you can see below. The interface is configured with IP address, color, and encapsulation type and a default route is specified. That's about it.

The "magic" happens on the neighboring router-2. In there, we should configure two things. The first one is to tell interface Ge0/5 that it extends the MPLS cloud attached to Ge0/1. The other important thing is to advertise the subnet 10.51.1.0/30 toward the service provider using BGP.

Once both steps are configured, we can see that vEdge-1 has a valid TLOC marked with the mpls color that has a BFD session in UP state even though the router is not directly connected to the MPLS cloud.

Extending vEdge-2 to the Internet

Let's now extend WAN edge 2 to the Internet provider. The configuration is simpler than the MPLS one because we do not need to advertise the interconnecting subnet 192.168.51.0/30 to the Internet provider, but only to NAT it when leaving the transport interface toward the provider.



Configuring vEdge-2

All required configuration is shown in the diagram above. Once the config is applied, we can see that vEdge-2 has a valid TLOC marked with the public-internet color. Note that in the TLOC route, we can see the private-public IP address pair.

If we now check whether there is a BFD session over that color, you can see that there is one is UP state sourced from the TLOC extension interface 192.168.51.2 to a public IP address 80.1.1.1.

6. CISCO SD-WAN MANAGEMENT PLANE

Cisco SD-WAN Policies

In traditional networking, configurations are typically applied on a device-per-device basis using CLI. This leads to a lot of boilerplate code and management inefficiencies. Cisco SD-WAN has been designed to overcome this by implementing a centralized management-plane that administers all devices. The solution uses policies to manipulate the overlay fabric in a centralized fashion and templates to eliminate the boilerplate configurations and reuse code.

What are Cisco SD-WAN Policies?

Policies are an essential part of the Cisco SD-WAN solution and are used to influence the packet flow across the overlay fabric. They are created on vManage through the Policy Wizard GUI and when applied, are pushed via NETCONF transactions either to vSmart controllers (centralized policies) or directly to vEdges (localized policies).

A Cisco SD-WAN policy is the sum of at least one list, that identifies interesting values, one policy definition, that defines actions, and at least one application, that defines where this policy will be applied. This concept is visualized in figure 1.

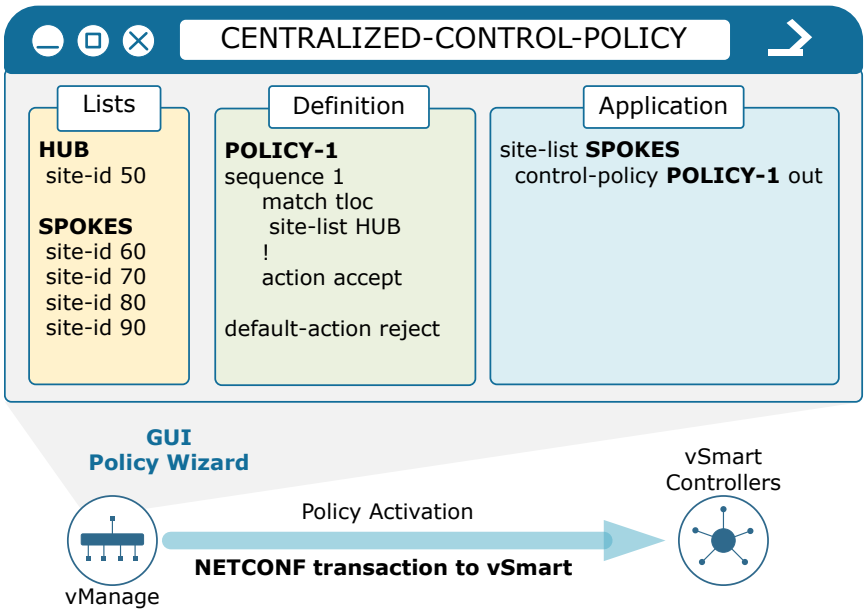


Figure 1. Cisco SD-WAN Policy Construction

It is important to understand that policies are configured on vSmart or vEdge. vManage is only a graphical user interface used to create and store policies, but once a policy is activated through the vManage GUI, it is configured with a NETCONF transaction either on vSmart or vEdge. Therefore, activating a policy via vManage is equal to manipulating the configuration of vSmart. If we define the policy shown in figure 1 using vManage and apply it, we are going to see that it will appear in the running-configuration of vSmart as shown in the cli output below.

We can clearly see the lists marked in green, the policy definition highlighted in yellow, and the policy application colored with orange. Another important nuance is that vSmart does not store policies, it only loads the currently active policy in its running-configuration. All policy versions and revisions are stored on vManage. Therefore vManage is responsible for roll-backs, version control, and making sure that policy changes are persistent across multiple vSmart controllers.

While all policies are defined using the vManage Policy Wizard, different types are enforced on different devices at different locations in the network. This is visualized in figure 2 below. Localized Control and Data policies such as Access-lists, Classification, Marking, Policing, or local-site routing are enforced directly on the WAN edge routers with NETCONF transactions. On the other hand, centralized policies, that affect the whole overlay fabric, are applied to vSmart and only the result of the policy is advertised to the WAN edge routers using the Overlay Management Protocol (OMP).

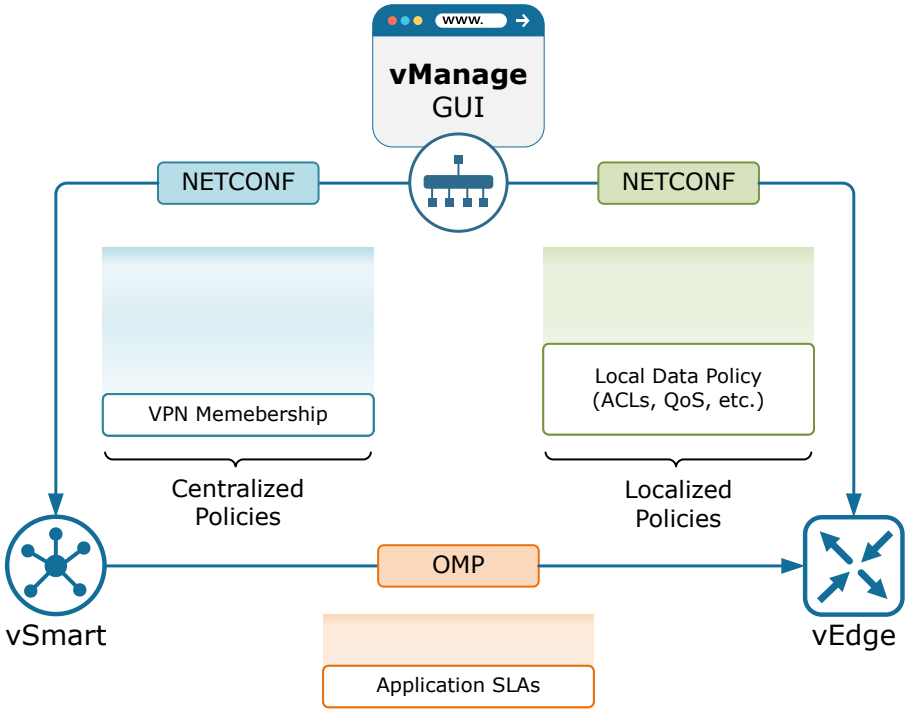


Figure 2. Policy Application Location

At this point, you may be wondering what is NETCONF. It is a popular network management protocol designed to manage remote configurations. Its operations on top of a Remote Procedure Call (RPC) layer and uses XML or JSON for data encoding. Typically the protocol messages are exchanged over Transport Layer Security (TLS) with mutual X.509 Authentication. You do not need to know more about it in order to work with the Cisco SD-WAN Viptela solution.

Cisco SD-WAN Policy Types

As you are already well aware, the Cisco SD-WAN solution follows the SDN principles and separates the control plane from the data plane. It is also designed to be centrally managed but at the same time allows you to make changes on a single device. To encompass these principles, the solution allows us to configure a few different types of policies. Centralized policies allow us to manipulate the whole overlay fabric in a centralized fashion and localized ones give us the ability to manipulate only a particular device or location. Because the control and data plane are separated, centralized policies are also separated into centralized-control-policies that affect the control-plane operations and centralized-data-policies that directly affect the forwarding of packets. The following figure visualizes the Cisco SD-WAN policy's structure.

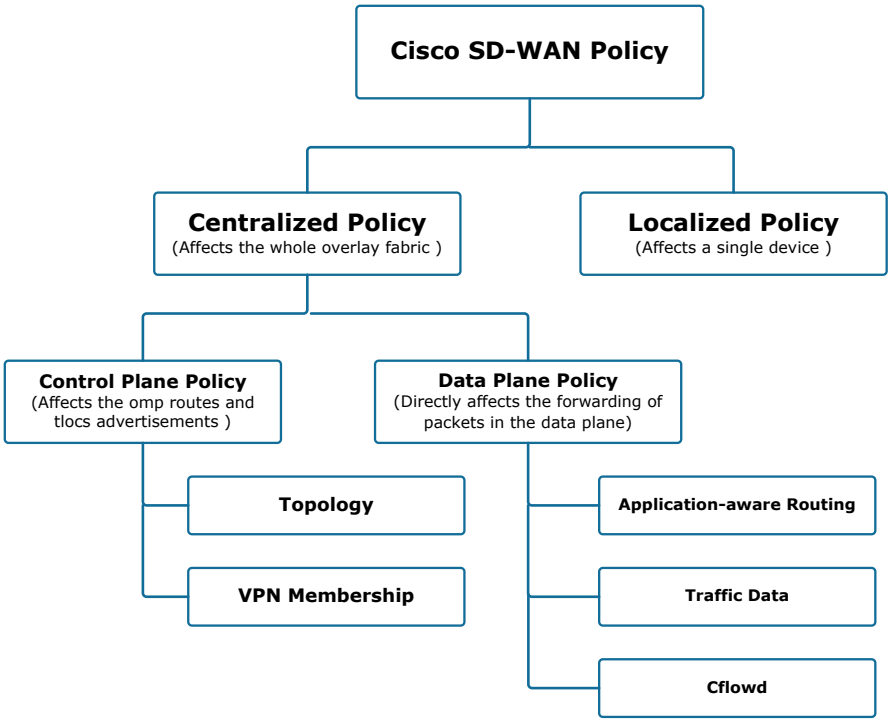


Figure 3. Cisco SD-WAN Policy Structure

Policy Key Points

Cisco SD-WAN policies are not easy to work with at first and the best way to get comfortable with them is to practice. However, before we begin playing around, I would like to highlight some key points that are good to be known beforehand.

A policy is processed in the following order of steps:

- All match–action clauses are processed in sequential order, starting from the lowest sequence number upwards.
- When a match occurs, the configured action is performed and the sequential processing does NOT continue further (all other match-action pairings are skipped).
- If a match does not occur, the configured entity is subject to the default action configured (by default it is reject).

Centralized policies (the ones configured on vSmart) are always applied to a site-list.

- Only one of each type of policies can be applied to a site-list. For example, you can configure one control-policy in and one control-policy out but not two control policies in the outbound direction.
- Cisco does not recommend including a site in more than one site-list. Doing this may result in unpredictable behavior of the policies applied to these site-lists.
- Centralized-Control-policy is unidirectional applied either inbound or outbound. For example, If we need to manipulate omp routes that the controller sends and receives, we must configure two control policies.
- Centralized-Data-policy is directional and can be applied either to the traffic received from the service side of the vEdge router, traffic received from the tunnel side, or both.
- VPN membership policy is always applied to traffic outbound from the vSmart controller.

vEdge Order of Operations

As you have seen by now, there are several different types of policies in the Cisco SD-WAN solution. That basically means that multiple different types can be applied to a given WAN Edge device. To design and implement policies in large-scale deployments, you must have a good understanding of the order of operations of the WAN edge nodes.

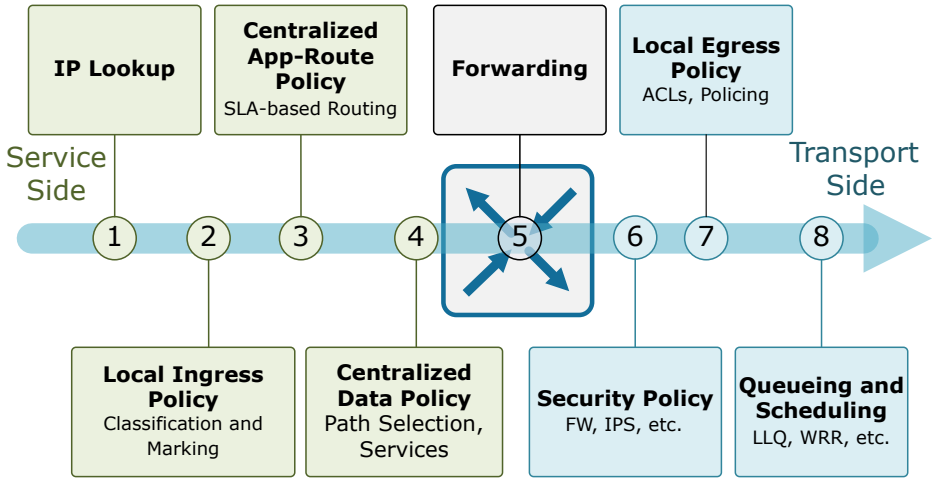


Figure 4. vEdge Order of Operations

Figure 3 visualizes the steps that a WAN edge router takes when forwarding a packet through:

- IP Destination Lookup: WAN edge devices are in essence just routers, so the forwarding decision always starts with IP address lookup.
- Ingress Interface ACL: Localized policies are typically used to create ACLs and tie them to vEdge interfaces. As in traditional networking, these ACLs can be used for filtering, marking, and traffic policing.
- Application-Aware Routing: If there is an Application-Aware Routing policy applied, it makes a routing decision based on the defined SLA characteristics such as packet loss, latency, jitter, load, cost, and bandwidth of a link.
- Centralized Data Policy: The centralized data policy is evaluated after the Application-Aware Routing policy and is able to override the Application-Aware Routing forwarding decision.
- Forwarding: At this point, the destination IP address is compared against the routing table, and the output interface is determined.
- Security Policy: If there are security services attached to the WAN edge node, they are processed in the following sequence - Firewall, IPS (Intrusion Prevention), URL-Filtering, and lastly AMP (Advanced Malware Protection). The necessary tunnel encapsulations are performed and VPN labels are inserted.
- Egress Interface ACL: As with ingress ACLs, local policy is able to create ACLs that are applied on egress as well. If traffic is denied or manipulated by the egress ACL, those changes will take effect before the packet is forwarded.
- Queueing and Scheduling: Egress traffic queueing services such as Low-Latency (LLQ) and Weighted Round Robin (WRR) queueing are performed before the packet leaves.

Cisco SD-WAN Templates

In Cisco SD-WAN, you can apply configurations to network devices with either one of the following two methods:

- Via the CLI - This is the well-known way of configuring network nodes in traditional networking. You connect to the device via TELNET/SSH or CONSOLE and modify the running configuration. As much as we network engineers love the CLI, it has not been designed to make massive scale configuration changes to multiple devices at the same time.
- Via the vManage GUI - This is the recommended centralized approach of configuring the devices in the Cisco SD-WAN solution. It is significantly less error-prone, can easily scale, and has support for automation, backups, and recovery.

Configuration Templates

The actual process of configuring Cisco SD-WAN nodes via vManage is done by applying device templates to one or multiple devices. A device template holds the whole operational config of a device. When vManage provisions the configuration of a node, it acts as a single source of truth and "locks" the device in a configuration mode called "vManage mode". That means that configuration changes can only be applied via vManage and changes via CLI are not allowed.

A device template can be either Feature-based or CLI-based as is shown in figure 1. Something very important about templates is that when we create a CLI-based template for a specific device, the whole configuration of the device must be in the CLI template and not only a specific snippet of the configuration. The opposite is true about feature templates.

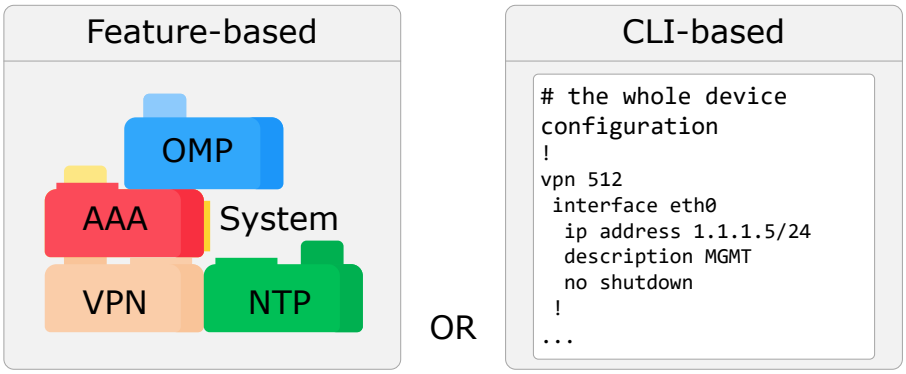


Figure 1. Feature-based vs CLI-based Template

Creating a feature-based template is comparable to assembling a template of lego blocks where each block is a different technology feature. For example, OSPF is one lego block, BGP is another lego block, AAA is another, and so on. Let's highlight the main benefit of configuring devices using feature-based templates:

- Feature templates can be reused across multiple devices. This brings greater flexibility and scale.
- It is more granular than CLI-based templates. You can modify only a specific device feature such as AAA or BGP.
- You don't need to know the device-specific syntax of different platforms. You just apply the template and vManage handles the actual configuration behind the scenes.

Configuration Variables

Network engineers know very well that network devices have many device-specific parameters that are unique per device. For example, each one has a unique name, IP addresses, interface names, router-id, and so on. To account for that, Cisco SD-WAN gives us the ability to specify three different types of values when creating feature templates:

- **Global -** When we specify a value to be Global that means that it will be applied to all devices to which the feature template is attached. For example, this will most probably be the case for the SNMP communities, Syslog servers, or the company's banner message. At a later stage, when we want to change the Banner of all nodes, we would just update the feature template value and it will update every device template that is using this feature template.
- **Device-specific -** When we know that a particular parameter will be unique for every different device, we specify a device-specific value. When we do that, the wizard will ask for a variable name. In the example in figure 2, this is [inet_if_name]. Upon applying a device template to a given device, the vManage Wizzard will ask us to provide the actual unique value for this variable.
- **Default -** The default value simply represents the factory default settings. It cannot be changed, that is why the text-box is always greyed-out and inactive. When we want to overwrite the default value, we change the value type to either Global or Device-specific.

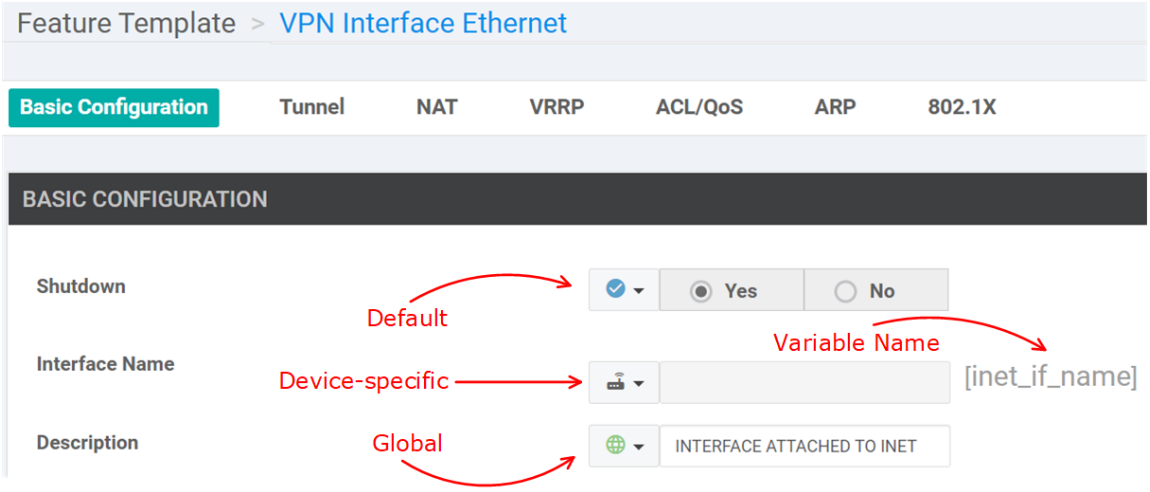


Figure 2. Feature Template Parameters

In the example shown in figure 2, you can see a Feature Template of VPN Interface Ethernet type. You can see that there is a drop-down box before each parameter that specifies the parameter type.

Device Templates

It is important to understand that a device template defines a given device's complete operational configuration. The structure of a device template is shown in figure 3. It is made of a number of feature templates depending on the specific device, role, and so on.

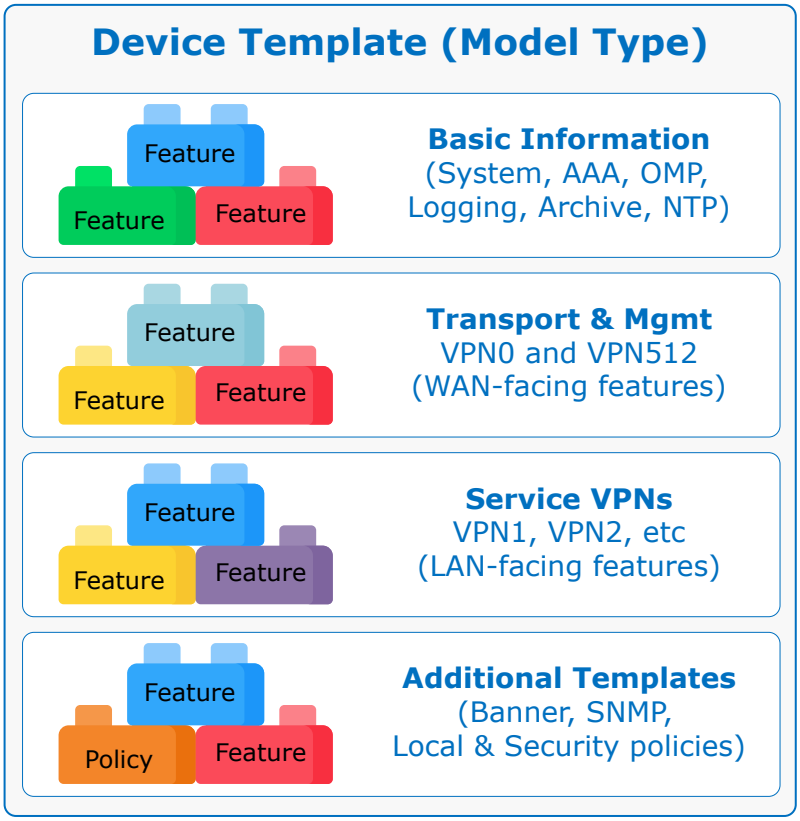


Figure 3. Device Template Structure

As you know, on the traditional Cisco networking devices, some essential features are mandatory and turned on by default (for example spanning-tree, vtp, etc). In the same way in Cisco SD-WAN, when creating a device template, some features are mandatory, indicated with an asterisk (*). That is why there are factory-default templates named Factory_Default_{Feature-Name}_Template that is applied by default in case you do not overwrite them with a more specific configuration. This can be seen in the screenshot below.

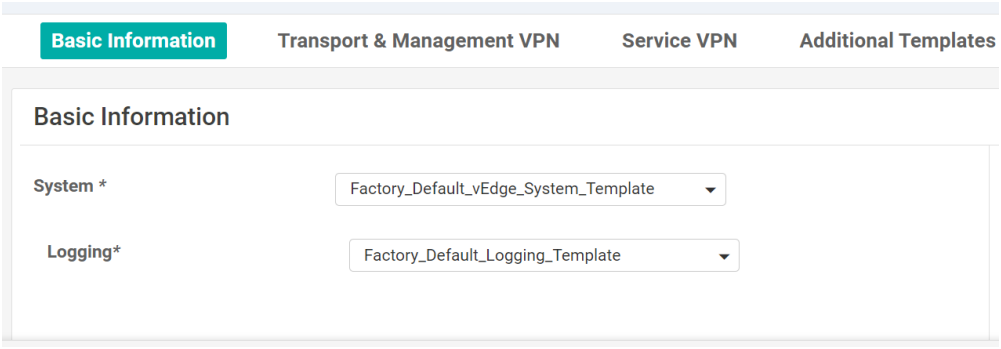


Figure 3. Factory Default Templates

Upon attaching a configuration template to a cisco SD-WAN node, vManage requires all device-specific values to be filled in. This can be done through the vManage GUI directly, or by or by using a CSV file. In the case of large-scale deployment, the CSV method allows you to configure a large number of WAN Edge nodes very quickly.

In cases where we attach a device template to a WAN edge router and it for whatever reason loses control plane connectivity to the vManage controller, the vEdge will immediately start a 5-min rollback timer. If the control-plane connectivity does not come up within that 5 minutes, the vEdge will revert back its configuration to the last-known working setup and will eventually reconnect to vManage.

Cisco SD-WAN Templates seem difficult and convoluted at first. However, when you go through our lab lessons and start playing around with them, you will see that they are very straightforward and very flexible at the same time.

vManage Mode

A Cisco SD-WAN device can be either in one of these configuration modes at any given time:

- CLI mode – a template is not attached to the device by vManage and the device's configuration can be modified locally using the cli, for example via console or SSH. This is the default mode for all Cisco SD-WAN devices.
- vManage mode – a template is attached to the device by vManage and the device's configuration can not be modified locally using the cli.

When I started playing with policies for the first time - it took me quite some time to understand what I am doing wrong and how to get around that. Going through Cisco's Viptela documentation wasn't much of a help because this particular topic was somehow convoluted. That is why I decided to create this lesson so that engineers that study Cisco SD-WAN know about this in advance. An actual screenshot of this can be seen in the following figure:

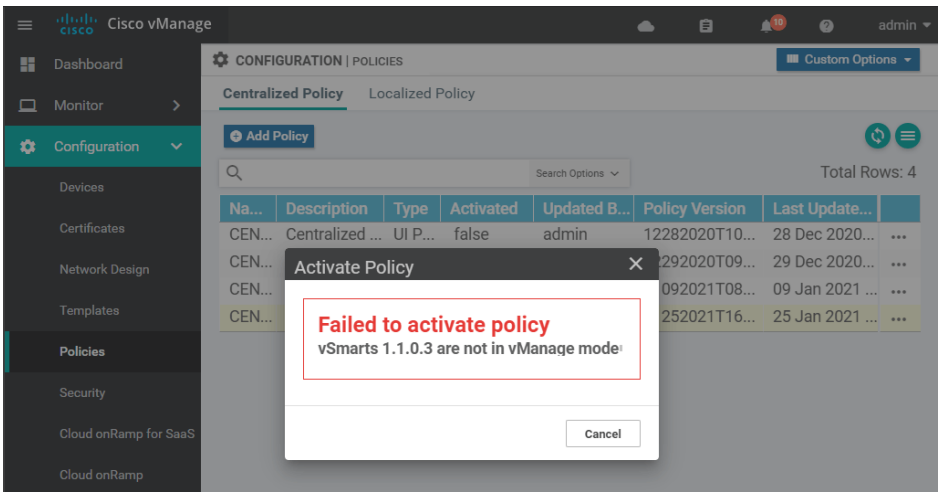


Figure 1. Failed to Active Policy

What is vManage Mode?

By default, all Cisco SD-WAN controllers are in "CLI mode". That means that they allow configuration changes done using the CLI only. However, as we have explained in our lesson for Cisco SD-WAN Policies when we activate a centralized policy through the vManage GUI, what happens behind the scenes is that the vManage is actually making configuration changes on the vSmart controller using NETCONF. But by default, like all other devices, the vSmart controller is in CLI mode (allowing config changes via cli only) and thus it does not accept NETCONF transactions from vManage. That is why the policy activation fails.

To successfully activate a policy, we should change the configuration mode of the device, that the policy will be applied to, to be in “vManaged" mode. This is done by applying a template from vManage to that device. This tells the affected node that from now on it will not be configured manually via CLI but in a centralized fashion using templates and policies from vManage.

Or alternatively on vManage under Configuration > Devices > Controllers, the configuration mode is listed in a tab as you can see in the following screenshot:

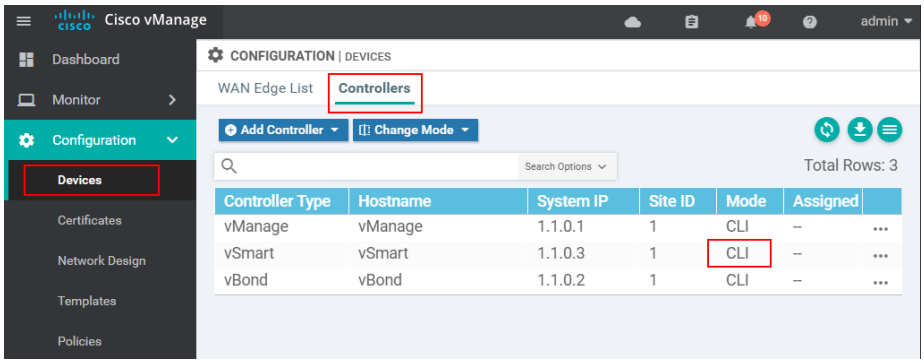


Figure 2. vSmart Configuration Mode

Applying a template to vSmart

Applying a configuration template to vSmart allows vManage to have authoritative control of vSmart’s configuration. Any type of template does the job - it does not matter whether it is a CLI or Feature template. In typical production deployments, it is very common to use CLI templates for this use case, as they are very simple and quickly made, and do not require administration beyond the initial deployment.

Practically speaking, the easiest way to change a controller to be in vManaged mode is to create a CLI template and attach it to vSmart. Let's create a CLI template by going to Configuration > Templates > Create Template > CLI.

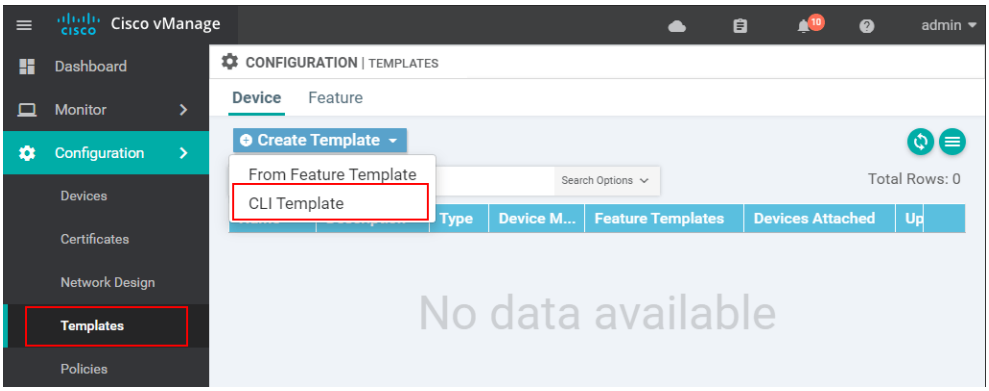


Figure 3. Creating a CLI Tempalte for vSmart

Then we select the device model (in our case vSmart) from the dropdown menu and specify the name and description for the template. At this point, we SSH to the controller, get the output of the show run command, and paste it in the CLI configuration section as shown below.

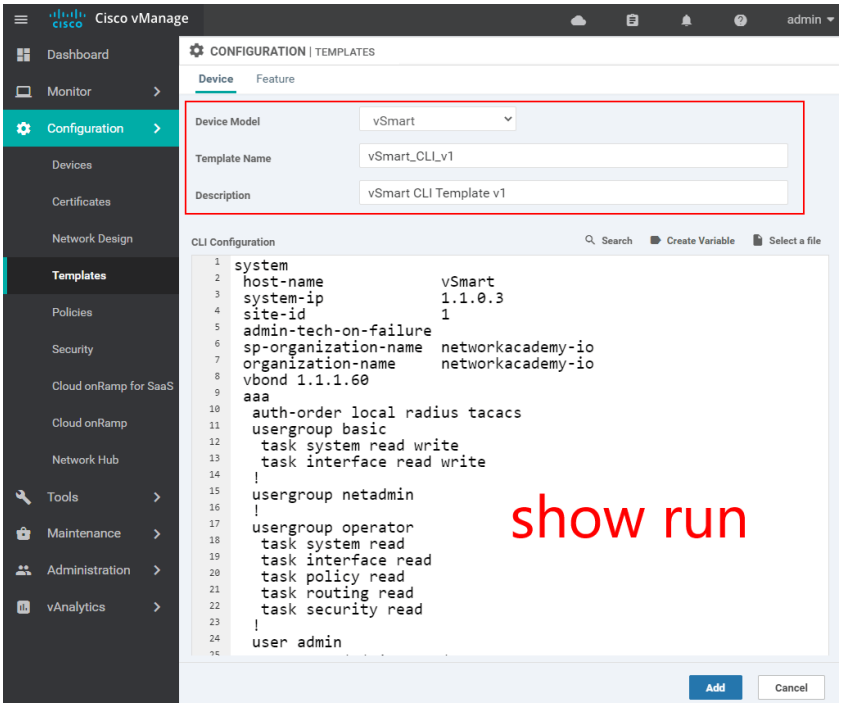


Figure 4. CLI Template Construction

Then we go to the additional options and select Attach Devices. In the next window, you are going to see all vSmart controllers that are known to vManage. You select the one, which you got the show run output from.

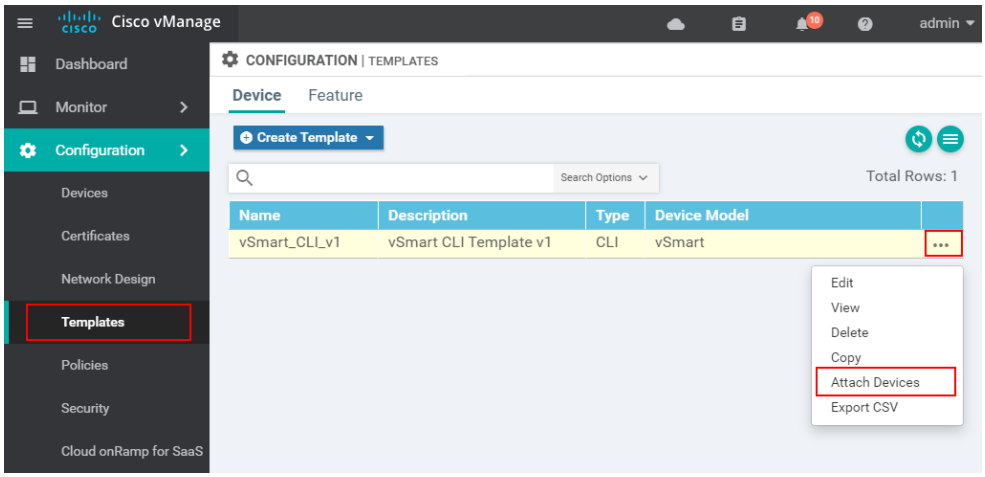


Figure 5. Attaching CLI Template to vSmart

vManaged Mode

Or alternatively, you can go to Configuration > Devices > Controllers, and check under the Mode column.

7. CISCO SD-WAN HOME LAB

Cisco SD-WAN on EVE-NG

Cisco SD-WAN is a major topic in the CCE Enterprise blueprint now. Network engineers that want to pass the lab exam should have extensive hands-on experience with the solutions. However, for people who do not have the chance to touch it at work, it is actually not easy to get access to a practice lab. In this lesson, I will show you one way to set up a fully functional Cisco SD-WAN home lab on EVE-NG that can be used to practice every topic in the exam's blueprint.

Setup EVE-NG

One thing I'd like to mention is that Cisco SD-WAN requires a lot of processing power. For a small practicing topology consisting of 1 controller of each type and 3-4 vEdges, you should give the EVE-NG VM at least 8 vCPUs and 16GB of RAM. However, if you want to make a large topology with redundant controllers and many vEdge devices, you must have a lot of computing resources at your disposal.

Cisco SD-WAN Images

You will need to have the following Cisco SD-WAN images to set up this practice lab environment on EVE-NG:

EVE-NG Image	Filename	Version
vmanage-16.3.2	viptela-vmanage-genericx86-64.ova	16.3.2
vsmart-16.3.2	viptela-smart-genericx86-64.ova	16.3.2
vbond-16.3.2	viptela-edge-genericx86-64.ova	16.3.2
vedge-17.1	viptela-edge-genericx86-64.ova	17.1

Software Images Required

Once you have the images, you need to create a folder for each one and then transfer the image to EVE-NG using a Frezzila or WinSCP tool. After you upload the images to their respective folders in EVE-NG, you need to convert the ova files to qcow2. Note that we need to create an additional virtual disk for vManage. This is done with the command highlighted in green:

Physical Topology

The physical topology that we are going to use is as follows. You should re-create it on EVE-NG. If you want to copy/paste some of the configs, make sure that you use the same interfaces when creating the topology on EVE-NG.

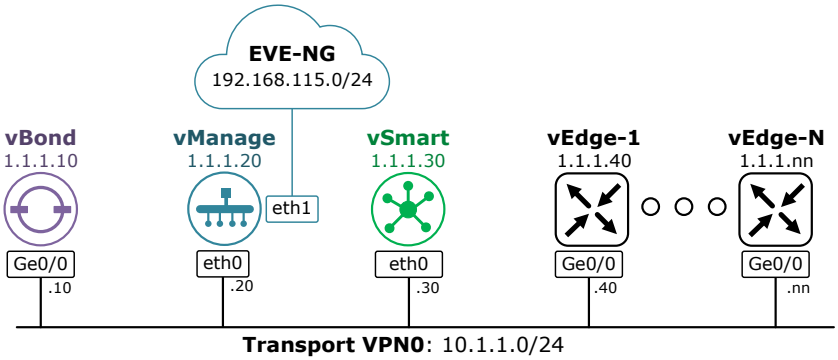


Figure 1. Physical Topology

You have to decide how many vEdge devices you are going to need and add them in the same manner as vEdge-1. The process of building the lab starts with Cisco SD-WAN version 16.3.2. Once you upgrade to the higher versions you won't be able to add more vEdges.

Default credentials for all devices are admin/admin. When vManage boots for the first time, it will ask on which storage device to install the software. Please make sure to use the virtual disk you have created in the previous step.

Bootstrap Configuration

Once all devices boot up it is time to enable basic connectivity between the controllers and all WAN edge devices. The following bootstrap snippets are the minimum required configuration in order to achieve basic connectivity.

At this point, each device should successfully ping any other in VPN0. If for whatever reason there is no reachability to one of the devices, you should not continue ahead but troubleshoot and resolve the issue.

Certificates

Cisco SD-WAN Controllers can not be brought into operation unless their identity is validated by an established chain of trust. This identity validation process is intended to ensure that only trusted devices can join the SD-WAN solution while still retaining flexibility. Each controller must have a root certificate installed and a controller certificate installed and signed by a trusted CA (Certification Authority).

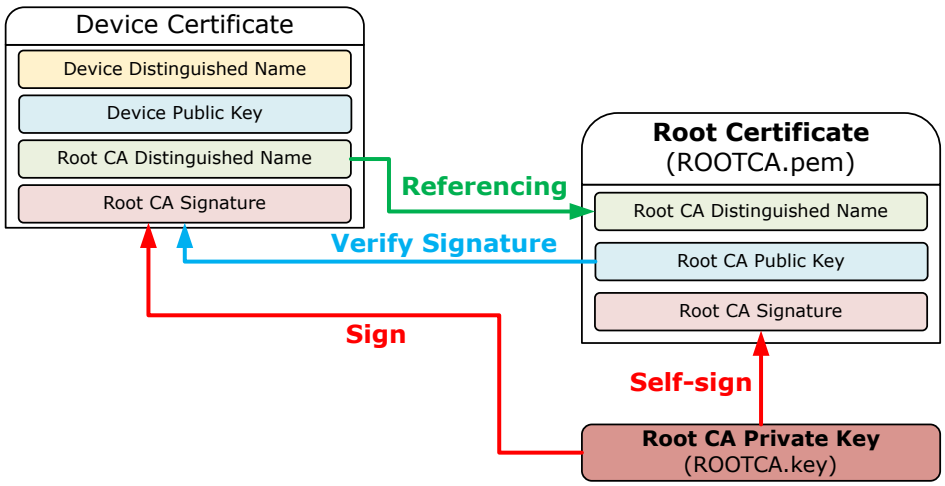
For creating this lab environment we are going to use the vBond controller as a Root CA.

vBond as Root CA

Configure vBond to act as a root of trust. The first step is to generate an RSA private key. Then we generate a ROOTCA.pem certificate and sign it with the ROOTCA.key private key that we have just created.

Once that is completed, the root certificate should be installed on all other devices. It will act as a root-of-trust for all controller certificates.

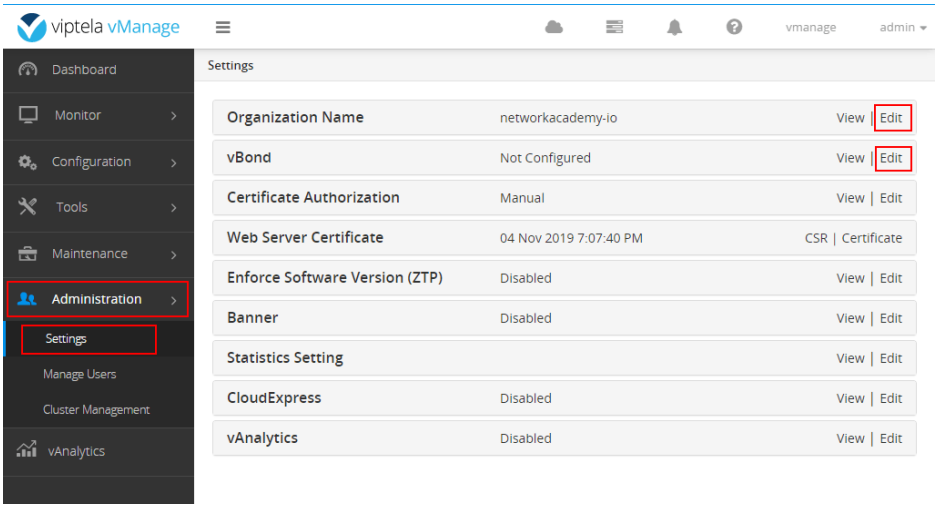
If that step is successful, that means that all devices will have their chain-of-trust pointing to the vBond controllers ROOTCA.pem.



Certificates' Relations

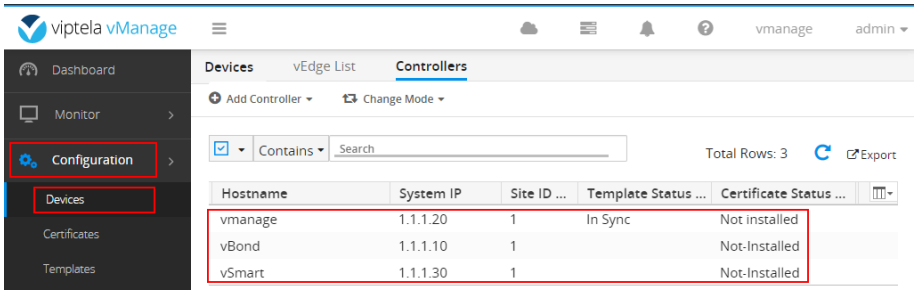
Now you need to log in to the vManage GUI interface. This is done using a web browser and entering the URL https://[vManage-VPN512-IP-address]:8443. Default credentials are admin/admin.

Once logged in, you need to go to Administration > Settings and set the Organization Name to be networkacademy-io, then edit the vBond address, and set it to 10.1.1.10. Make sure that Certificate Authorization is set to Manual as shown on the screenshot below.



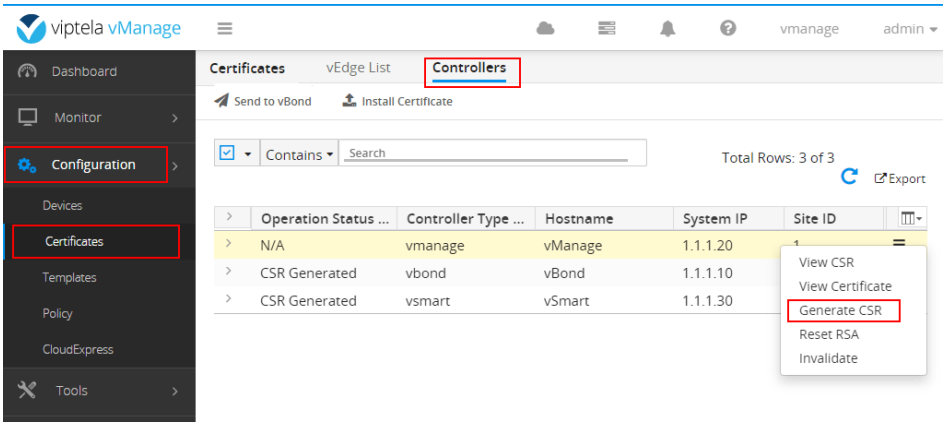
vManage GUI Initial Config

Once that is done, go to Configuration > Devices > Controllers > Add Controller and add both vBond (10.1.1.10) and vSmart (10.1.1.30) using the GUI.



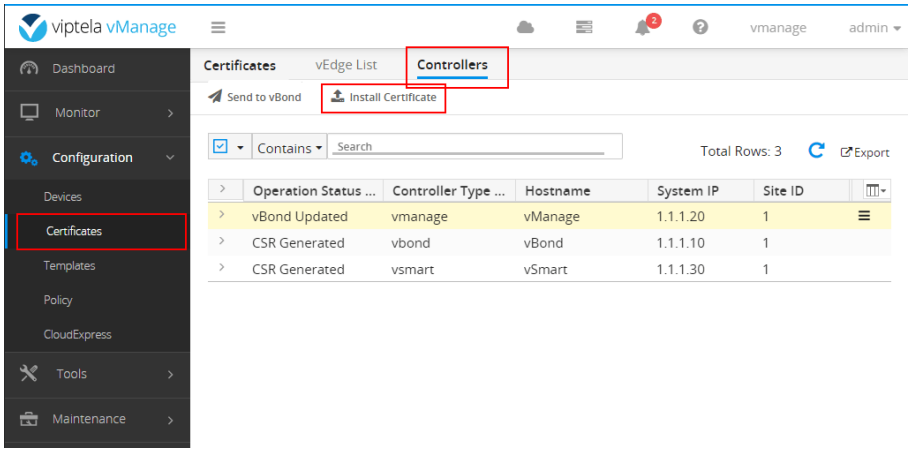
Adding the controllers via vManage GUI

Now you need to go to Configuration > Certificates > Controllers and Generate CSR for all controllers. When you are done, all should be in Status "CSR Generated".



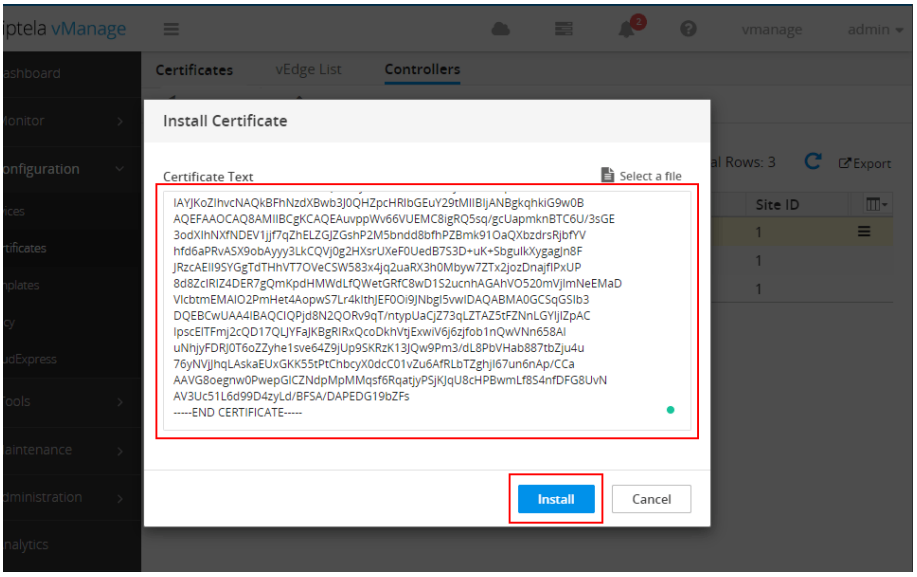
Generating CSRs

It is time to install all controller certificates via the vManage GUI. Go to Configuration > Certificates > Controllers > Install Certificate.



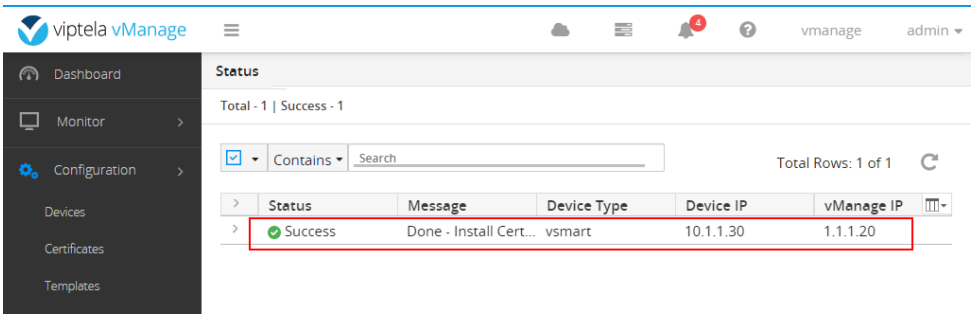
Installing the certificates using vManage GUI

Now that you have all certificates (.crt) in vBond's directory, you just cat each of them and paste the output in the Install Certificate window as shown below:



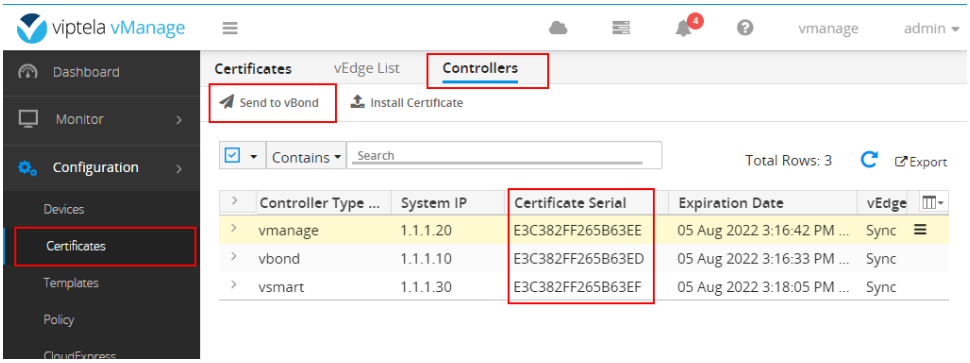
Paste the certificate into vManage

If everything is good up to this point, the certificate should install successfully.



Certificate installed successfully

You repeat this for all controllers. In the end, when you go to Configuration > Certificates > Controllers, you should see that all controllers have Certificate Serial numbers. If that is the case, you click the Send-to-vBond function to propagate this information to vBond.

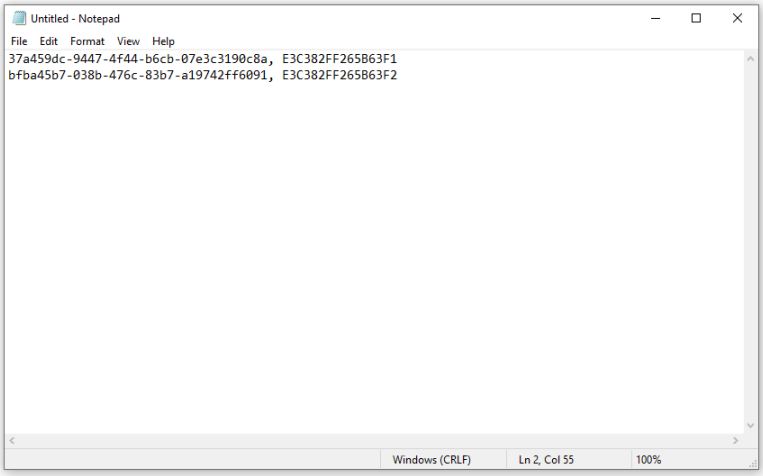


Updating vBond

At this point, all controllers should be operational with valid certificates. Now it is time to validate the vEdges.

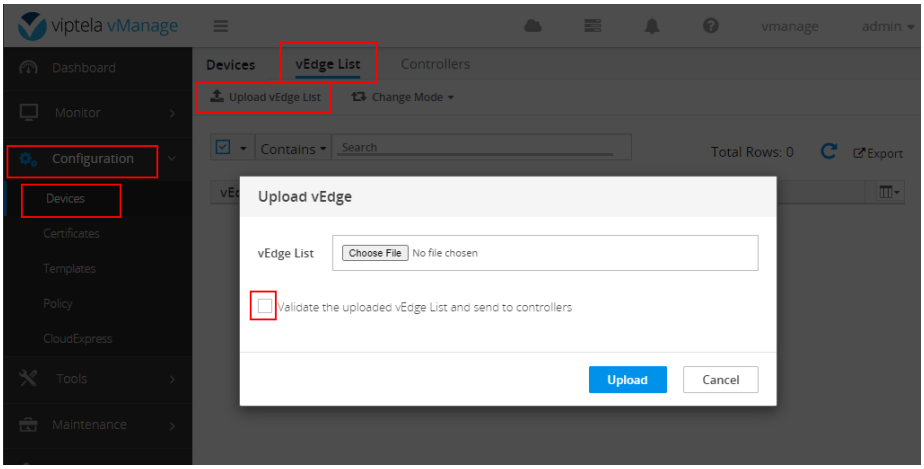
vEdges Validation

Now you need to copy the output of the show certificate serial of all vEdges in a Notepad file in the format shown below, and then save the file as vedges.csv.



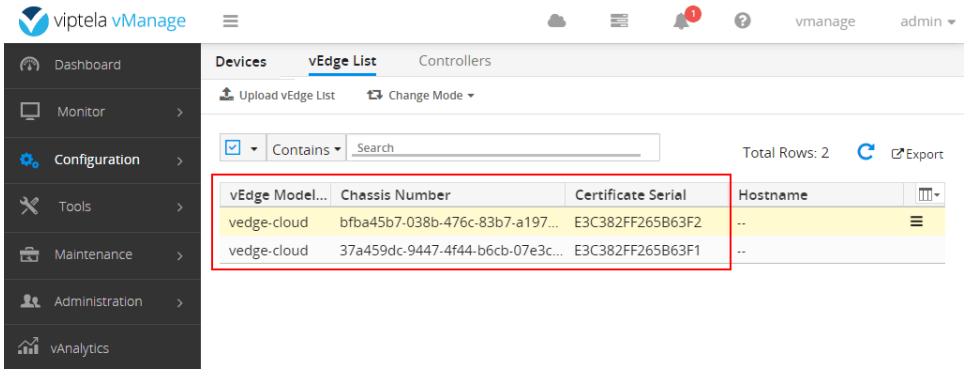
Collect all vEdges serial numbers

Then you go to Configuration > Devices > vEdge List > Upload vEdge List and select the vedges.csv file. Make sure to check the "Validate the uploaded vEdge list and send to controllers" option.



Upload vEdge-list

If the upload is successful you should see all vEdge devices having a Chassis number and Certificate Serial as shown below.



All vEdges

At this point, the validation of all devices is done. We need to bring up the control plane and upgrade to the CCIE Enterprise Infrastructure lab exam's version 18.4.4.

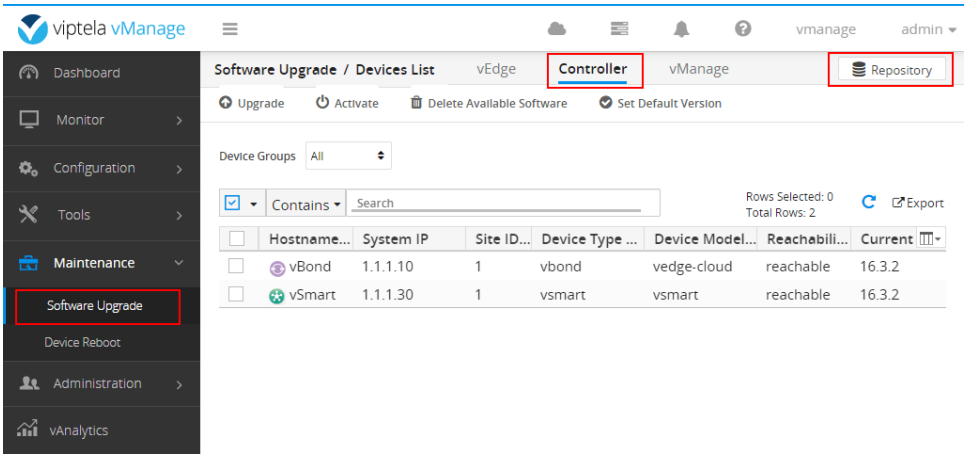
Software Upgrade

The images required for upgrading to 17.2.8 are as follow:

Device	Image Name
vSmart/vBond	viptela-18.4.4-x86_64.tar
vManage	vmanage-18.4.4-x86_64.tar

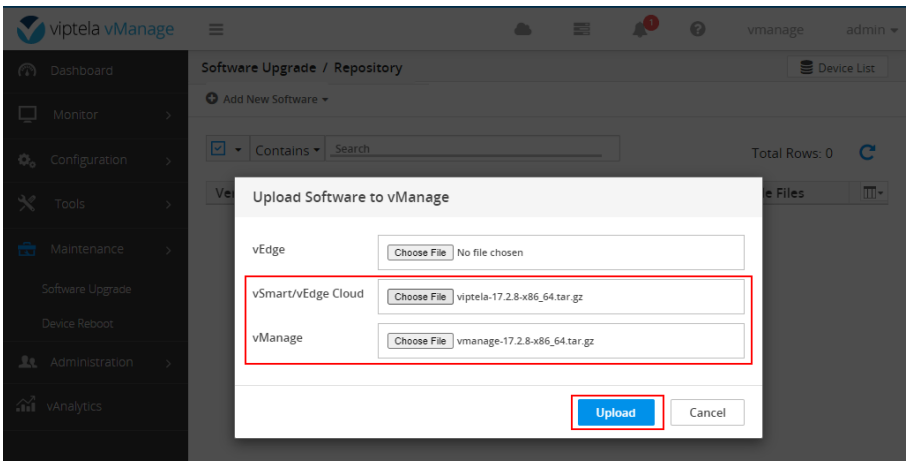
Images for upgrading to 17.2.8

The software upgrade is pretty simple and straightforward. You upload the necessary files in the Software Repository by going to Maintenance > Software Upgrade > Controller > Repository.



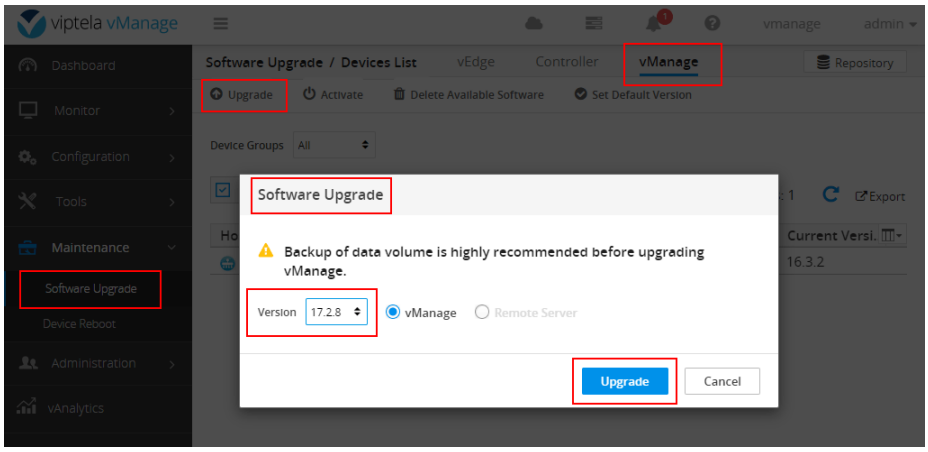
vManage Repository

In there, you select Add new software and upload the files for version 17.2.8 to vManage.



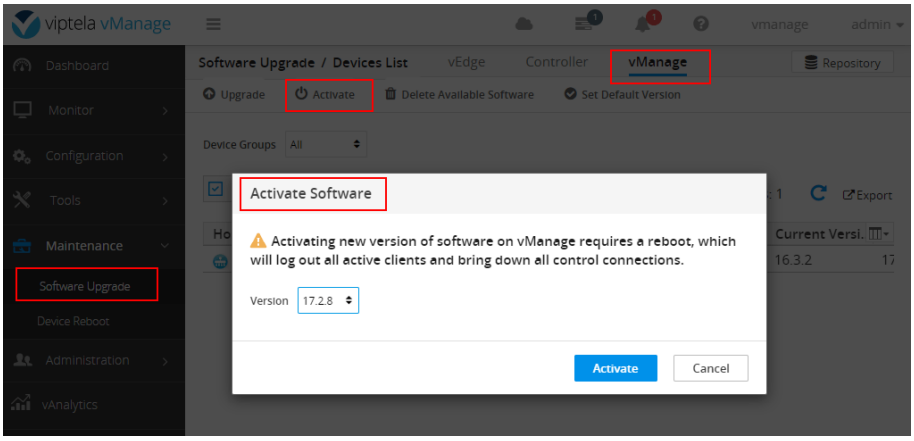
Uploading Software images to the vManage repository

Then you go to Maintenance > Software Upgrade > vManage > Upgrade, select version 17.2.8 and select Upgrade. The upgrading process is quick, the new software will be installed on vManage, but will not be activated.



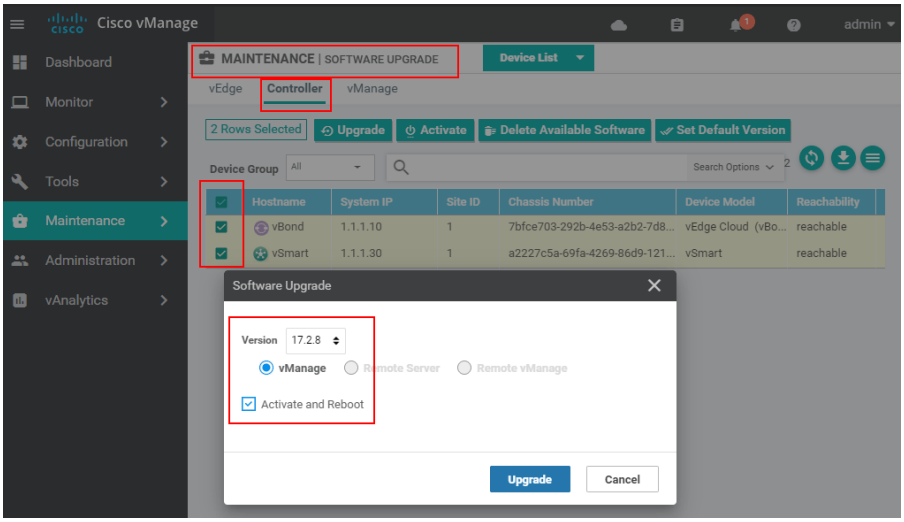
Upgrading vManage to 17.2.8

The last thing that you need to do is to Activate the new version software. This is done at Maintenance > Software Upgrade > vManage > Activate. At this point, the controller will reload. The process usually takes some time. In some cases, you may need to delete the browser's cookies to log in to the new version of the GUI once it boots with the new image.



Activating the new software image on vManage

Once you log in again, you will note that the GUI's layout is more modern. At this point, vManage is upgraded. However, the other controllers are not upgraded yet. You must go to Maintenance > Software Upgrade > Controller and select all controllers. Then select the new version 17.2.8 and check the Activate and Reboot option. Then click Upgrade. The controllers will reload.



Upgrading vBond and vManage

Once all controllers are upgrade and fully loaded, you must reload all vEdges in order to make them join the control plane.

Upgrade to 18.4.4

The images required for upgrading to 18.4.4 are the following:

Device	Image name
vSmart/vBond	viptela-18.4.4-x86_64.tar
vManage	vmanage-18.4.4-x86_64.tar

Images for version 18.4.4

The process of upgrading to 18.4.4 is absolutely the same as the one to version 17.2.8:

- Upload the images to the Software Repository
- Upgrade vManage
- Activate vManage
- Upgrade and activate the controllers

So you just go ahead and perform it and that is it - you have a Cisco SD-WAN home lab on EVE-NG that is running on the same version as in the CCIE Enterprise Infrastructure lab exam.

At this point, you can modify the topology in any way you wish to practice. You can add other network devices like IOL switches, routers or firewalls. The only drawback is that you cannot add additional vEdge devices.

Happy practicing ;)

Credits

I found out about this method of setting up a Cisco SD-WAN practicing home lab in a blog post at codingpackets.com. Kudos to the blog post writer! The full link to the article is below:

https://codingpackets.com/blog/viptela-control-plane-setup/

Packet loss, Latency and Jitter on EVE-NG

Emulating WAN properties on EVE-NG

One of the most important improvements of Cisco SD-WAN over the Traditional WAN architecture is the ability to reroute application traffic around WAN performance degradations and brownout conditions such as packet loss, latency, and jitter. The solution has multiple features related to WAN link brownouts like Application-aware routing (AAR), Centralized Data Policies, Forwarding Error Correction (FEC), Packet Duplication, and many more. However, many network engineers do not know how to practice these SD-WAN capabilities at home using any of the homelab emulators such as GNS3 or EVE-NG. In this lesson, we are going to show how we can emulate wide-area network (WAN) properties on EVE-NG using an open-source Linux tool called NETem.

What is NETem?

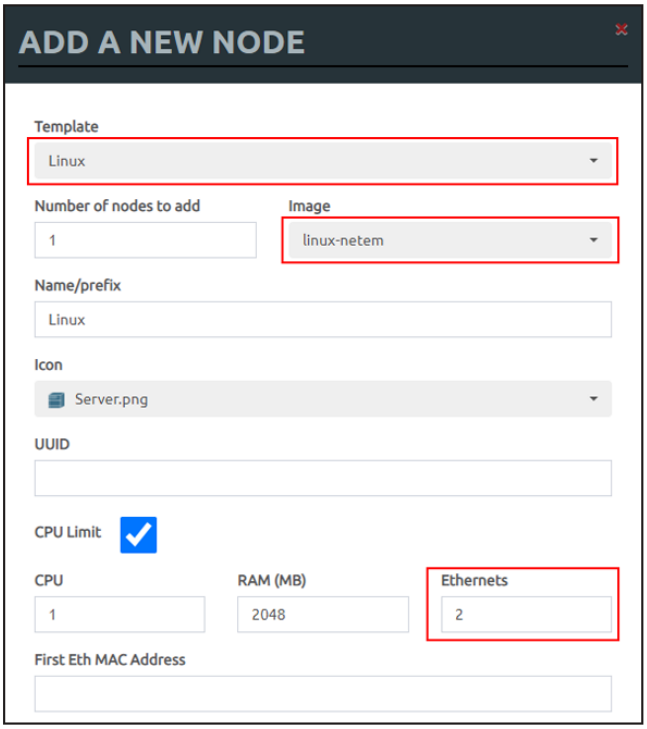
NETem is a Network emulation tool that provides functionality for testing network protocols by emulating the properties of wide-area networks. The latest version of the tool supports the emulation of latency, packet loss, packet duplication, bursts, congestion, and packet re-ordering.

Installing NETem

If the virtual machine has an Internet connection and enough available disk space, the installation should be successful in no time.

Adding NETem node on EVE-NG

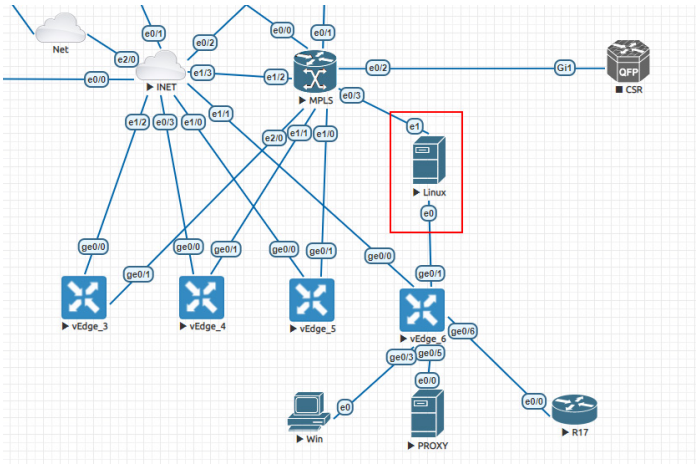
Once the addon is installed using the commands shown above, it will be automatically placed as a Linux node in EVE-NG. Therefore, to use it on a link between two network devices, we need to add a new node on EVE-NG. We select linux and then select the linux-netem image as shown in the screenshot below:



Adding NETem node on EVE-NG

Notice that by default, the node is added with only one ethernet interface. However, to put it inline between two network devices, we are going to need at least two Ethernet interfaces, one to connect to device A and one to device B.

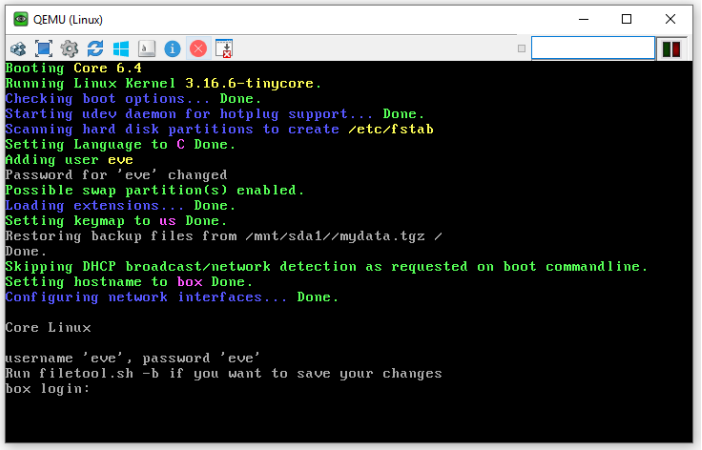
Once the node is added, we just connect it inline between two network devices as shown in the screenshot below. It acts as a layer 2 device, so it is layer 3 transparent and does not need an IP address information to any of the interfaces.



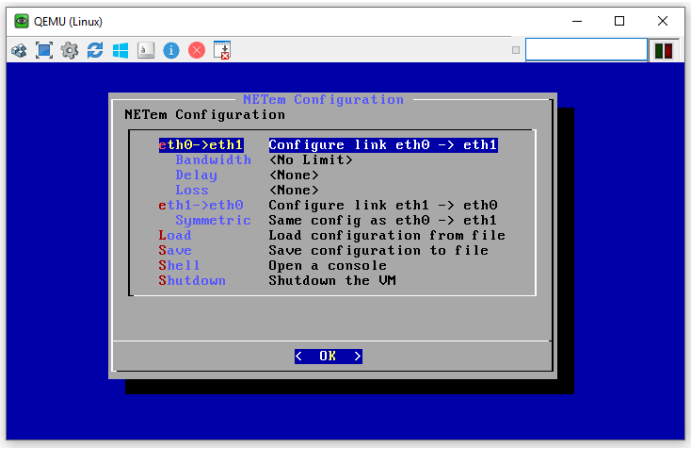
NETem node on EVE-NG

Configuring NETem

Once the linux-netem node boots up, we connect to it with VNC and log on with user "eve" and password "eve".

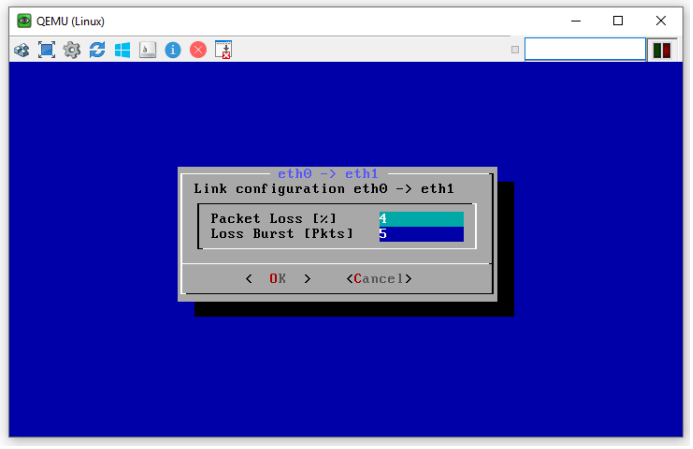


Once logged in, there is a very basic graphical interface that allows us to configure the desired values for bandwidth, delay, jitter, packet loss, and packet bursts.



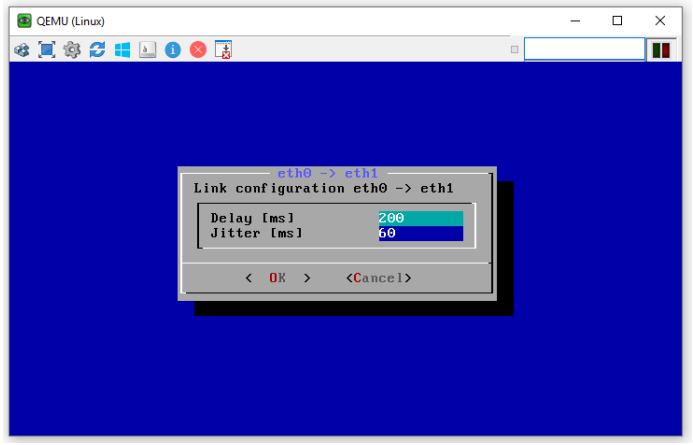
NETem initial screen

Entering each configuration hierarchy, we specify the required values. The screenshot below shows how we configure packet loss and loss burst.



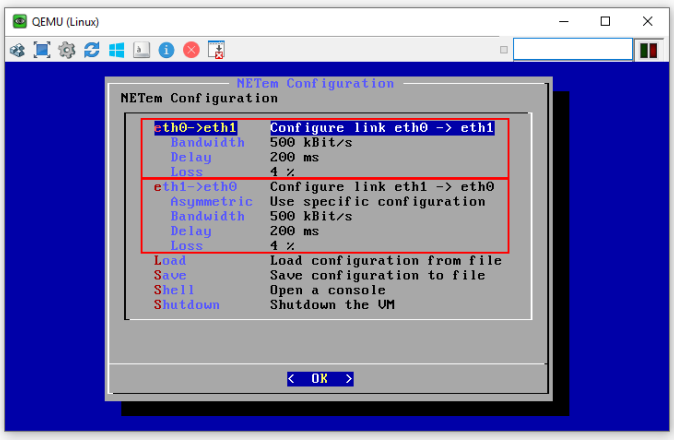
Configuring Packet Loss on Netem

And the next screenshot shows how we configure delay and jitter.



Configuring delay and jitter on NETem

Notice that we configure each traffic direction separately. Also, notice that the configured values shown in the screenshot below are for each traffic direction. Therefore, if you run traffic through a netem node configured as shown in the screenshot below, you will get approximately 8% of packet loss and 400ms of round trip times.



Configuration summary

You can see that there is approx 8% packet loss and the average round trip times are as configured in the netem node.

Key takeaways

In this lesson, we have shown an open source add-on that can be used to emulate packet loss, latency, and jitter in EVE-NG. The add-on is a very useful tool when it comes to practicing SD-WAN capabilities such as Application-aware Routing, SLA-based routing, Forwarding Error Correction, Packet Duplication, and so on.

8. CENTRALIZED CONTROL POLICIES

What is a Centralized Control Policy?

A centralized control policy is a policy that manipulates the route and tloc information that is exchanged between the vSmart controllers and the vEdge devices in the Cisco SD-WAN overlay fabric. It can influence the overlay topology of IPsec tunnels and the routing paths through the fabric.

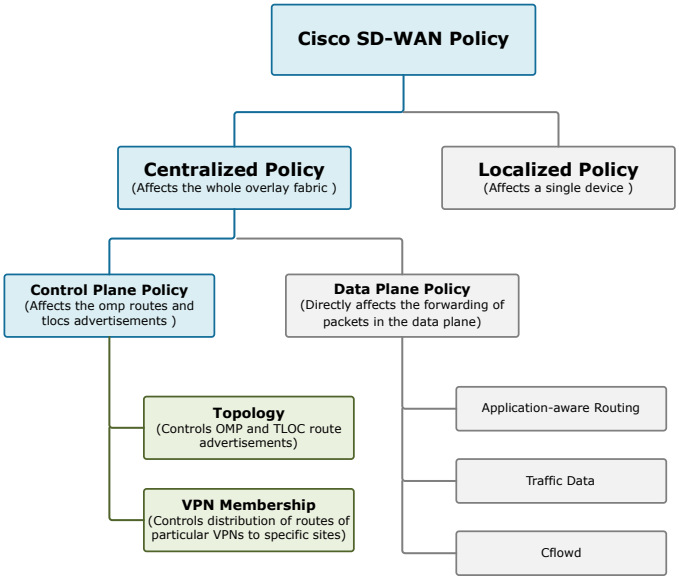


Figure 1. Types of Centralized Policies

In Cisco SD-WAN, there are two types of Centralized Control Policies that fulfill different objectives:

Topology - Topology policies control the route information such as omp, tloc, and service routes that are being redistributed to a list of sites. As the name implies, they are typically used for limiting the number of overlay tunnels between sites and controlling the overlay topology.

VPN Membership - VPN Membership policies are used to control the distribution of routing information for specific VPNs to a list of sites. A typical use-case is for creating guest networks that have Internet access but site-to-site communication is restricted.

To fully understand why do we use centralized control policies, you need to have a good understanding of the control plane in the Cisco SD-WAN solution. Let's recall how it works.

Control Plane Overview

The Cisco SD-WAN Viptela solution is a Wide Area Network (WAN) overlay architecture that applies the principles of Software-Defined Networking (SDN) where the control plane of the network is completely separated and isolated from the data plane. In the Cisco SD-WAN, the vSmart controller represents the control plane of the overlay fabric and is effectively the centralized route engine that manages a network-wide routing table. It receives three types of route information from all WAN edge routers and builds a centralized route table. Then it redistributes this route information towards all vEdge devices.

IMPORTANT It is very important to understand that WAN edge devices do not exchange any kind of control plane information, such as routes and tlocs, to one another.

Figure 1 visualizes this concept. If vEdge-1 sends an OMP Update to the controllers, vSmart processes this message and if needed sends out an update to all other WAN edge devices. However, vEdge-1 would never send an OMP update directly to another vEdge router directly.

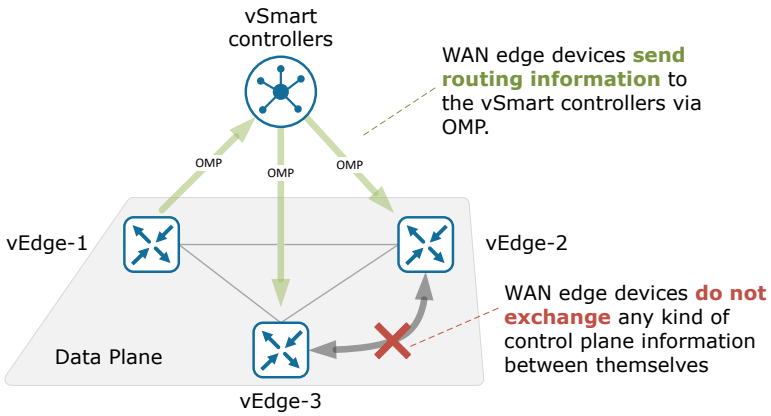


Figure 2. Cisco SD-WAN Exchange of Control Plane information

Route Types

There are three types of OMP routes that the vSmart controllers learn from the WAN edge devices:

OMP routes - These routes represent prefix information that WAN edge devices learn from their local network. For example, these could be connected networks, static routes, or OSPF process running onsite redistributed into OMP. The OMP routes could be displayed using the show omp command.

TLOC routes - These routes simply carry TLOC information from vEdges to the vSmart controllers. They could be displayed using the show omp tlocs command.

Service routes - These routes represent network services, such as firewalls and IPS, that are running on the local-site network to which the vEdge router is connected. They could be displayed using the show omp services command.

Centralized Control Policy

Network engineers that do not have any prior experience with SD-WAN solutions at first struggle to realize where the centralized policy is applied. That is why we tried to visualize where exactly the "magic happens". If you look at the diagram shown in figure 3, the red arrows represent a policy applied in the direction of the arrow. You can clearly see that an inbound control policy affects the OMP route information coming from vEdge routers before it is stored in the vSmart controllers' database. And an outbound one affects the OMP route advertisements from the controllers toward the WAN edge devices.

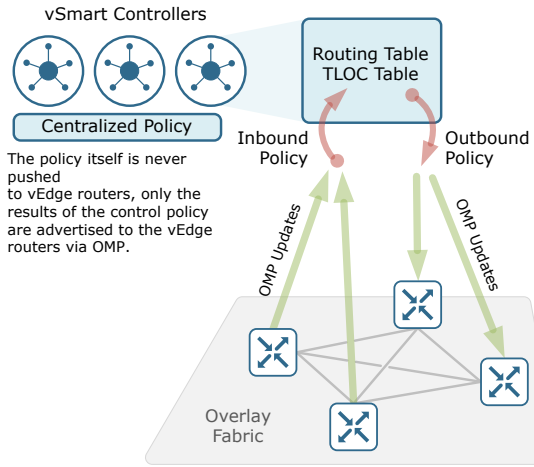


Figure 3. Cisco SD-WAN Centralized Control Policy Directions

Default vSmart Behavior

As we have already learned, there is no centralized control policy configured on the vSmart controller by default. Therefore, by default, the controller acts in the following way:

- Each vEdge device sends all site-local prefixes, tlocs, and service routes toward the controller using all established DTLS control connections.
- The vSmart controller accepts all incoming OMP routes (omp, tloc, or service) and stores them in the respective route tables per VPN.
- The vSmart then redistributes all learned routes to all WAN edge devices. This results in a full-mesh overlay fabric and full IP reachability between all nodes.
- Each vEdge device continually sends route updates.
- The vSmart updates its routing table based on each update and advertises any routing changes to all edge devices.

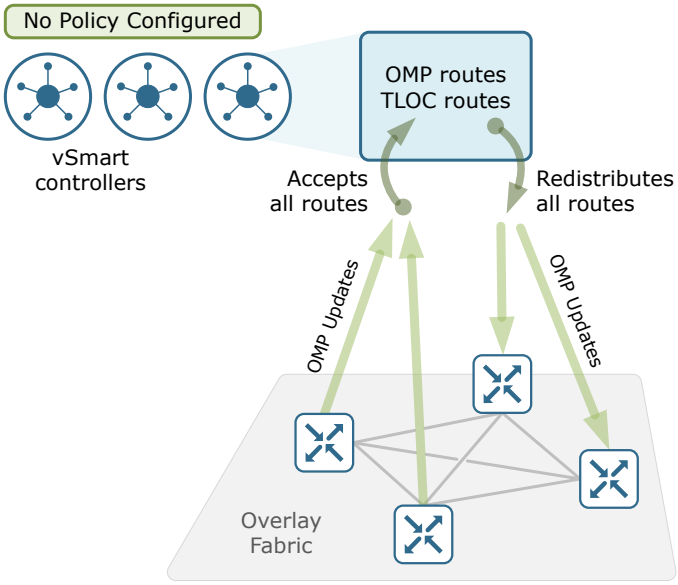


Figure 4. Cisco SD-WAN Overlay behavior without Centralized Policy

vSmart Behaviour with Centralized Policy

As we said, the default behavior of the controller is to advertise all routes (omp, tloc, and service) which results in a full mesh overlay fabric and any-to-any IP reachability. However, in most cases, this is not the desired network outcome that companies require. Therefore, in most scenarios, the network topology should be customized and the IP reachability must follow the company's policy.

When we want to control the route information that is stored in the controllers' route tables or the route information that is advertised to vEdges, we provision a Centralized Control Policy. When such a policy is applied, the behavior of the controllers change as follow:

- When a Centralized Control Policy is applied in an inbound direction, it filters or modifies the route information that is coming from vEdges before it is placed in the controller's routing table.
- When a Centralized Control Policy is applied in an outbound direction, it filters or modifies the route information that is advertised to vEdges.

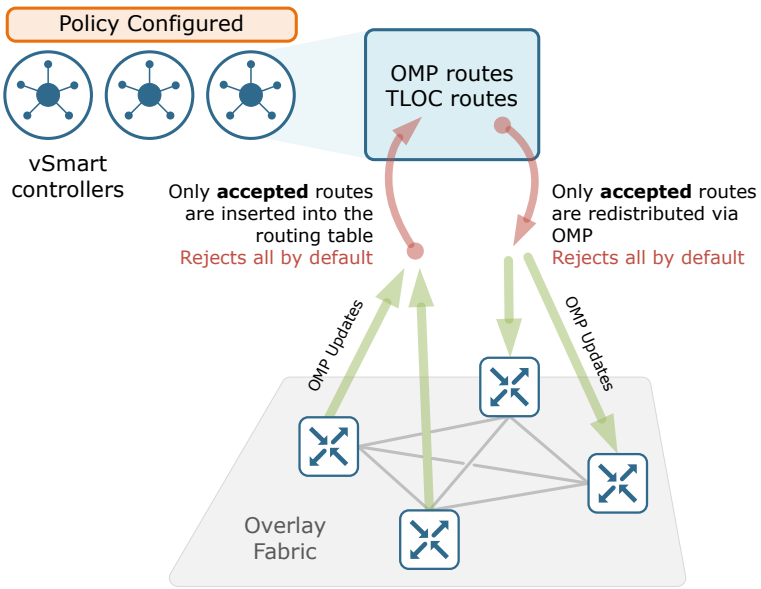


Figure 5. Cisco SD-WAN Overlay behavior with Centralized Policy

Policy Components

Defining a new policy through the vManage Policy Wizard is a three steps process:

- **Step 1 -** Create Groups of Interest. In Cisco IOS, there are a few types of lists that can identify specific groups of interest. Network engineers are pretty familiar with access-lists, prefix-lists, as-path lists, and so on. In Cisco SD-WAN there are even more and different types of lists such as site-lists, VPN lists, TLOC lists, and so on. However, the idea is pretty much the same, you just identify specific values of interest.
- **Step 2 -** Configure Topology and VPN Membership. At this step, we define our Control and VPN membership policies that are used to manipulate the propagation of routing information in the control plane such as OMP and Transport Locator (TLOC) routes. A control policy is defined as a sequence of match-action statements. Typically a list created at step 1 is matched and then a specific set of actions is taken upon it.
- **Step 3 -** Apply Policies to Sites and VPNs. Cisco SD-WAN centralized policies are always applied to a site-list that identifies one or more site-ids. If the policy is applied in the inbound direction, it affects the route updates coming from the vEdge routers that have these site-ids. If the policy is applied in the outbound direction, the policy affects the OMP route advertisements from the vSmart controller to the vEdge devices with the specified site-ids.

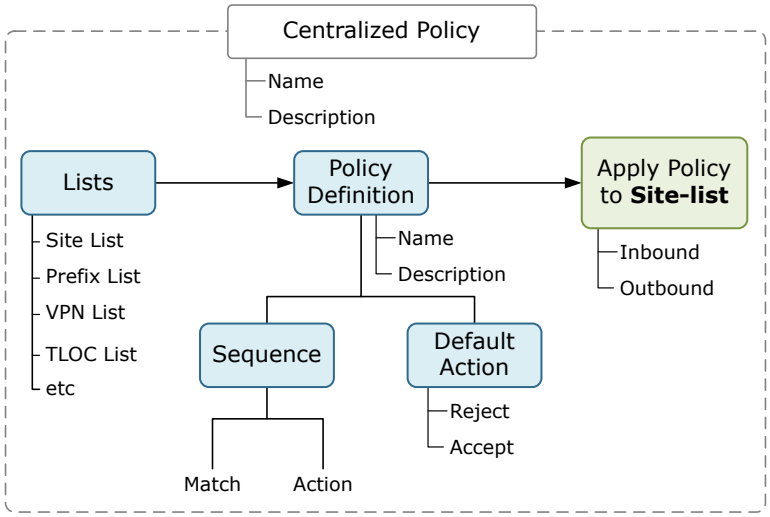


Figure 6. Cisco SD-WAN Centralized Policy Components

Activating a Centralized Control Policy

In Cisco SD-WAN, vManage is the single pane of glass for administration and operation of the overlay fabric. This is the centralized tool where all configuration, management, and troubleshooting takes place. Of course, this includes the configuration of Centralized policies.

When a Centralized Policy is defined and activated in vManage, vManage pushes that policy as a NETCONF transaction to the vSmart controllers. The policy itself is never pushed to the WAN edge devices, only the results of the policy are advertised by the vSmart controllers via OMP to the overlay. It is also important to mention that the vSmart controller only keeps the last configured policy in its configuration database.

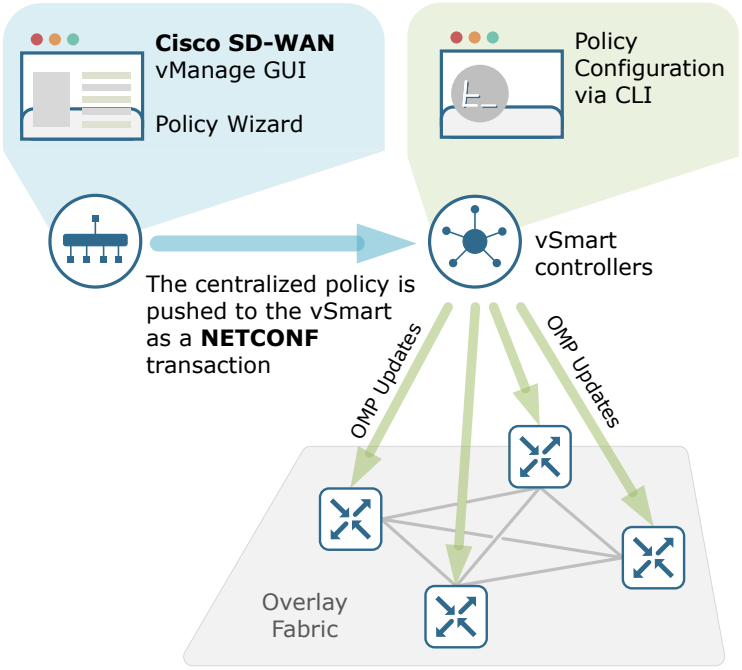


Figure 7. Ways to configure a Centralized Policy

In a typical production deployment, there are at least two or three vSmart controllers that provide a control plane redundancy. In this scenario, it is the responsibility of vManage to ensure that the centralized policy configuration is synchronized across all controllers. When a new policy is being applied, if for whatever reason it cannot be pushed successfully to one of the controllers, vManage automatically rolls back to all changes to the last synchronized configuration.

Inbound vs Outbound Control Policy

Cisco WAN Edge routers periodically send and receive OMP updates with the vSmart controllers. These OMP updates contain vRoutes, TLOC, or Service routes. When a vSmart controller receives an OMP route from a vEdge, it performs the OMP best-path algorithm and updates its own routing table. The best routes are then re-advertised to all other WAN edge routers.

IMPORTANT vSmart advertises only the best routes according to the OMP best-path algorithm.

A Control policy examines the OMP updates and can modify the attributes in an update that matches the policy. Control policies are always applied directionally to a site list.

We have seen in the previous lab lessons that Control Policies can be applied in an inbound or outbound direction. In many cases, network requirements can be fulfilled by applying a control policy in either direction. However, there is a huge difference in the outcome of an inbound control policy versus an outbound one.

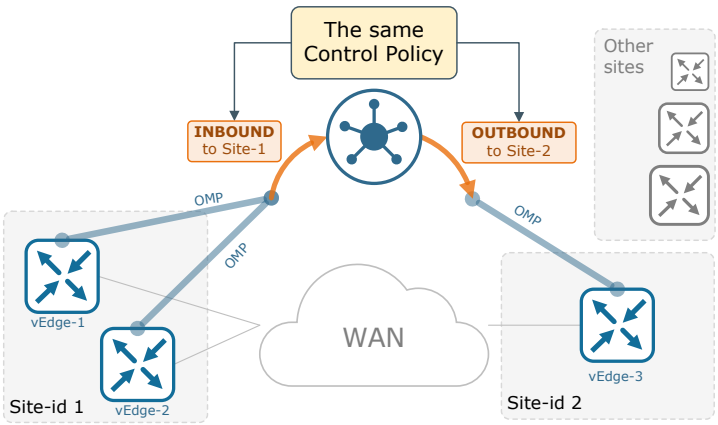


Figure 1. The same control policy in different directions

In this lesson, we are going to apply a policy in the inbound direction and see the results. Then we are going to apply the same policy in the outbound direction and compare the differences.

Outbound Policy

Figure 2 shows a control policy named PREFERENCE that is applied to Site-list SITE-2 in an outbound direction. Let's break down the policy construct into bullet points and analyze every aspect of it.

- The direction of the policy is always from the perspective of the vSmart controller. Outbound means that the policy matches and modifies attributes in the OMP advertisements from vSmart to vEdges.
- The policy is applied to site-list SITE-2 means that only OMP advertisements send to WAN edge devices with Site-IDs listed in the SITE-2 list are processed against the policy.
- In the policy itself, sequence 1 match route means that this sequence matches and modifies only vRoutes (and not TLOCs or Service routes).
- The action accept means that the vRoutes that are matched in the statement will be sent out to site-2.
- Set preference 90 means that the OMP Preference attribute of the vRoutes that are matched will be changed to 90 (default is 0).
- Note that the default action is accept (it is reject by default). We do not intend to filter out any route or tlocs but to only modify route attributes.

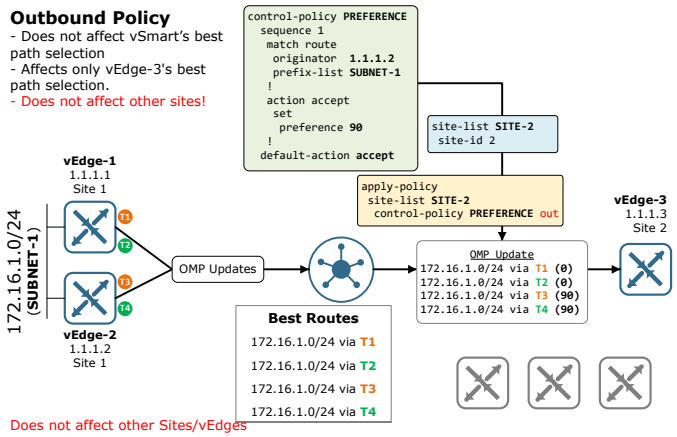


Figure 2. Outbound Centralized Control Policy

In simple words, the policy affects the OMP advertisement to vEdge-3 (site-list SITE-2). By default, vSmart advertises the first four equal-cost best routes for a prefix. In this case, for prefix 172.16.1.0/24, from the perspective of vSmart, there are four equal-cost best routes (via T1, T2, T3, and T4). However, before vSmart sends the advertisement to vEdge-3, the policy matches the routes that originated from vEdge-2 (1.1.1.2) and sets the OMP Preference to 90 for these routes (default is 0, higher is better).

When vEdge-3 receives this OMP advertisement, it performs the OMP Best-path selection algorithm and selects the omp routes with Preference 90 as best. The first stop and most important place to check the overlay routing in Cisco SD-WAN is always on the vSmart controller because it is the only authorized device that can redistribute routing information between WAN edge routers.

You can see that vSmart has four equal-cost routes for 172.16.1.0/24 via TLOCs 1,2,3 and 4. As expected, the OMP preference of all routes is 0 because the policy affect the outbound OMP advertisements from the perspective of the controller and do not affects the controller's RIB.

We can see that when the control policy shown in figure 2 is applied in the outbound direction to a site list (in our case to vEdge-3), it only affects the WAN edge routers listed in the applied site list. It does not affect the OMP best-path selection algorithm on vSmart nor the one on any other vEdge routers in different sites.

Let's now see what will be the results if we apply the same policy in the inbound direction.

Inbound Control Policy

Figure 3 shows a control policy named PREFERENCE that is applied in an inbound direction to Site-list SITE-1. Let's break down the policy construct into bullet points and analyze every aspect of it.

- The direction of the policy is always from the perspective of the vSmart controller. Inbound means that the policy matches and modifies attributes in OMP updates before the information enters the OMP RIB (routing information base) of the controller.
- The policy is applied to site-list SITE-1 means that only OMP updates with Site-IDs listed in the SITE-1 list are processed against the control-policy.
- In the policy itself, sequence 1 match route means that this sequence matches and modifies only vRoutes (and not TLOCs or Service routes)
- The action accept means that the vRoutes that are matched in the statement will be inserted in the RIB of the controller
- Set preference 90 means that the OMP Preference attribute of the vRoutes that are matched will be changed to 90 (default is 0) before the vRoutes are inserted in the RIB of the controller.

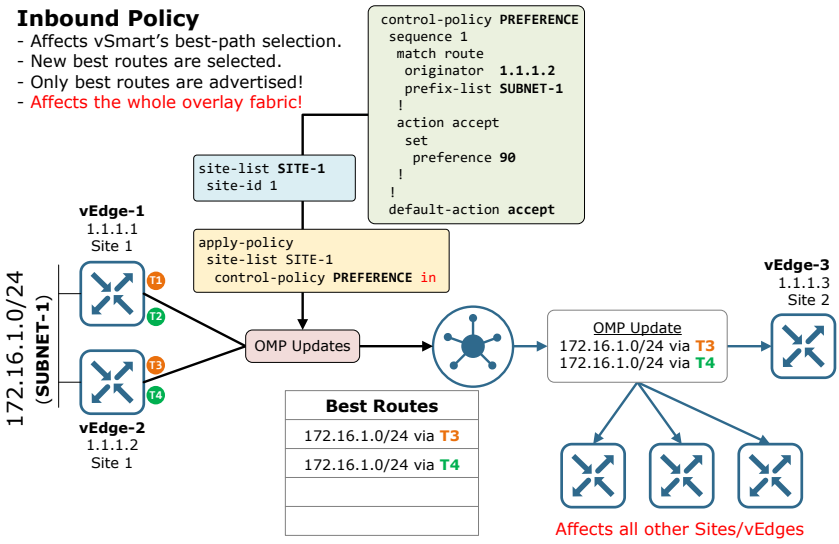


Figure 3. Inbound Centralized Control Policy

In simple words, the policy matches the omp advertisements for 172.16.1.0/24 (SUBNET-1) coming from vEdge-2 (1.1.1.2) to vSmart and sets the OMP Preference to 90 to these vRoutes. This happens before the OMP routing information is inserted in the controller's RIB. Therefore, when vSmart runs the OMP best path algorithm against the vRoutes for 172.16.1.0/24, it will select the vroutes via vEdge-2's TLOCs as best because they have a higher OMP Preference (90) than the vroutes via vEdge-1's TLOCs (0).

Recall, that vSmart controllers only advertise the best equal-cost routes! Therefore, in our example, vSmart will re-advertise only the vroutes via vEdge-2 to the overlay fabric.

Key Takeaways

Let's summarize what we have learned in this lesson using two diagrams.

Outbound policies are used to influence the OMP routing information on specific WAN edge devices listed in the applied site list as shown in figure 4.

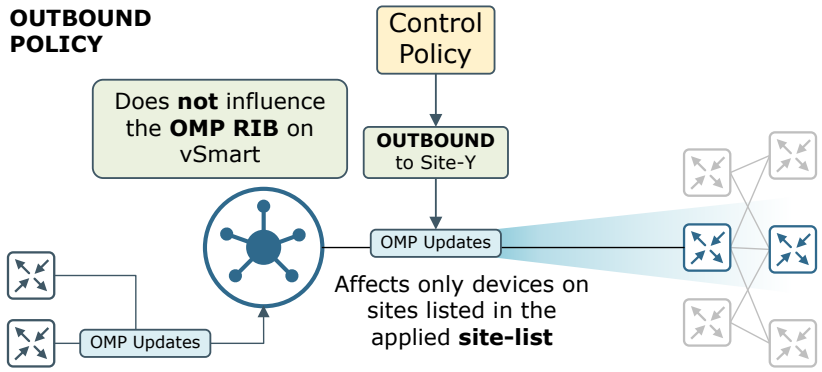


Figure 4. Outbound Policy Summary

Inbound policies are used to influence the OMP routing information on vSmart controllers which subsequently affects the OMP routing information of all WAN edge devices as shown in figure 5.

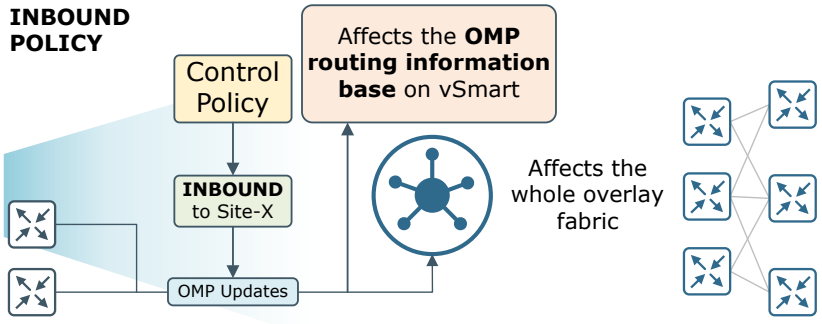


Figure 5. Inbound Policy Summary

LAB 1: Hub-and-Spoke - Restricting spoke-to-spoke tunnels

The Default Overlay Fabric

As we have already discussed throughout this course - the default behavior of the Cisco SD-WAN overlay fabric is to build a full mesh of IPsec tunnels between all WAN edge routers with different site-ids. Let's visualize this with the example shown in figure 1. If we have six WAN edge routers connected to a single WAN transport as shown on the left, the default overlay outcome with no policies applied results in 15 IPsec tunnels as shown on the right side. You can easily calculate how many tunnels will be established with the full mesh formula n*(n-1)/2. In case the company has 100 branches, there will be 4950 overlay tunnels.

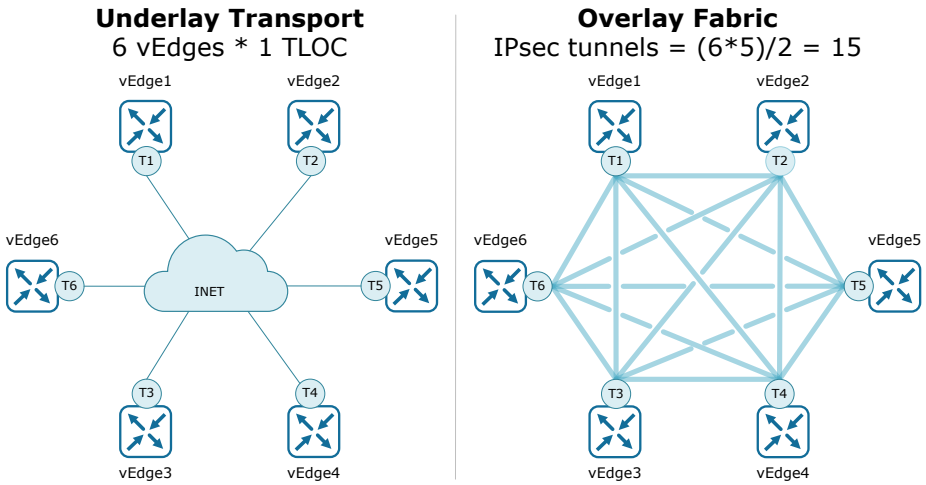


Figure 1. Cisco SD-WAN Default Overlay Fabric with one transport

In a real-world example, it is more likely that each vEdge device is connected with at least two underlay transports typically one public internet circuit and one private MPLS one. In this case, there will be twice as many overlay connections as shown in figure 2. In this scenario, both WAN transports are completely independent, meaning that blue TLOCs cannot reach green TLOCs.

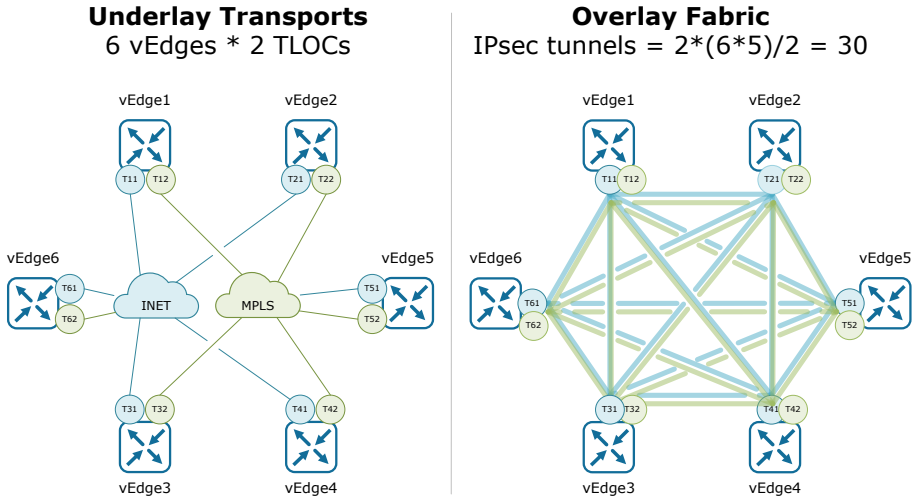


Figure 2. Cisco SD-WAN Default Overlay Fabric with two transports

At this point, it is important to understand the behavior of the WAN edge routers and what is a TLOC. A TLOC (Transport Locator) is a data plane attachment point identified by a 3 values tuple - (System IP, Color, Encap). TLOCs do not have a straight one to one analog in traditional networking, but the closest simplest explanation is that they are just tunnel endpoints. Therefore, if a vEdge device receives a TLOC (tunnel attachment point) from the vSmart controllers, it attempts to establish a tunnel from each of its own TLOCs to this new TLOC. If the tunnel is successfully built, it establishes a BFD session over the tunnel and starts keeping track of the state and the characteristics of it.

IMPORTANT The number of established IPsec tunnels and the arbitrary network topology is controlled by the advertisements of Transport Locators (TLOCs) by the vSmart controllers. By default, all TLOCs are advertised to all vEdge routers, resulting in a full mesh overlay fabric. To create a custom network topology, you must filter which TLOCs are advertised to which specific WAN Edge devices.

The next example is not very common but illustrates well the overlay fabric creation. If we have four WAN edge routers attached to two underlay transports as shown on the left in figure 3 and there is full reachability between both clouds, in this case, IPsec tunnels are established between different color TLOCs resulting in the overlay shown on the right. Even with only four sites, there are 24 tunnels, because blue TLOCs can reach green TLOCs, and tunnels between blue and green endpoint are also built. Therefore, if the company has 100 sites, this results in 200*199/2-100 = 39700 IPsec tunnels! If there are two vEdge devices per site, the number is doubled, and so on.

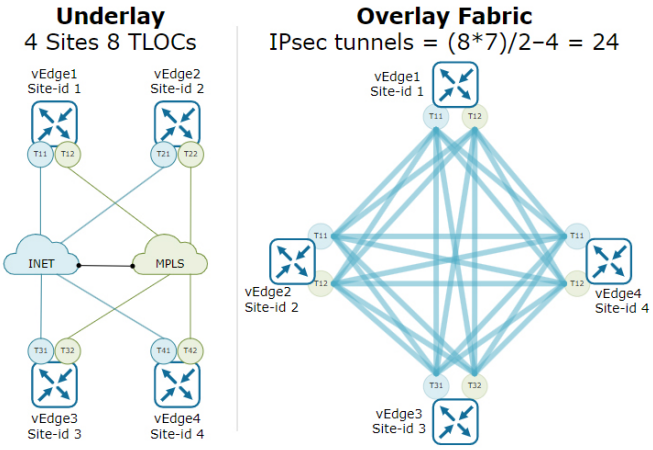


Figure 3. Cisco SD-WAN Full MEsh Overlay Fabric

You can clearly see that in some rare cases, this could be the desired topology, but in most mid-sized and enterprise networks, there is little need to have direct branch-to-branch communication. There is also a scaling limitation because WAN edge devices at the remote sites are typically not sized to handle hundred of thousands of IPsec tunnels and BFD sessions. A better more practical design approach is the use of a Hub-and-Spoke topology.

Configuring a Centralized Policy

In Cisco SD-WAN, to implement hub and spoke topology means to restrict the spoke-to-spoke overlay connections. To do this, a centralized policy must be created and applied so that the remote sites will only receive the Transport Locators of the data center WAN Edges from the vSmart controllers. Each vEdge router attempts to establish a tunnel to all known TLOCs, so the goal is to make the vSmart advertise only the data center TLOCs toward the branches.

For this set of lab examples, we are going to use the topology shown in figure 4.

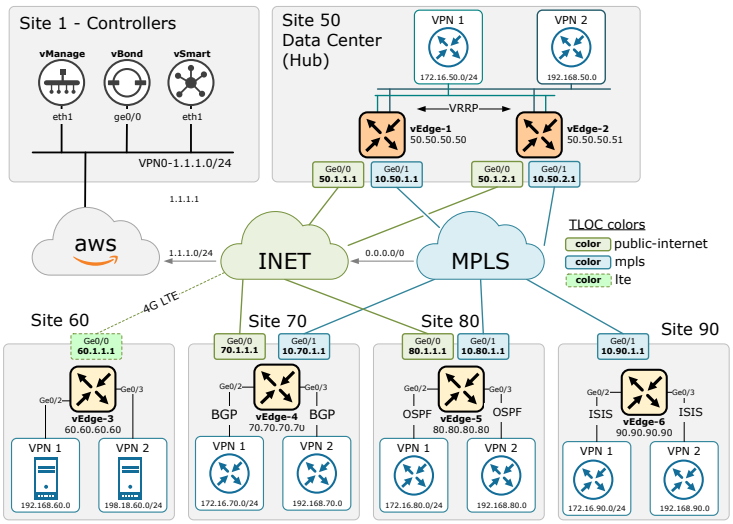


Figure 4. Cisco SD-WAN Deep-Dive - Main Lab Topology

The end goal after applying the centralized policy is to have the data center Site-50 act as a hub and all other sites (60,70,80,90) act as spokes and establishing tunnels to the DC only and not in between them.

To understand the overlay fabric at the moment, let's check how many BFD sessions one of the spoke routers has. This is the easiest way to quickly check how many operational IPsec tunnel a WAN edge device has.

You can see based on the site-id, that vEdge-4 has overlay connections to all other spoke sites (highlighted in yellow). If we successfully configure the hub-and-spoke topology, we should only see connections to Site-50, which would be the hub site.

The first step to constructing the Centralized Control Policy is to use the Policy Wizard in Configuration > Policies as shown below.

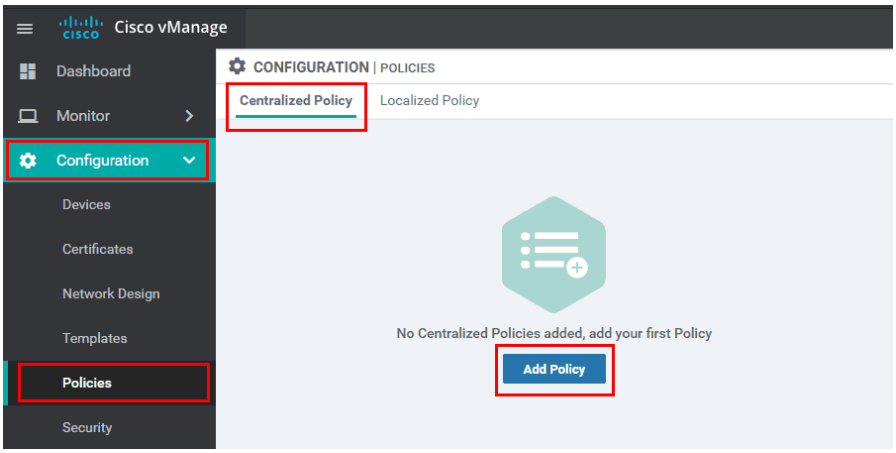


Figure 5. Lab1 - Restricting Spoke-to-Spoke tunnels Step 1

From inside the Centralized Policy Wizard, we need to create two Lists that will be used to match the hub and spokes site-ids. Lists are very simple constructs that are used to match specific values. In Cisco SD-WAN, lists work the same way as they do in traditional networking where we use access-lists and prefix-lists to match specific routes and then specify the action in another construct called a route-map.

For this example, we create one list called Hub that matches site-id 50 and one called Spokes that matches the site-ids of all remote sites - 60,70,80,90.

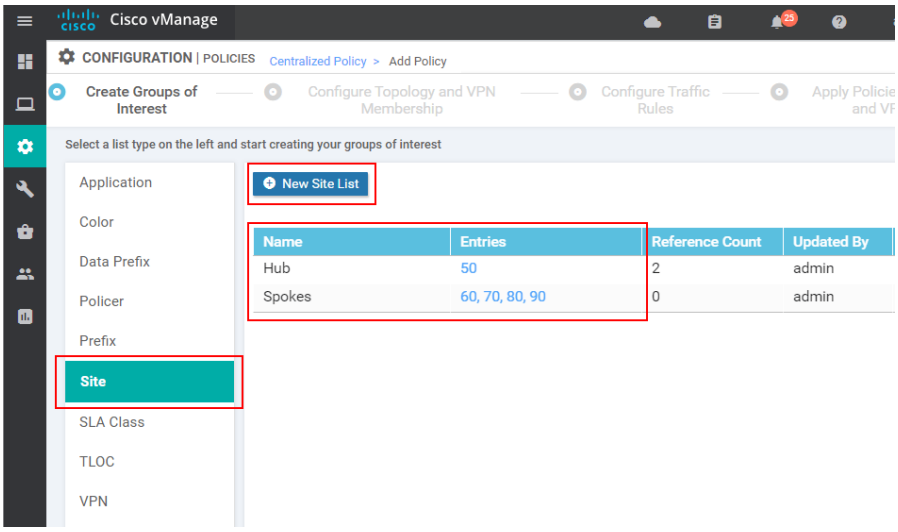
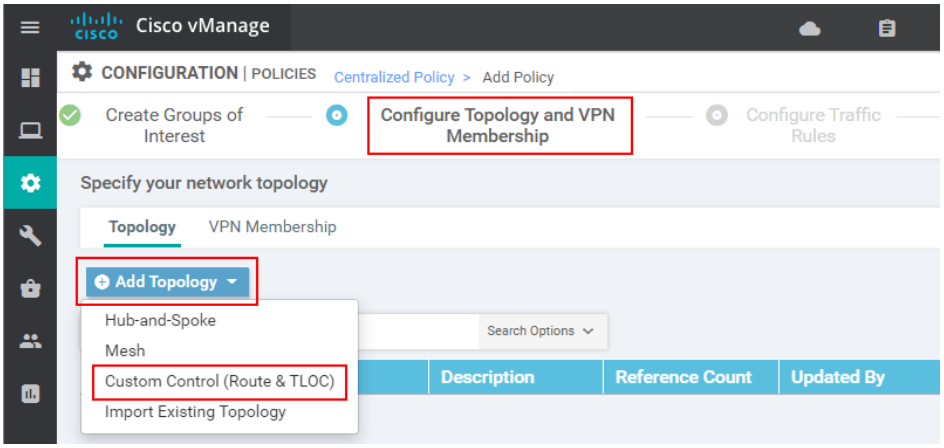


Figure 6. Lab1 - Restricting Spoke-to-Spoke tunnels Step 2

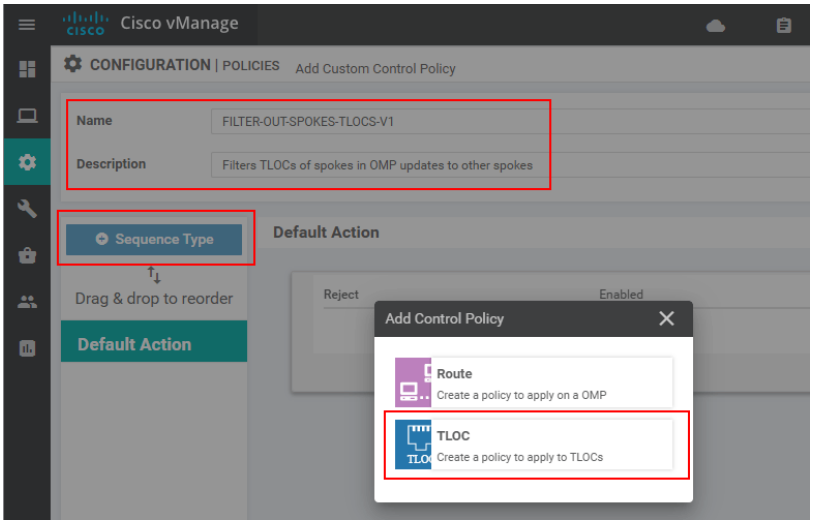
After clicking Next, the wizard moves to the next page called "Configure Topology and VPN Membership". There we select the Custom Control (Route and TLOC) from the Add Topology drop-down menu. You can see in the available options that there is a pre-defined Hub-and-Spoke topology, but to better understand the principles of the Centralized policies, it is better to go down this route.



Lab1 - Restricting Spoke-to-Spoke tunnels Step 3

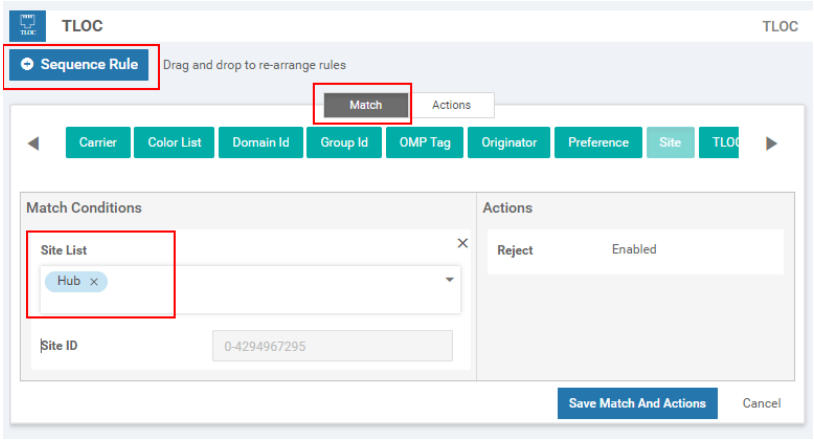
On the next page, the first thing that we have to specify is the Name and Description of the Control Policy. Composing good and scalable policy names is a skill in itself. A general rule of thumb is to use all capital letters and always include a version number.

Next, we click Sequence Type and select TLOC to create a policy that applies to TLOCs.



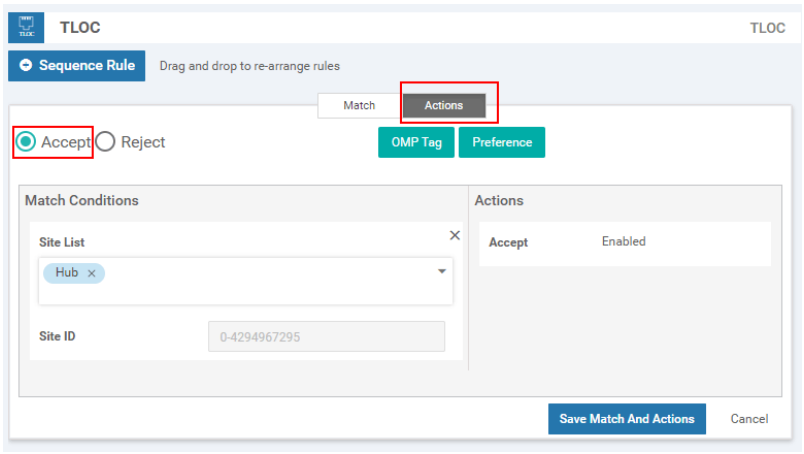
Lab1 - Restricting Spoke-to-Spoke tunnels Step 4

At this point, we should add a new Sequence Rule that matches the Hub sites and then specify Action: Accept. The idea is that the control policy has a default action Reject at the end similarly to Cisco IOS access-lists having explicit deny all at the end. Therefore, the idea is to explicitly math the TLOCs of the Hub site and accept them, and all other TLOCs (of the spoke sites) will be rejected by the default action. In the end, the spokes will only know the TLOCs of the hub and won't establish IPsec tunnels to other spokes.



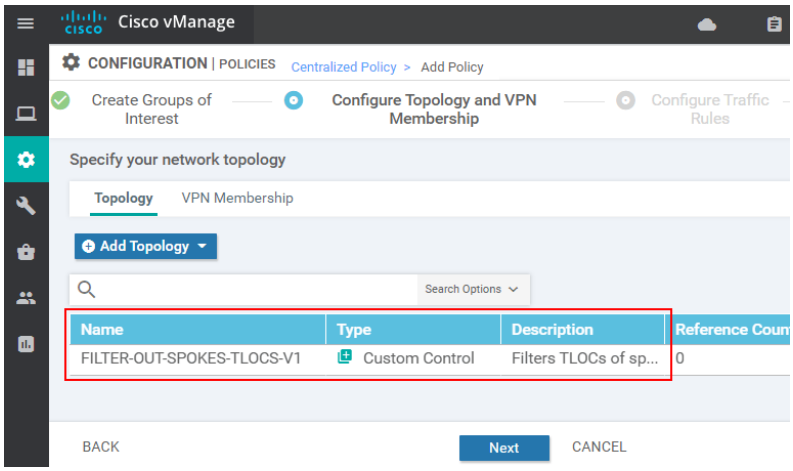
Lab1 - Restricting Spoke-to-Spoke tunnels Step 5

As we explained above, after we match the hub sites, we specify the Accept action.



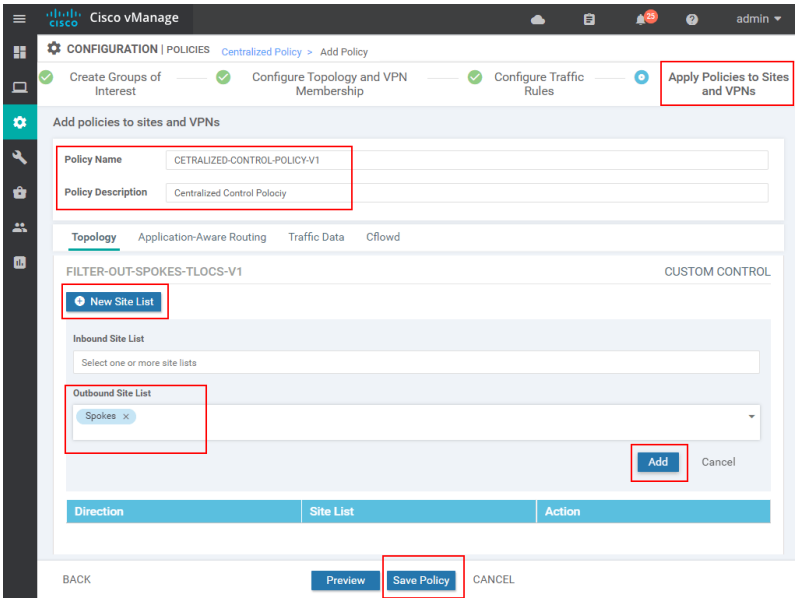
Lab1 - Restricting Spoke-to-Spoke tunnels Step 6

In the end, on the Configure Topology and VPN Membership page, you should see the control policy we have just created. If that is the case, we click Next and go to the Configure Traffic Rules page. For this lab, there is nothing that needs to be configured to this page, so we click Next and go to Apply Policies to Sites and VPNs page.



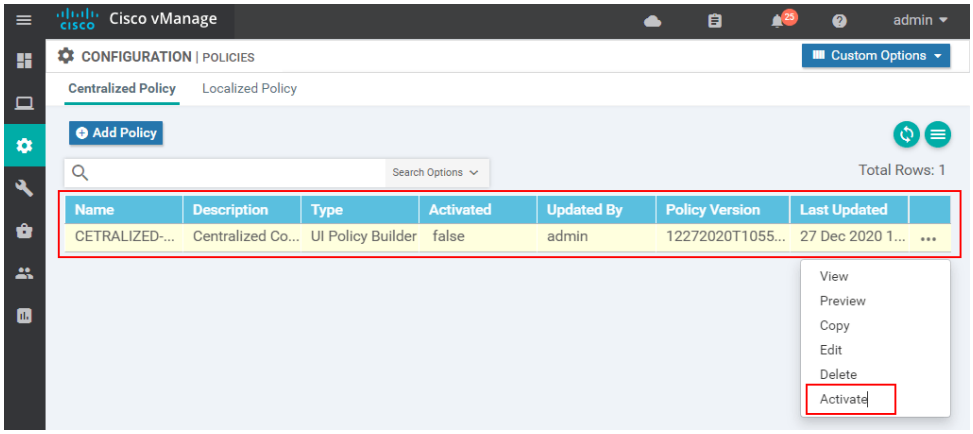
Lab1 - Restricting Spoke-to-Spoke tunnels Step 7

The first piece of configuration required on this page is the policy name and description. Then under the Topology tab, we must specify where and in what direction the FILTER-OUT-SPOKES-TLOCS-V1 control policy will be applied. In our case, we apply it in the outbound direction to the spoke sites.



Lab1 - Restricting Spoke-to-Spoke tunnels Step 8

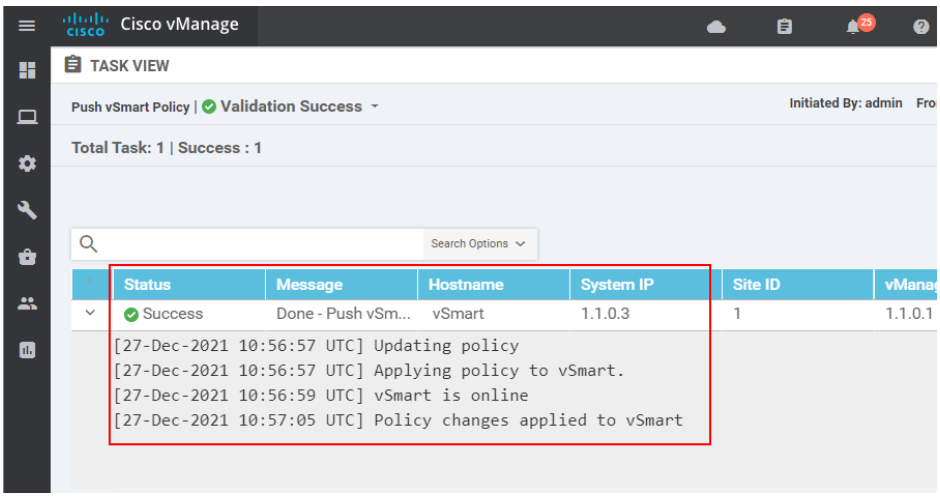
In the end, we should see the Centralized Policy created but not activated. The last thing we must do is to activate it as shown in the screenshot below.



Lab1 - Restricting Spoke-to-Spoke tunnels Step 9

Note that the vSmart won't accept the policy if it is not in a "vManaged mode". This means that the vSmart must have a template applied from vManage. The type of the template does not matter, it could be a Device template or a Feature template, but a template has to be applied. This enables vManage to have authoritative control over the vSmarts. It is common for production deployments to initially deeply the vSmart controllers with a CLI template from vManage, so the controllers are in "vManaged mode" from the get-go.

In our lab example, the vSmart has been initially deployed with a CLI template so vManage can successfully push the Centralized Policy to the controllers.



Lab1 - Restricting Spoke-to-Spoke tunnels Step 10

At this point, the overlay fabric must be analogous to a traditional Hub and Spoke topology. We can now check again the active BDF sessions on one of the spokes WAN edge devices. If our control policy is correct and has taken effect, we must see IPsec tunnels to the hub site only (site-id 50) and not to other spokes (sites 60,70,80,90).

Key Takeaways

Let's recap what we have done in this lab and highlight the key takeaways from this lesson. Use figure 15 for a reference.

We created a Centralized Policy using the vManage GUI. In this policy, we have created another Control Policy that matches the hub TLOCs and accepts them, all other TLOCs are rejected by the default action at the end - Reject. After that, we applied this policy in an outbound direction to the spoke sites. Note that the OUT direction is from the perspective of the OMP updates of the vSmart controller to the WAN edge devices. So what happened after we activated the policy. As it is shown in figure 15, vManage pushed this Centralized policy as a NETCONF transaction to the vSmart controllers.

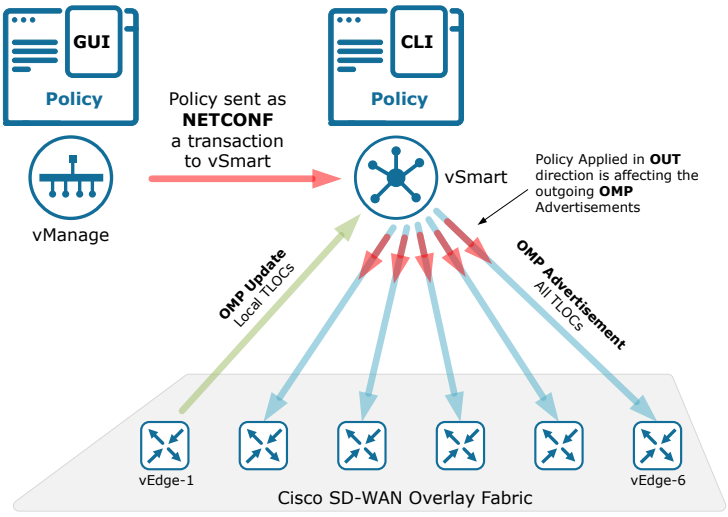


Figure 5. Cisco SD-WAN Policy Direction

If we check the policy in the running configuration of the vSmart, it looks like the output shown below. Note that it has a very similar construct as the route-maps in Cisco IOS. We have lists that match something (in green). In route-maps, we typically use access-lists, prefix-lists, or as-path lists to specify values of interest.

Then we have a sequence with a match-action construct (in yellow). In Cisco IOS we have route-maps with similar match-set logic, for example, match ACL1 set local-preference 110.

And then we apply this match-action construct somewhere (in orange). In the case of Cisco SD-WAN, we have applied it to the OMP updates in the outbound direction. In the case of a typical route-map, we apply it on a BGP neighbor for example.

You can see that there is nothing quite new in the way the overlay fabric is configured with Centralized Policies, only the underlying mechanisms are different. The logical construct is pretty much the same as it has always been in traditional networking.

LAB 2: Hub-and-Spoke - Allowing hub-to-spoke routing

In our previous lab lesson for Cetralizaed Policies, we have restricted spokes of establishing data plane tunnels to other spokes by filtering out TLOC advertisements at the vSmart controllers. This allowed us to create a hub-and-spoke overlay topology of IPsec tunnels. However, after we finished with the topology manipulation, we didn't check whether there is actual IP reachability between the hub and the spokes and between the spokes themselves.

Let's look at our main topology for this lab series and check if any of the spokes (site-ids 60,70,80,90) has connectivity to the hub (site-id 50) or to other spokes.

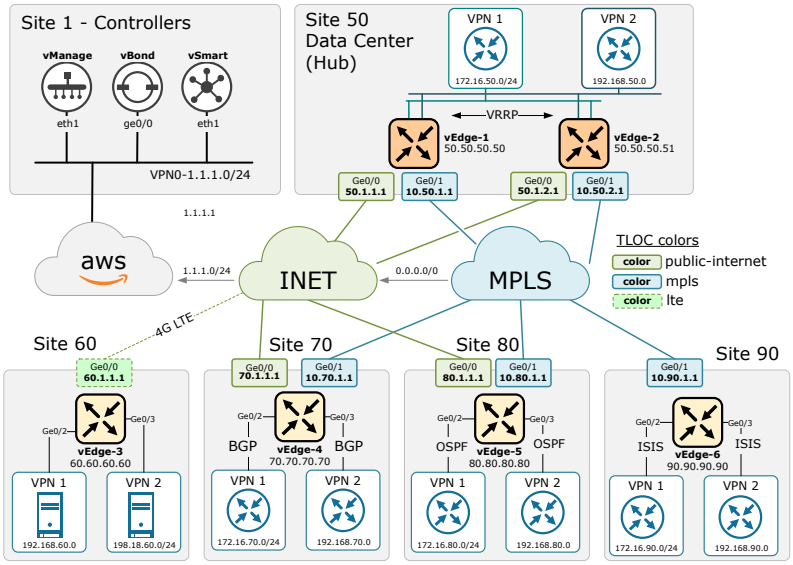


Figure 1. Cisco SD-WAN Deep-Dive - Main Lab Topology

We can verify if vEdge-4 (site 70) has all route entries in VPN1 by checking its routing table for that VPN. Obviously, it only knows about its own connected network and do not receive any routing information about the other sites.

There had been full IP connectivity between all VPN1 subnets in all sites before we had applied the Centralized Control Policy that changed the overlay topology. But obviously, there is no reachability neither between the spokes nor between the hub and the spokes at the moment. Let's see why this happens?

Default Overlay behavior without Centralized Control Policy

By default, the Cisco SD-WAN Overlay Fabric builds a full-mesh of data plane tunnels and there is no Centralized Control Policy applied on the vSmart controllers.

All WAN Edge devices redistribute all prefixes they learn from their site-local network to the Cisco vSmart Controller. This route information is sent via OMP route advertisements to the vSmart controllers.

- All WAN Edge devices redistribute all prefixes they learn from their site-local network to the Cisco vSmart Controller. This route information is sent via OMP route advertisements to the vSmart controllers.
- All WAN Edge devices send their respective TLOCs to the vSmart controllers using OMP.
- The Cisco vSmart Controller accepts all the OMP route and TLOC updates it receives from all the devices and redistributes them to all the devices in the same VPN.

That is why we had had full IP connectivity before the Hub-and-Spoke Control Policy was applied. Let's now look at how the overlay fabric's behavior changes when a policy is applied.

Behavior with Centralized Control Policy

By default, the vEdge controller redistributes all route information to all WAN edge devices. When this is not the desired behavior or when we want to modify the route information stored in the Cisco vSmart's route table or that is advertised by the Cisco vSmart Controller, a centralized control policy must be provisioned and applied. The policy is then activated in either inbound or outbound direction with respect to the vSmart controller. This is visualized in figure 2.

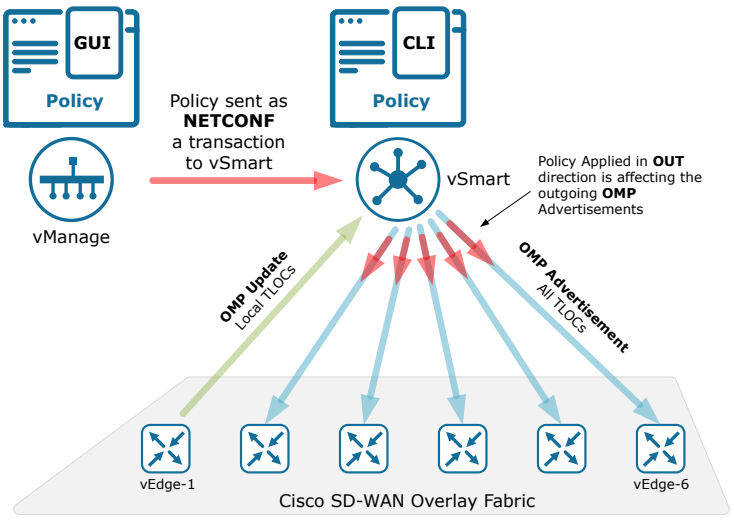


Figure 2. Cisco vSmart Control Policy Directions

Let's now recall what we did in the previous lesson. We created a Centralized Policy using the vManage GUI. In that policy, we created another Control Policy that matches and accepts the hub TLOCs, all other TLOCs are rejected by the default reject action at the end. After that, we applied this policy in an outbound direction to the spoke sites. Note that the OUT direction is from the perspective of the OMP updates of the vSmart controller to the WAN edge devices.

Now that there is a Centralized Policy applied to the spoke sites in the outbound direction, all OMP updates go through the match-action sequence defined in the Control Policy. In that sequence, we matched the data center TLOCs and accepted them, but we didn't define any match-action clause for the routing updates. Therefore, the route advertisements are matched at the default Reject action. That is why we do not see any routes coming via omp at the spokes.

Let's create another centralized policy that filters out TLOCs to spoke but also permits the route advertisements at the same time.

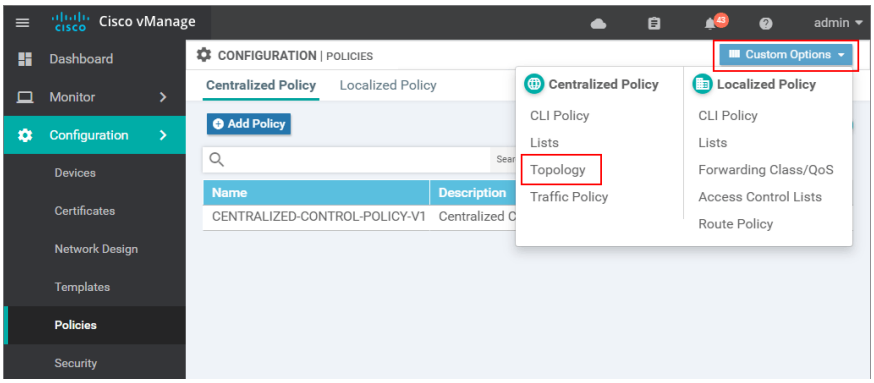
Configuring a Centralized Control Policy

If we log into vManage and enter the Policy Wizard in Configuration > Policies we can see the Centralized Policy we have created in the last lesson - "CENTRALIZED-CONTROL-POLICY-V1".

IMPORTANT Centralized Policies cannot be edited while they are active. If we want to modify something, we create a new centralized policy and activate it.

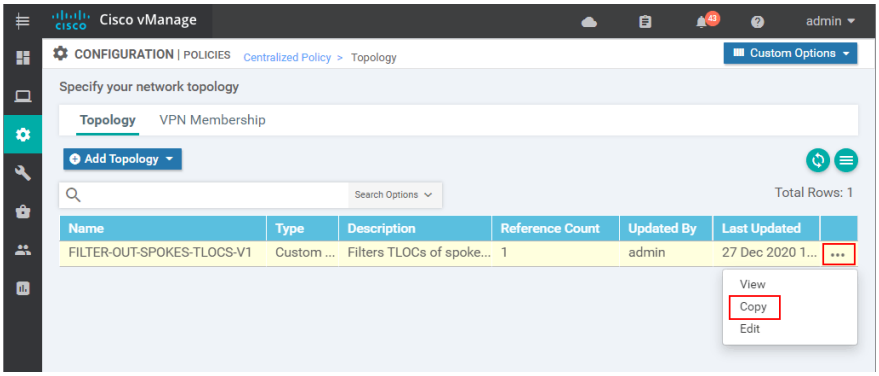
Therefore, we should create another policy - "CENTRALIZED-CONTROL-POLICY-V2" reusing the same parts of the previous one, and push it to the vSmart controller. You can now see why it is a good practice to have a version number in the name.

To reuse existing parts of a centralized policy, we make a copy of them and modify the copy. Let's see this in action, go to the Custom Options drop-down menu and select Topology as shown in step 1.



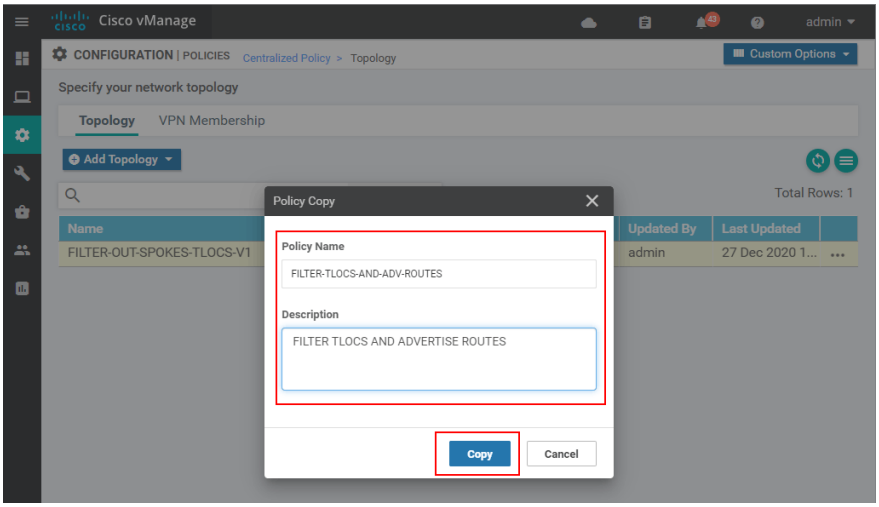
Configuring a Centralized Control Polocy - step 1

There you will see the Topology policy we have created in the last lesson. To reuse it, we make a copy of it with another name and description.



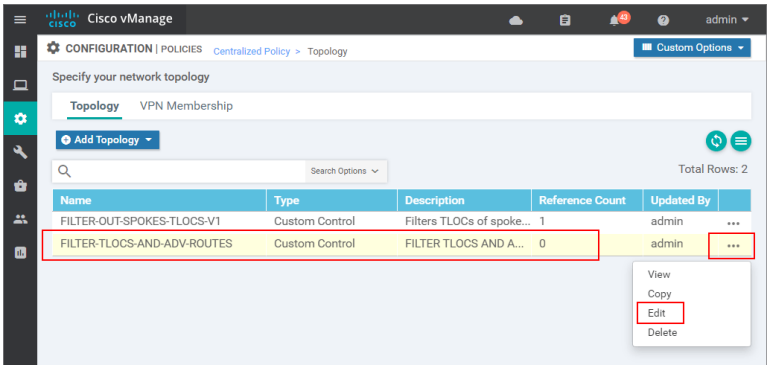
Configuring a Centralized Control Polocy - step 2

The first thing that must be specified in order to make a copy is the name and description. For this example, we are going to use "FILTER-TLOCS-AND-ADV-ROUTES".



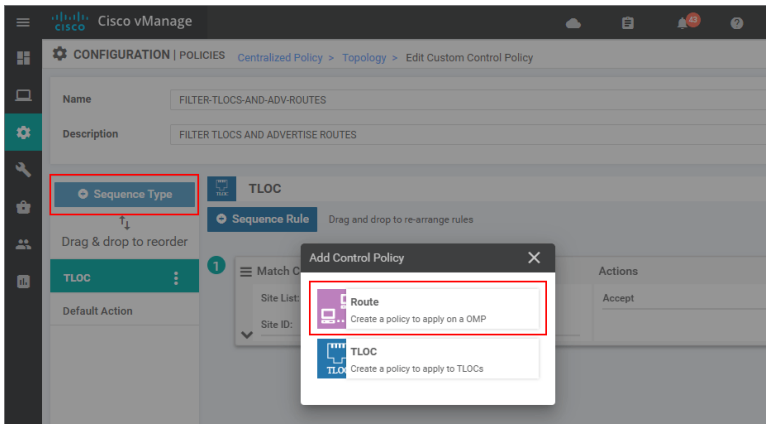
Configuring a Centralized Control Polocy - step 3

Now you can see that we have two topology definitions. The newly created one has a Reference Count of 0, which means that it is not used anywhere, so we can safely edit it without affecting anything in production.



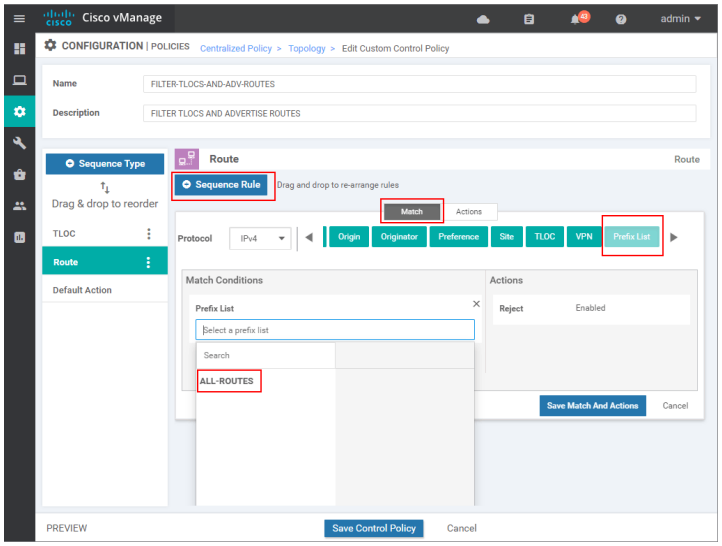
Configuring a Centralized Control Polocy - step 4

This leads us to the Centralized Policy > Topology > Edit Custom Control Policy page. There you will find the TLOC sequence we have created in the previous lesson because we use a copy of the last lesson's policy. Now we want to add another match-action sequence, but this time of type Route because we want to manipulate the route advertisements. Click on Sequence Type > Route.



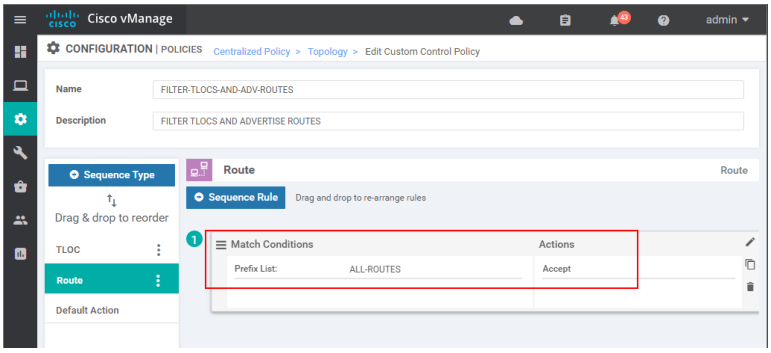
Configuring a Centralized Control Polocy - step 5

Then we add a new Sequence Rule that matches a prefix-list ALL-ROUTES. This is a pre-defined prefix-list that matches all routes - 0.0.0.0/0 le 32.



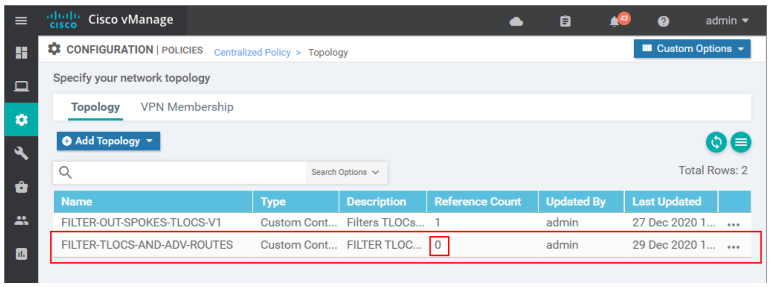
Configuring a Centralized Control Polocy - step 6

Then we go to the Actions tab and specify Accept action. In the end, we should have the following match-action condition. Then we click save.



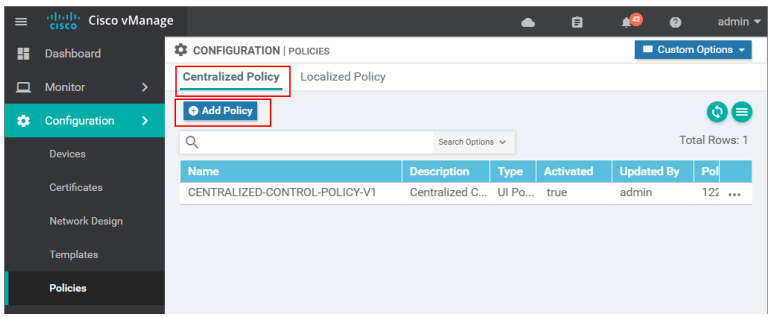
Configuring a Centralized Control Polocy - step 7

At this point, we have created a new topology match-action definition. However, based on the Reference Count = 0, you can see that it is still not used anywhere. We are going to create



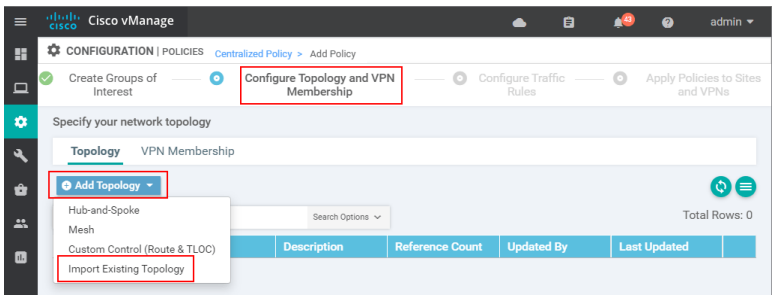
Configuring a Centralized Control Polocy - step 8

Then we go back to the Policy Wizard in Configuration > Policies and click Add Policy.



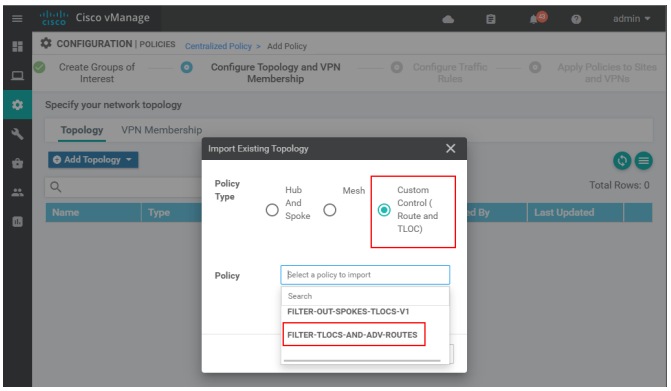
Configuring a Centralized Control Polocy - step 9

We skip the first Create Groups of Interest page because we do not need any new lists. On the Configure Topology and VPN Membership page, we go to Add Topology, but this time we select Import Existing Topology.



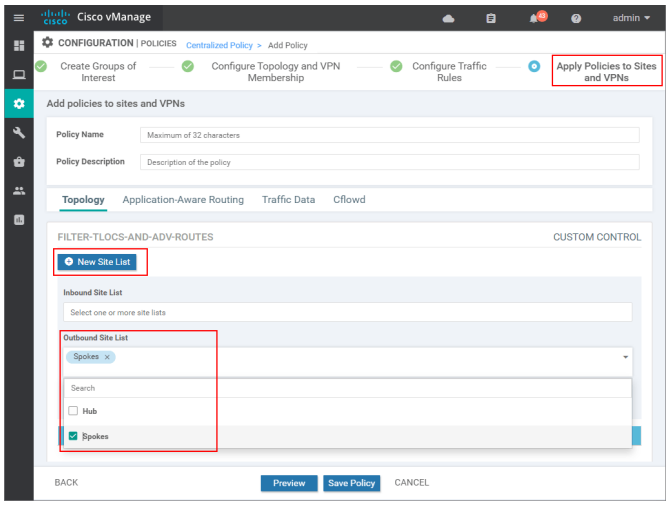
Configuring a Centralized Control Polocy - step 10

There under the Custom Control (Route and TLOC) radio button, we should see the new custom topology policy and we have just created - "FILTER-TLOCS-AND-ADV-ROUTES".



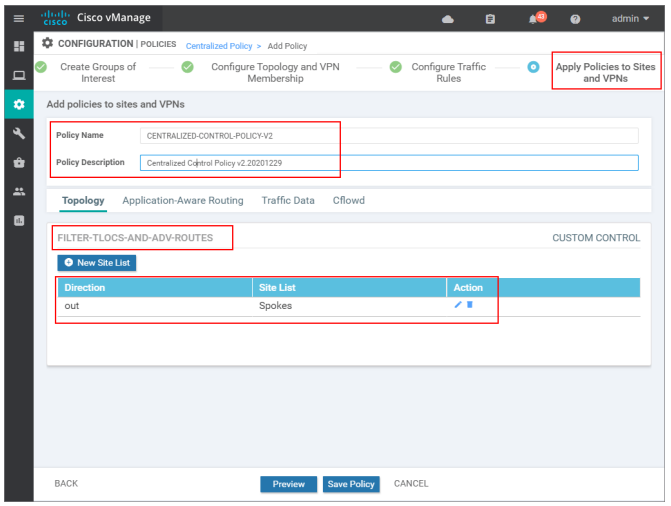
Configuring a Centralized Control Polocy - step 11

Then we skip the Configure Traffic and Rules page and end up on the Apply Policies to Sites and VPNs. There we apply the policy in an outbound direction to the spokes using the Spokeslists we have created in the last lesson.



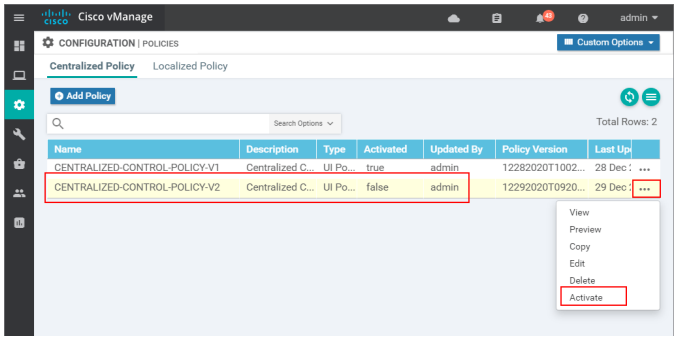
Configuring a Centralized Control Polocy - step 12

In the end, we specify the name and description, verify that the Direction and Site list are correct, and click Save.



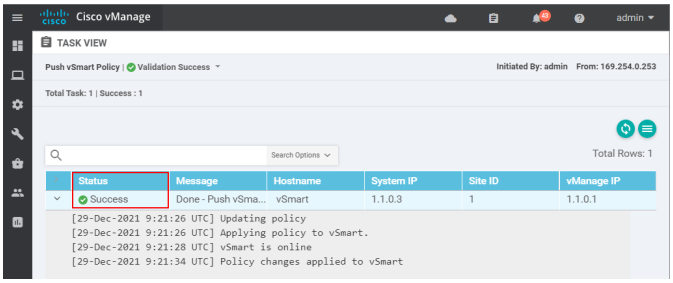
Configuring a Centralized Control Polocy - step 13

Now you can see that we have two Centralized Policies. You can see that the one we have created in the last lesson has the Activated tab value as true, which means that it is currently active on the vSmart controllers. Now we need to activate the new one that we have made in this lab as shown in the following screenshot.



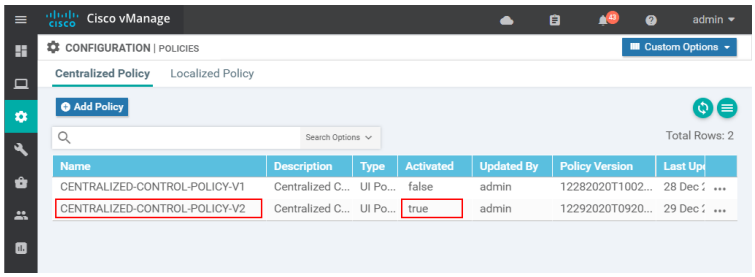
Configuring a Centralized Control Polocy - step 14

If the vSmart controller is online and is in vManaged mode, the policy should get pushed successfully.



Configuring a Centralized Control Polocy - step 15

Now the new one which has V2 in the name must have the Activated value true. You can now see why having some kind of version control in the name is really important. You can keep track of the configuration changes and can easily roll back to a previous version in case of a problem.



Configuring a Centralized Control Polocy - step 16

Figure 3 visualizes what we have done in this lab using the Policy Wizard. We have created a policy named CENTRALIZED-CONTROL-POLICY-V2 that holds everything together. There we use the two lists HUB and SPOKES that we had specified in the previous lesson. We have created Custom Topology definition called FILTER-TLOCS-AND-ADV-ROUTES. Inside it, sequence 1 of type tloc matches and accepts tlocs from the hub site. Sequence 11 of type route, matches and accepts all routes. Then, in the end, we have the default reject-action. At first, for most network engineers that are used to starting at CLI outputs, this is more clear and easy to understand than the vManage GUI. In the end, we apply this topology definition in an outbound direction to the spokes.

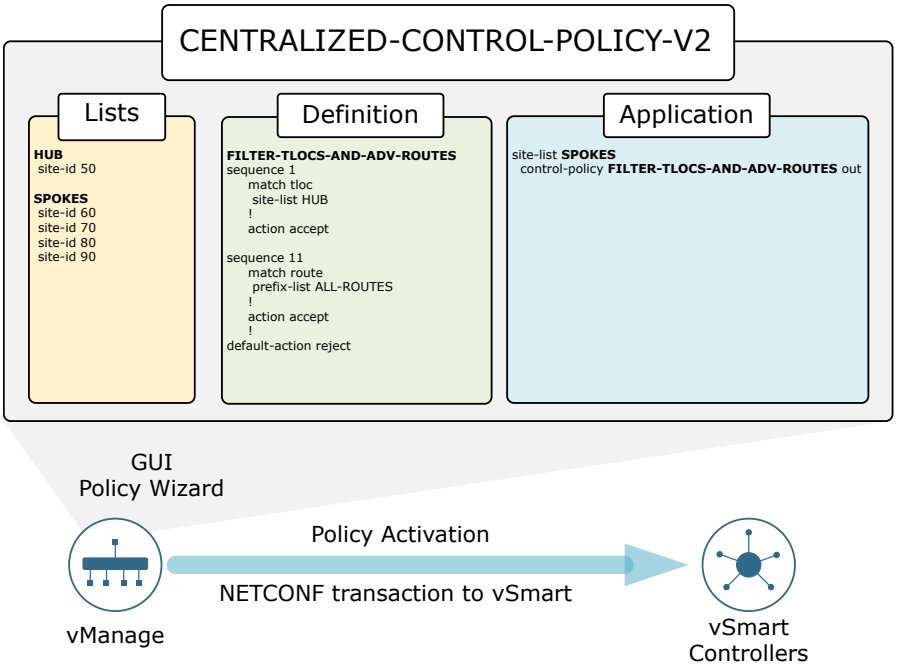


Figure 3. The Centralized Policy Lab 2

You can see that the vSmart controller in a way acts as a BGP route reflector. It reflects the omp routes but does not change the next-hop TLOC addresses. In a full mesh overlay topology, this is the desired behavior, but in the case of a custom topology as the hub-and-spoke overlay we have created, this breaks the data plane.

There are two ways to resolve this problem and enable spoke-to-spoke communication through the hub:

- Using summarization - The simplest way to enable branch-to-branch communication is to inject a default route into the overlay from the hub's vEdges. This is the technique we are going to use in this lesson.
- Using TLOC lists - The 'software-defined' approach to fix the data plane would be to edit our Centralized Policy in such a way that the vSmart controller advertises all spokes' routes with "next-hop" TLOC addresses the TLOCs of the hub's vEdges. This is what we are going to do in the next lab lesson in this series.

Injecting a default route is a pretty simple and straightforward process. We need to create a static route to 0.0.0.0/0 to null0 and just tell the omp protocol that we want to redistribute static routes.

LAB 3: Hub-and-Spoke - Enabling spoke-to-spoke communication

Recap of the previous lab

In lab 2, we have created a Centralized Policy named CENTRALIZED-CONTROL-POLICY-V2 using the vManage GUI. In that policy, we have created a Topology definition that matches the Transport Locators of the hub site (based on the site-id of the hub) and accepts them, all other TLOCs are rejected by the default action at the end which is Reject. In another sequence, we matched all omp routes and accepted them. After that, we applied this policy in an outbound direction to the spoke sites. Note that the out direction is from the perspective of the OMP updates of the vSmart controller to the WAN edge devices.

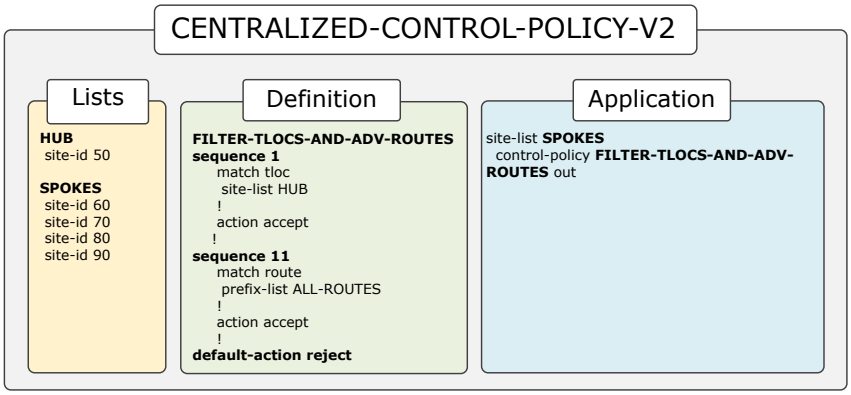
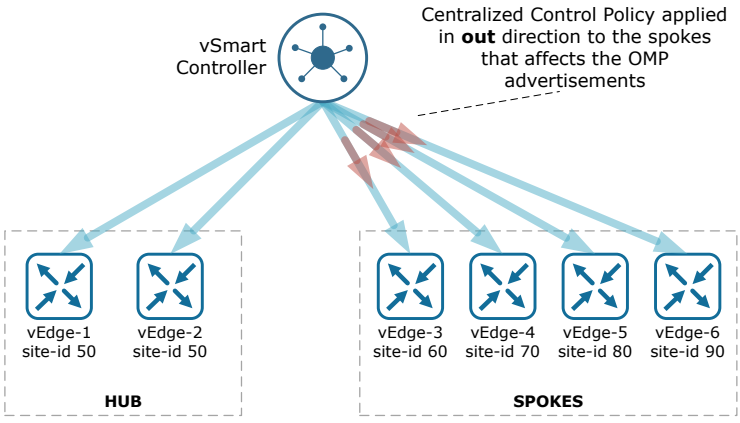


Figure 1. Centralized Policy used in Lab 2

So what are the results of this Centralized policy:

- Due to sequence 1 of the topology definition - the vEdges at the spoke sites receive only the transport locators (TLOCs) of the hub site. Therefore, they can only make overlay tunnels to the hub's vEdges which basically creates a hub-and-spoke overlay fabric.
- Due to sequence 11 - the vEdges at the spoke sites receive all omp routes from all sites. This means that each WAN edge router at the spoke sites receives omp routes from all other spoke vEdges. These omp routes have next-hop TLOC attributes pointing to the originating spoke vEdges. However, spoke vEdges do not know the transport locators of other spokes because spokes' TLOCs have been filtered out in sequence 1. That's is why the omp routes from other spokes have a status Invalid and Unresolved and are not installed in the routing table.

To enable spoke-to-spoke reachability, we have injected a default route from the hub's vEdges as a workaround to the issue created by sequence 11 of the centralized policy.



Centralized Policy from lab 2 in out direction to spokes

Typically, each attribute could be used as a match criteria and each one could also be modified with a set action. In this lab, we are going to manipulate the TLOC attribute of the routes that are advertised to the spoke vEdges. Because the spokes only know about the TLOCs of the hub site, we are going to change the tloc attributes of the omp routes to point to the available hub's TLOCs.

Modifying OMP routes with TLOC lists

Before we begin with this lab, let's first have a look at our topology diagram that we use for this Cisco SD-WAN Deep Dive series.

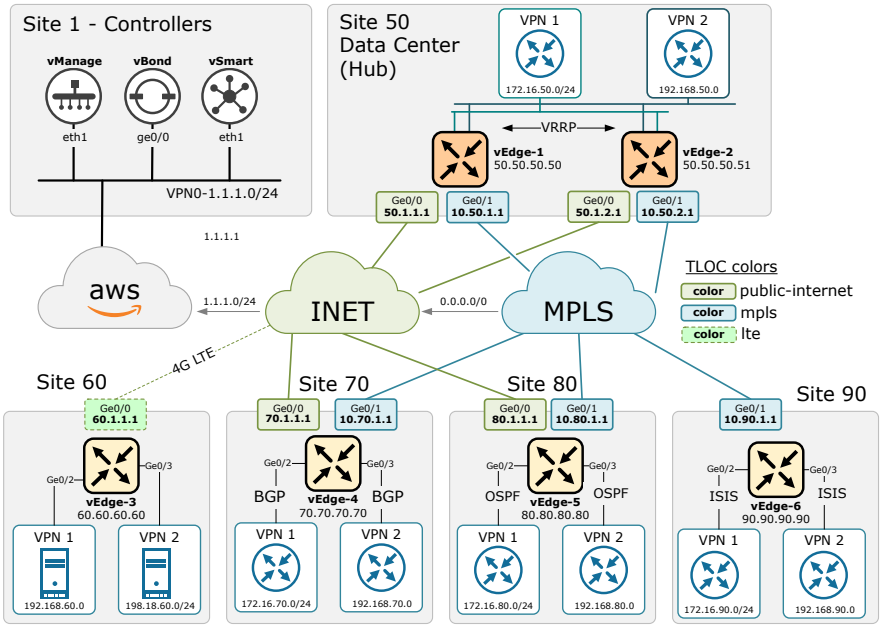
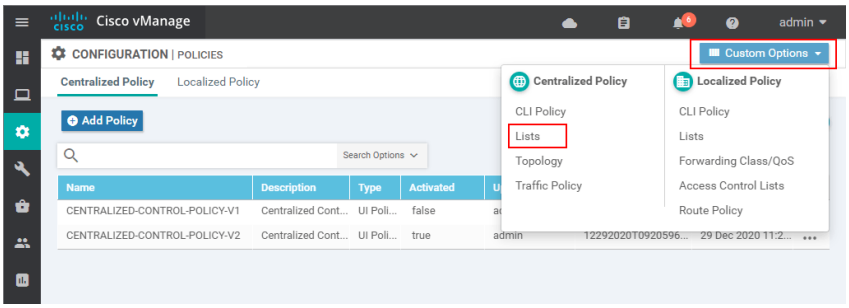


Figure 3. Cisco SD-WAN Deep-Dive - Main Lab Topology

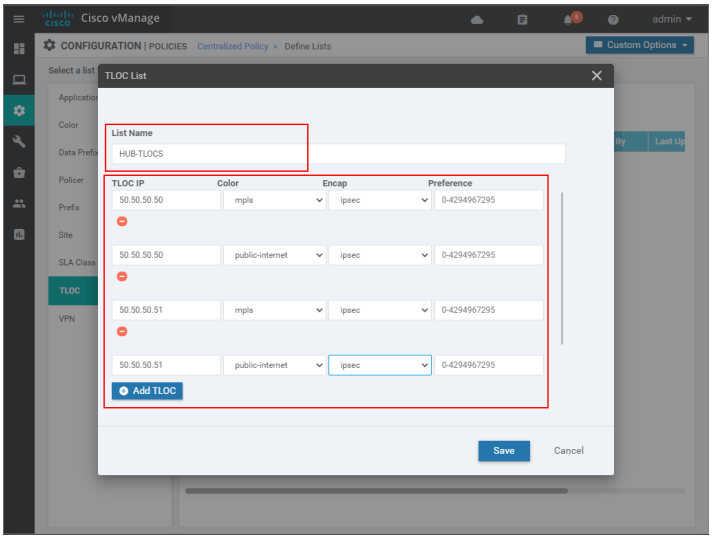
If we log into vManage and enter the Policy Wizard in Configuration > Policies we can see the Centralized Policy we have created in the last lesson - "CENTRALIZED-CONTROL-POLICY-V2" is currently active. As we learned in the previous lab lesson, centralized policies could not be edited while active. Therefore, we are going to create create a new version of the policy reusing some parts of the current one and when we are done, we are going to push the new version to vSmart.

For this lab, we are going to need a new list that matches the transport locators of the hub's vEdge routers. Let's create a new list called HUB-TLOCS by going to Custom Options > Lists.



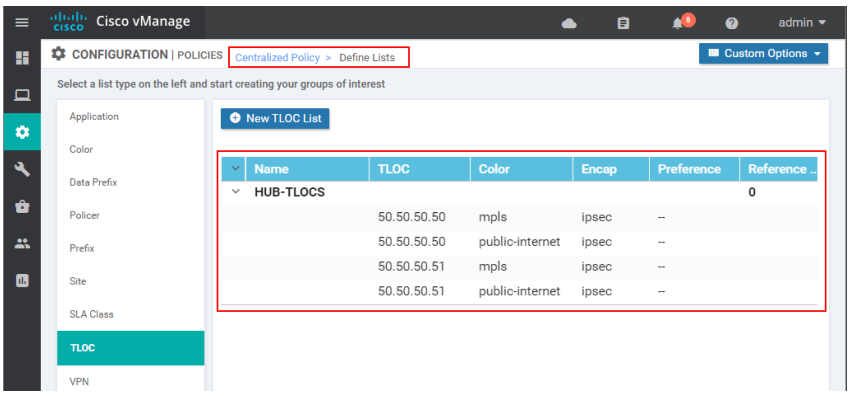
Enabling spoke-to-spoke communication - Step 1

In there, we select TLOC and enter the name of the list. After that, we specify the four transport locators that we have in the hub site. Note that a TLOC consists of three tuples of information - (TLOC IP, Color, Encap). The TLOC IP is typically the system IP address of the vEdge router. Based on personal experience, I can say that it is a common mistake to use the IP address of the interface connected to the transport network. I had personally done this many times before. For example, if you look at figure 3, it is a common mistake to use the TLOC (10.50.1.1, mpls, ipsec) for the transport locator of vEdge-1 that identifies the connection to the MPLS cloud. However, the correct TLOC tuple is (50.50.50.50., mpls, ipsec).



Enabling spoke-to-spoke communication - Step 2

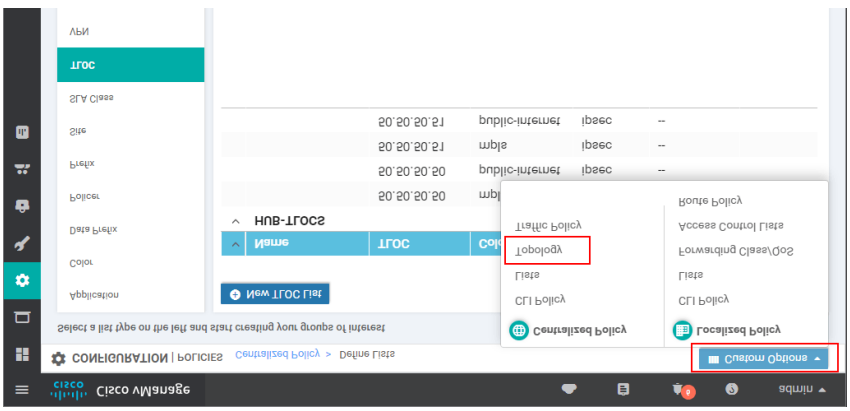
Once the HUB-TLOCS list is created, note that the Reference Count is 0, because it is not used anywhere at the moment.



Enabling spoke-to-spoke communication - Step 3

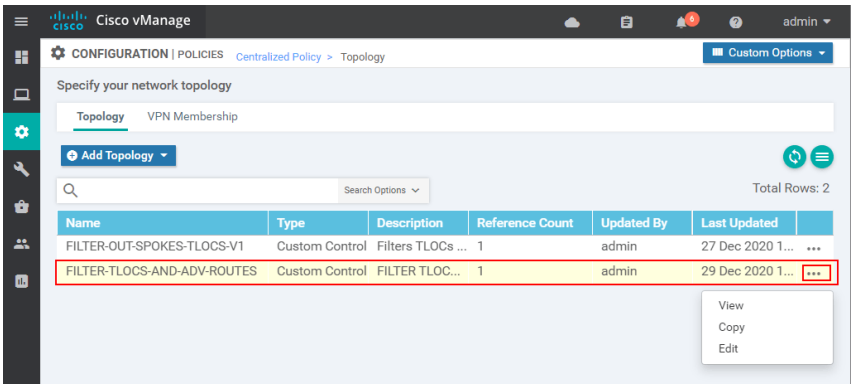
Now we need to create the Topology definition that will change the next-hop TLOC attributes of the OMP routes advertisements to the spokes. Let's reuse the topology policy that is currently in use. To do this, we copy the existing one using a different name and we modify it as per the requirements.

Go to Custom Options > Topology.



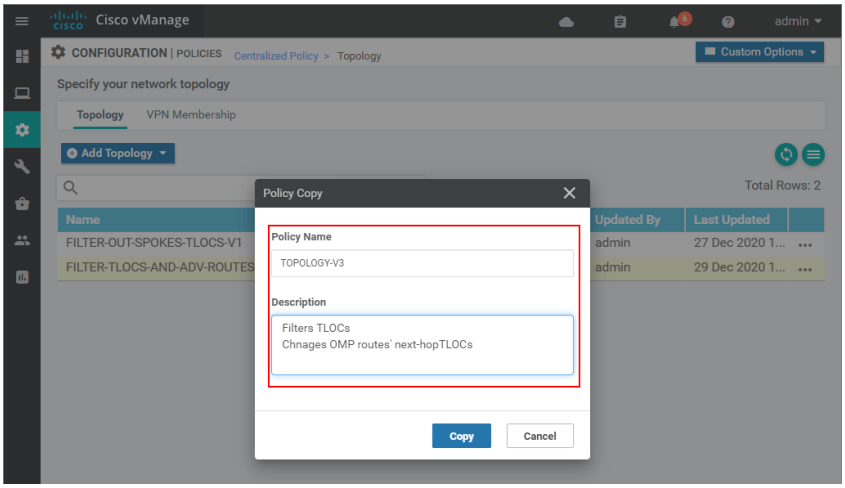
Enabling spoke-to-spoke communication - Step 4

In there we must see that policy FILTER-TLOCS-AND-ADV-ROUTES that we have created in the previous lab lesson. Note that the Reference Count is 1 because it is currently in use. To reuse it, we make a copy of it with a different name.



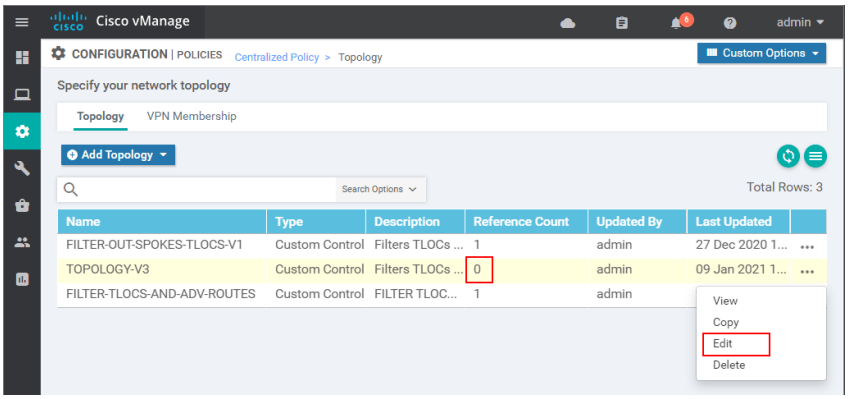
Enabling spoke-to-spoke communication - Step 5

For this lab lesson, we are going to use the name TOPOLOGY-V3.



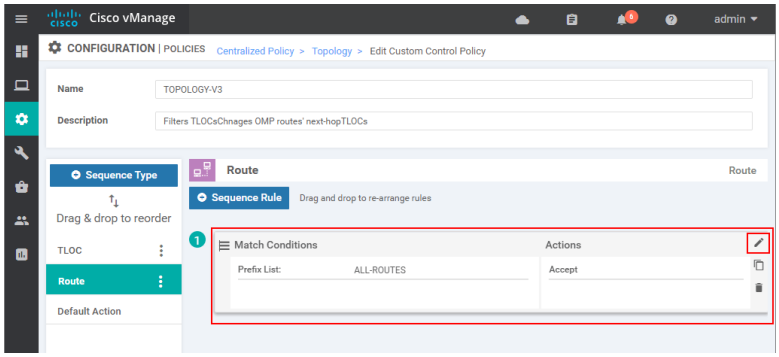
Enabling spoke-to-spoke communication - Step 6

Now you can see that the new topology policy has a Reference Count of 0 meaning that it is not used anywhere. Therefore, we can safely edit it without touching anything in production. From the additional options, we select Edit.



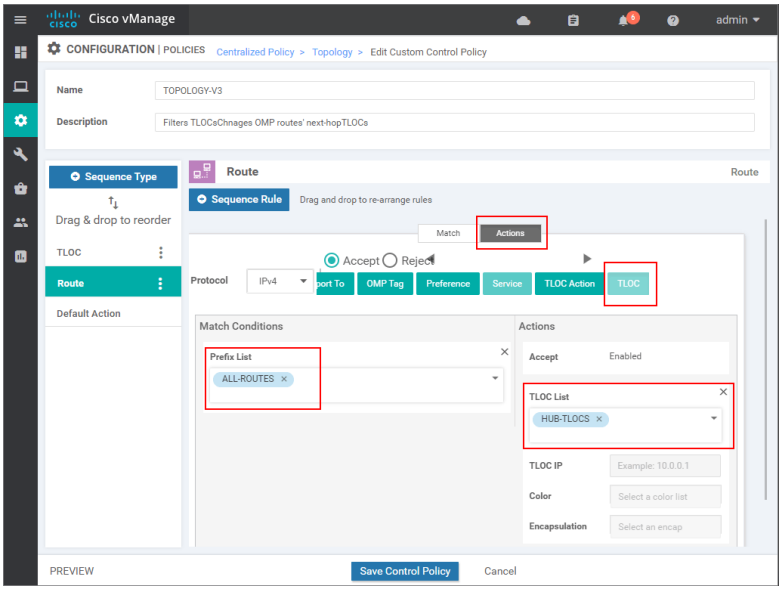
Enabling spoke-to-spoke communication - Step 7

In there on the left side, we can see the two sequence definitions that we have created in the previous lab. If we select the Route sequence type, we can see that it matches the ALL-ROUTES prefix-list and Accepts it. However, as we said earlier, we'd like to not only accept the routes but also to modify the next-hop TLOC attributes. That is why we are going to edit this match-action sequence.



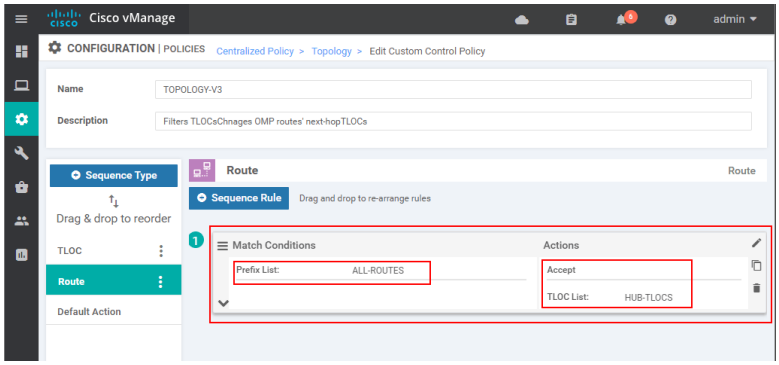
Enabling spoke-to-spoke communication - Step 8

There we go to the Actions tab > TLOC and select the HUB-TLOCS list that we defined earlier. Then we save the policy.



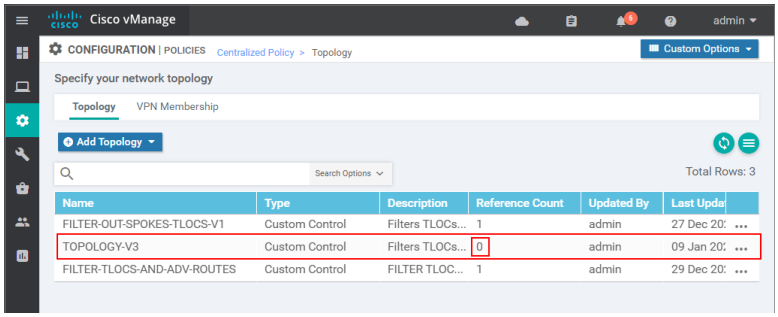
Enabling spoke-to-spoke communication - Step 9

In the end, we should have a match condition that matches the pre-defined prefix-list ALL-ROUTES and the applied action should be the TLOC list HUB-TLOCS.



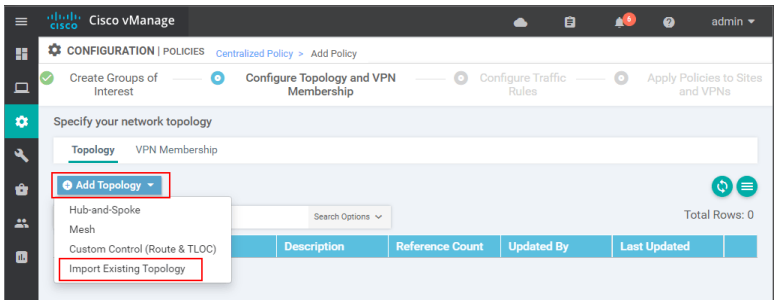
Enabling spoke-to-spoke communication - Step 10

At this point, the new Topology definition is done but obviously, it is still not used anywhere.



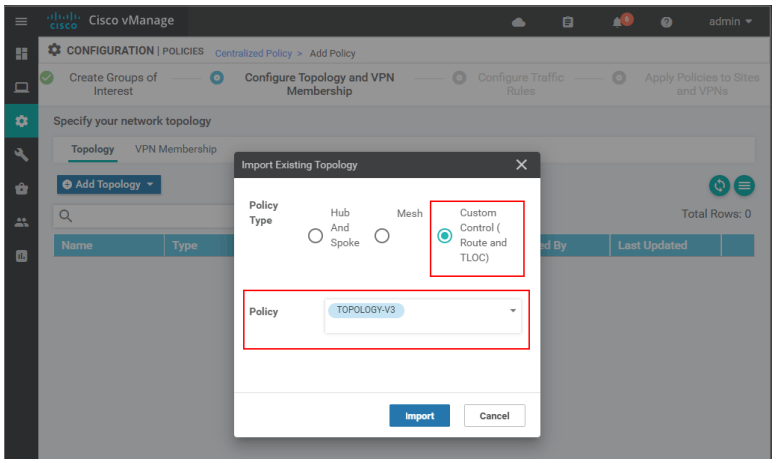
Enabling spoke-to-spoke communication - Step 11

At this point, all required parts for our new Centralized Control Policy are made so we can go ahead and define the policy through the wizard. We go to Policies > Add Policy. In there, we skip the first page "Create Groups of Interest" because we have already defined all required lists. On the next page "Configure Topology and VPN Membership" we go to Add Topology > Import Existing Topology because we have already defined a Topology policy (named TOPOLOGY-V3).



Enabling spoke-to-spoke communication - Step 12

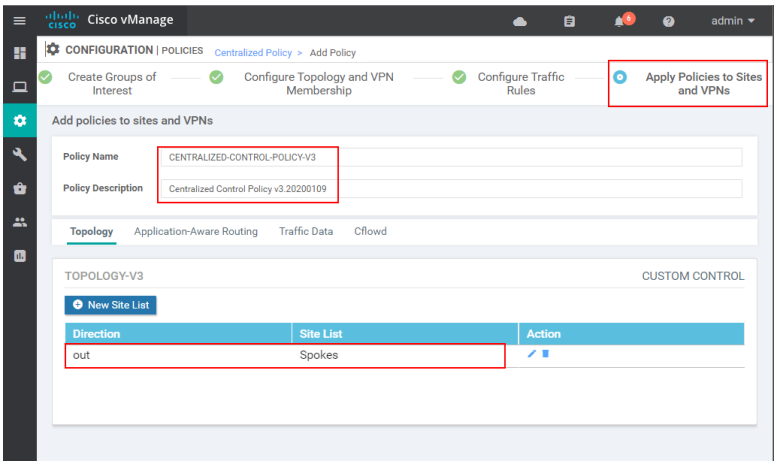
In the pop-up window, we go to Custom Control (Route and TLOC) and select the policy have defined earlier - TOPOLOGY-V3 from the drop-down menu.



Enabling spoke-to-spoke communication - Step 13

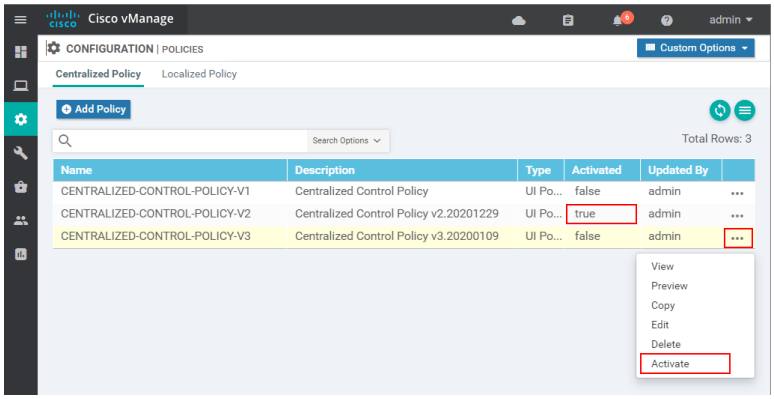
Then we skip the next page "Configure Traffic Rules" and end up on the "Apply Policies to Sites and VPNs" page. In there, we specify a name and description of the policy. As we said in the previous lessons, it is a general rule of thumb to use all capital letters for the name and always include some kind of a version number.

After that apply the policy in an outbound direction to the spokes by selecting the site-list SPOKES.



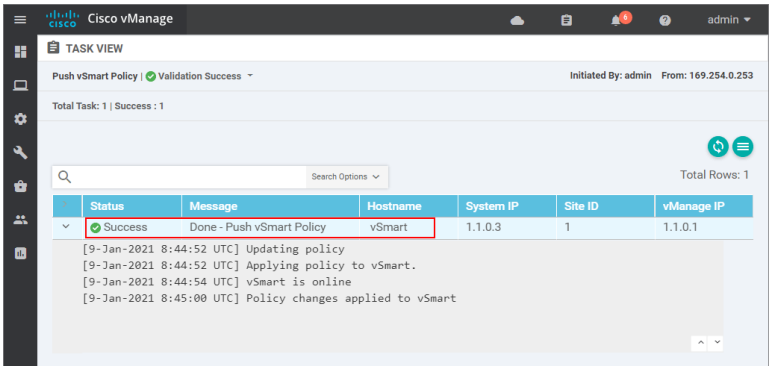
Enabling spoke-to-spoke communication - Step 14

Now you can see that we have a new Centralized Control policy created that has a V3 at the end of the name. Note that it is not activated though. Still, the policy that we have created in the previous lesson is active. To activate the new one we go to the options and click Activate.



Enabling spoke-to-spoke communication - Step 15

If the policy is pushed successfully to all vSmart controllers (in our case it is only one), we must see a status - Success.



Enabling spoke-to-spoke communication - Step 16

LAB 4: Traffic Engineering - TLOC Preference

In this lab, we are going to make some traffic engineering by modifying the TLOC preference of one of the WAN edge routers at the hub site.

The Lab Topology

By default, the overlay fabric load-share the traffic across all equal paths. In our lab topology, this means that the traffic towards the datacenter is load-shared across all IPsec tunnels between vEdge-4 and vEdges 1 and 2. If we look at the example shown in figure 1, vEdge-4 receives four OMP routes for prefix 172.16.50.0/24.

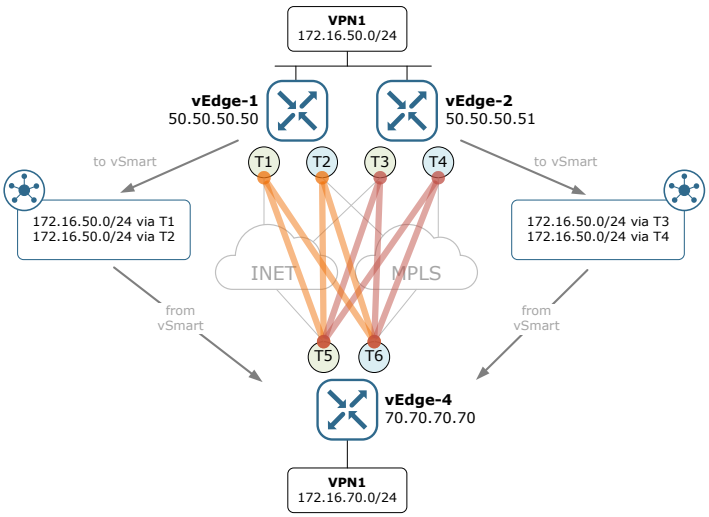


Figure 1. Traffic Engineering using TLOC Preference

The goal of this lesson is to create a Centralized Control Policy that will influence the TLOC preference values of vEdge-2's TLOCs. This will then influence the routing decision made by all spoke routers that have vRoutes pointing to vEdge-2's TLOCs.

OMP Best-Path Selection

As we have already learned, WAN edge routers advertise their local network routes to the vSmart controllers using OMP. Depending on the centralized control policy in place and the network topology, the vSmart controllers re-advertise these omp routes back to all other vEdge devices. Having in mind that most sites have multiple vEdge devices, a network can be advertised by multiple vEdge devices. Therefore, the other vEdge devices must perform the following Best-Path Selection algorithm in order to decide which omp route to use for the actual data forwarding:

0. The Best-Path selection algorithm is performed only against valid OMP routes. Invalid routes are ignored. A route must have a next-hop TLOC that is known and reachable to be valid.
1. Prefer ACTIVE routes over STALE routes. A route is ACTIVE when there is an OMP session in UP state with the destination TLOC.
2. Select the route with the lower administrative distance (AD) value.
3. If AD values are equal, select the route with the higher OMP route preference value.
4. If OMP preference values are equal, select the route with the higher TLOC preference value. (on vEdges only)

5. If the TLOC preference values are equal, compare the route type and select the route based on the origin in the following order:
    1. Connected
    2. Static
    3. EBGP
    4. OSPF Intra-area
    5. OSPF inter-area
    6. OSPF external
    7. iBGP
    8. Unknown
6. If everything up to this point is equal, select the route with the lower IGP metric.
7. If the origin and IGP metric are equal, select the route coming from the node with the highest router ID. (on vEdges only)
8. If the router IDs are the same, a vEdge router selects the OMP route that has the higher private IP address.

To enforce the desired traffic engineering (making traffic destined to 172.16.50.0/24 to prefer the routes via vEdge-2), we are going to manipulate step 4 of the OMP best path algorithm - we are going to increment the TLOC preference of both TLOCs of vEdge-2 and because all other values up to step 4 are equal, the OMP route from vEdge-2 will be preferred over the OMP routes advertised by vEdge-1.

Inbound Centralized Policy

With an inbound centralized policy, we can modify any given property of a particular vRoute before it goes into the routing table of the vSmart controller. This will then influence the best-path selection and lead to a different output from the best-path algorithm. Remember that the best routes are then advertised downstream to all WAN edge routers. Therefore, any manipulation of the controller's routing table will change the control-plane information across the whole overlay fabric.

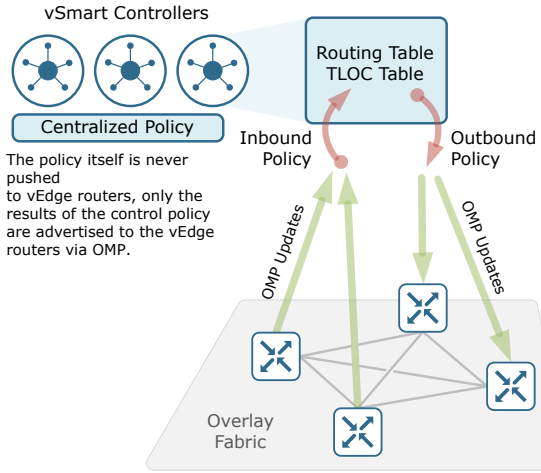
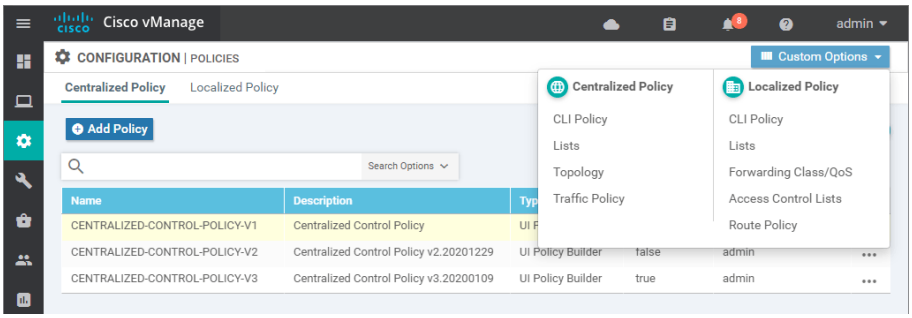


Figure 2. Cisco SD-WAN Inbound Centralized Policy

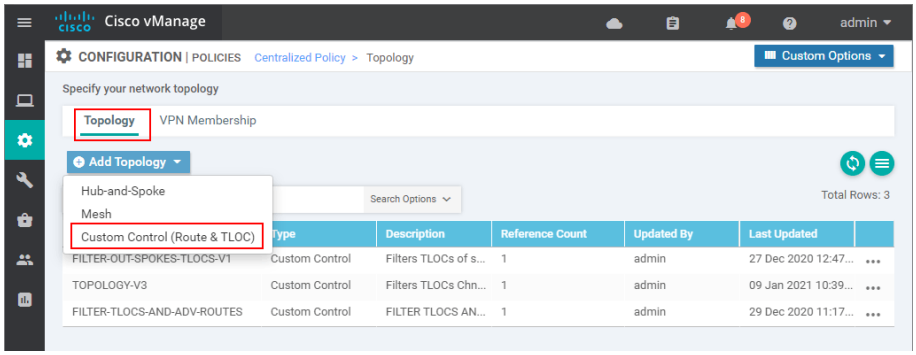
The goal of this lab is to make sure that traffic from all spoke vEdges enters the hub site through vEdge-2 instead of load-balancing between both vEdge-1 and vEdge-2. We are going to do this by creating an Inbound Centralized Policy that matches the TLOCs originated by the hub vEdges and modifying the TLOC preference values in the following manner. The TLOCs coming from vEdge-1 will be modified to have a TLOC preference of 100 and the ones from vEdge-2 will be modified to have a TLOC preference of 200. Note that the modification of this control plane information will take place before the TLOC routes are inserted into the vSmart's TLOC table.

Let's create a new Centralized Control Policy using the vManage GUI. We go to Configuration > Policies and from the dropdown menu at the right top corner we select Topology.



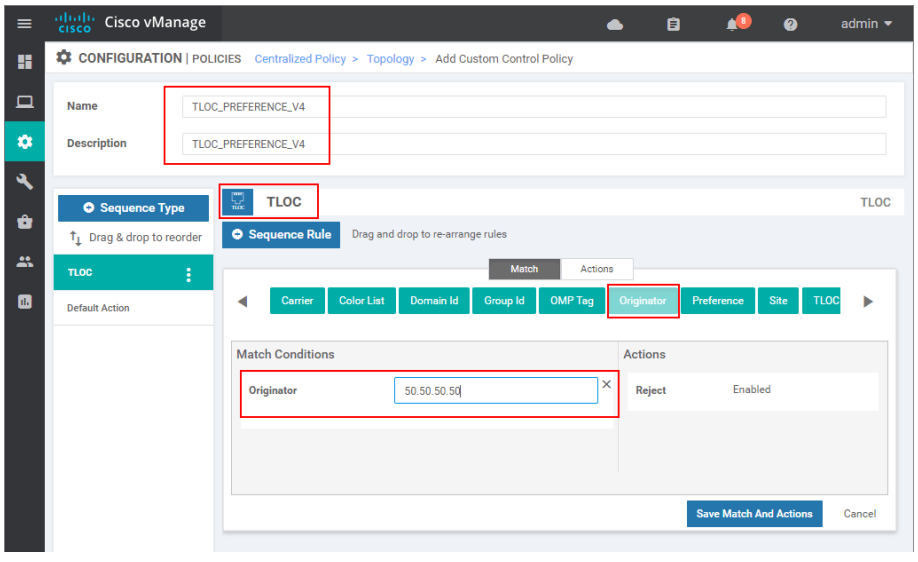
Creating an inbound Centralized Control Policy - step 1

There we are going to see the Topology policies we have created in the previous labs. To create another one we go to Topology > Custom Control (Route & TLOC).



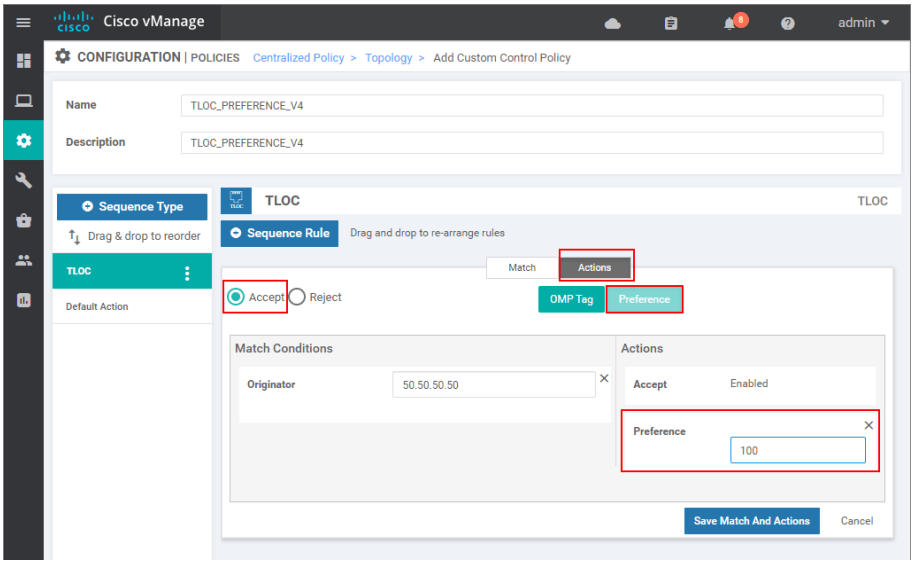
Creating an inbound Centralized Control Policy - step 2

For this policy, we are going to use the name TLOC_PREFERENCE_V4 and the same name for a description. Then we click the Sequence Type button to add a new sequence and select TLOC. Then we are going to match the TLOC routes coming from vEdge-1 by specifying the System-IP address of vEdge-1 as Originator.



Creating an inbound Centralized Control Policy - step 3

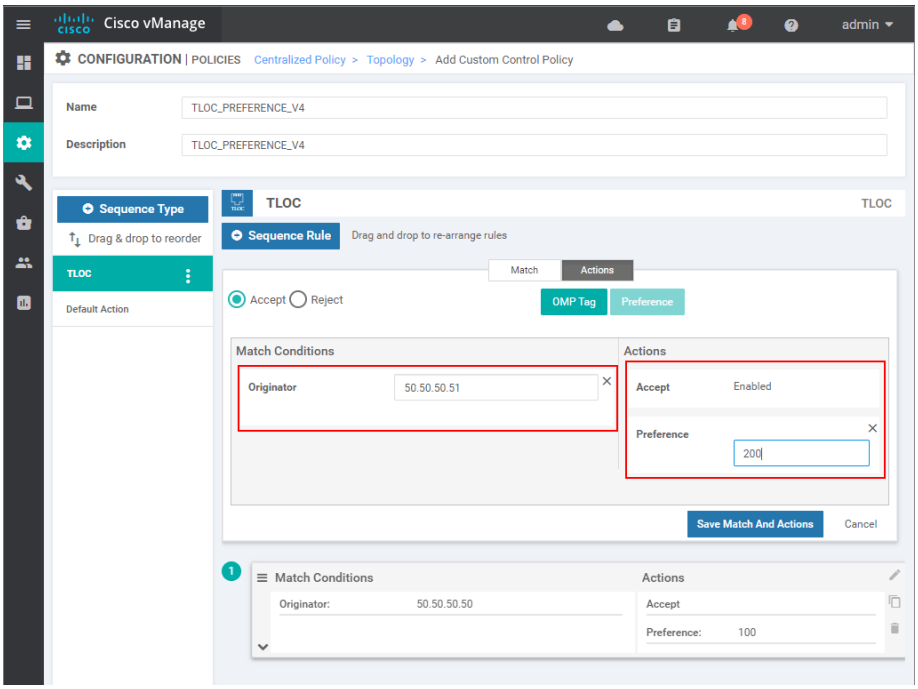
Then we go to the Actions tab and specify the action to be Accept. After that, we set the Preference value to be equal to 100.



Creating an inbound Centralized Control Policy - step 4

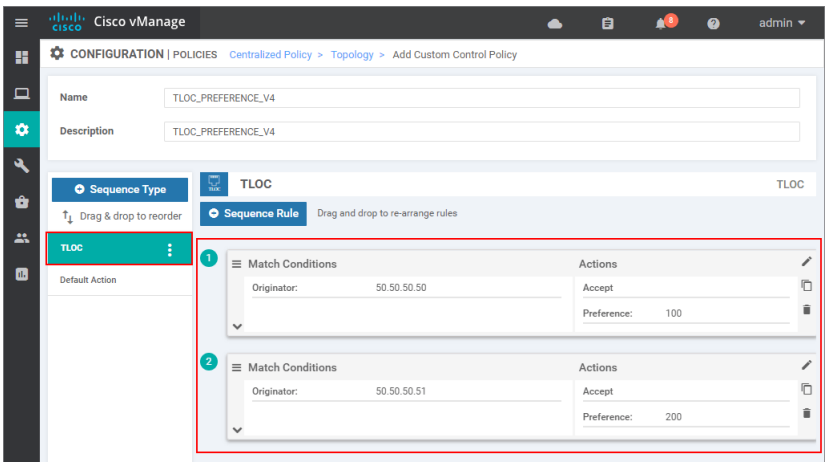
At this point, we have created a TLOC sequence that matches the TLOCs of vEdge-1 by using the Originator value of 50.50.50.50. Then we accept these TLOCs and set a preference value of 100.

Now we have to do the same for the TLOCs of vEdge-2 but this time we are going to specify a higher TLOC Preference value in order to make all WAN edge routers prefer vEdge-2 as an entry point for the hub site. We add another Sequence of type TLOC and specify the System-IP address of vEdge-2 50.50.50.51. Then on the Actions tab we accept these TLOCs and set the Preference to be 200 (higher than the one specified for vEdge-1's TLOCs)



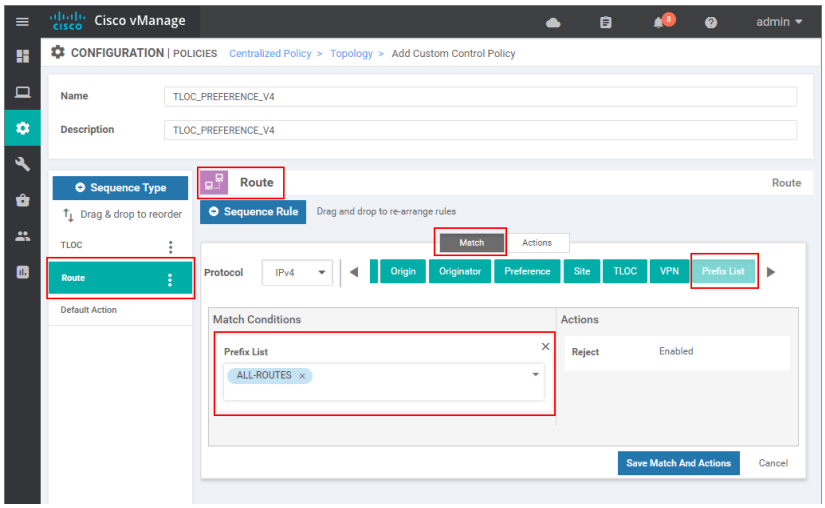
Creating an inbound Centralized Control Policy - step 5

At this point, we must have the following TLOC sequences as shown in the screenshot below:



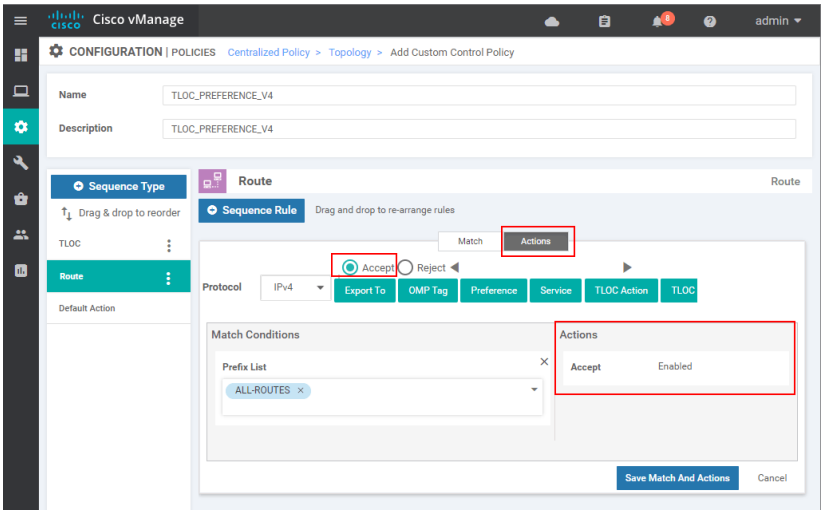
Creating an inbound Centralized Control Policy - step 6

Now we must add another sequence to the policy, but this time it would be of type Route. There we must match the pre-defined ALL-ROUTES prefix-list and then specify the action to be Accept. This basically tells the controller to accept all vRoutes because otherwise they will be rejected by the default action at the end which is Reject.



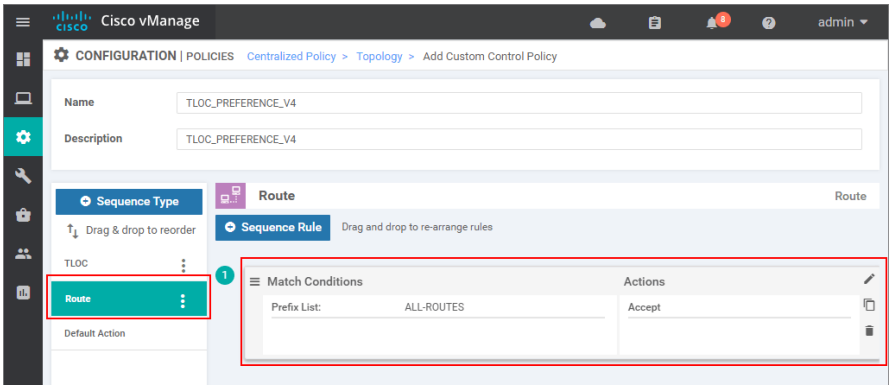
Creating an inbound Centralized Control Policy - step 7

On the Actions tab we click on the Accept radio button.



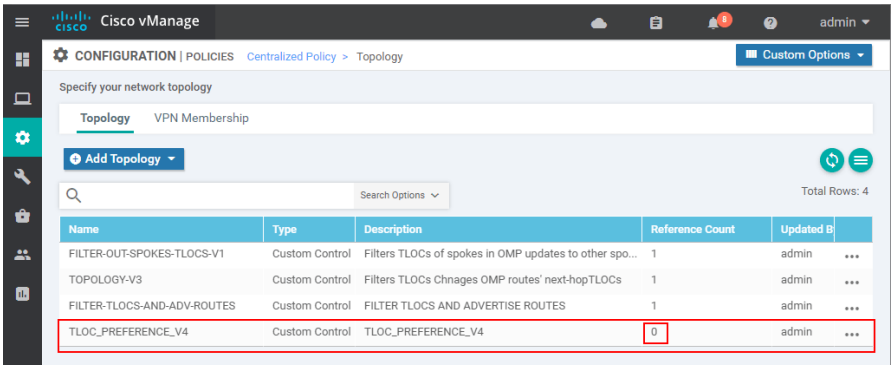
Creating an inbound Centralized Control Policy - step 8

In the end, we must have the following Route Sequence as shown in the screenshot below:



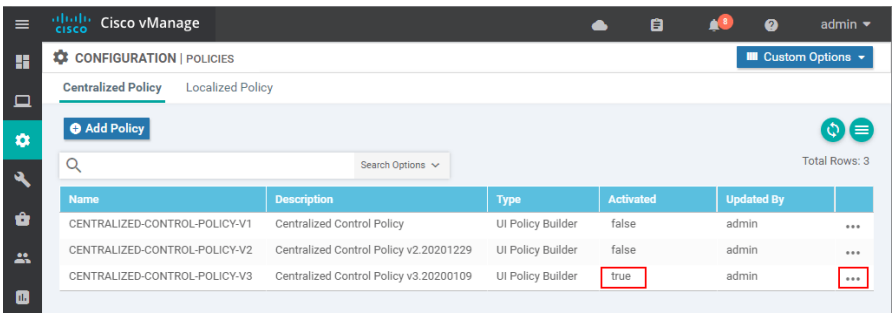
Creating an inbound Centralized Control Policy - step 9

Now our new topology policy is created. However, you can see that the Reference Count is still zero, which means that it is not used in any Centralized Policy at the moment.



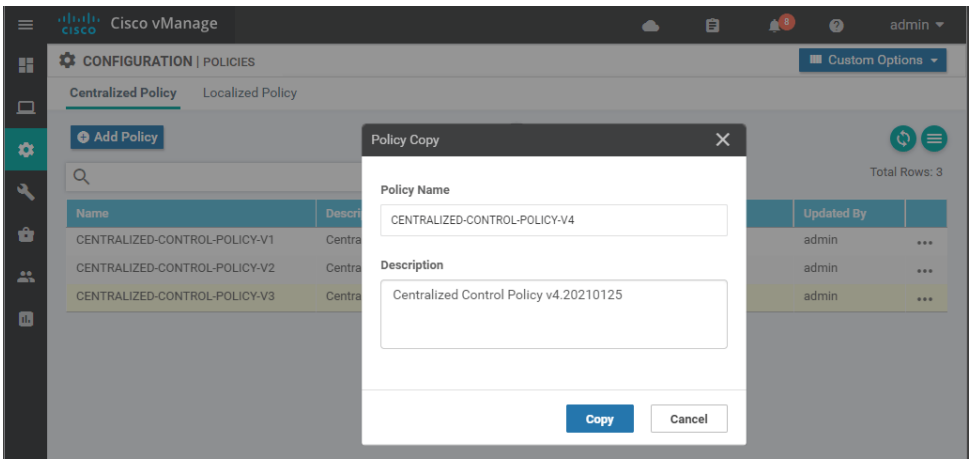
Creating an inbound Centralized Control Policy - step 10

If we go back to Configuration > Policies, we can see that the current active Policy is CENTRALIZED-CONTROL-POLICY-V3 (the one created in the previous lesson). As we have already said many times, we cannot edit a Centralized Policy while it is activated on the vSmart controllers and in effect. That is why we are going to just copy the policy into one with another name and will just change the Topology Definition, leaving all other settings as they are.



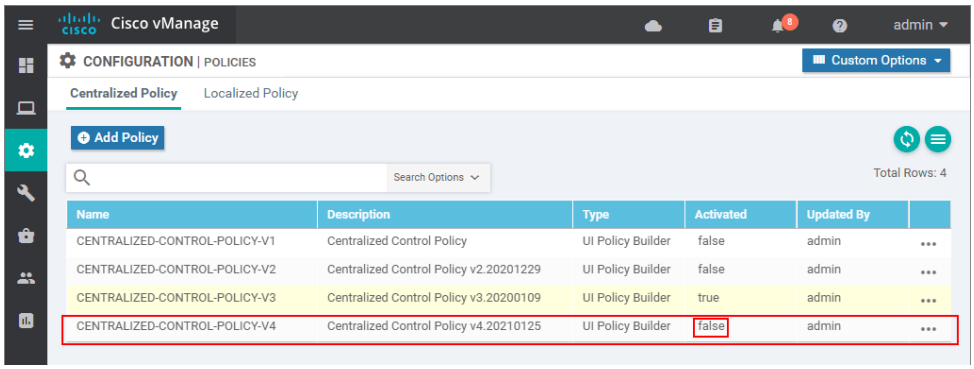
Creating an inbound Centralized Control Policy - step 11

We go to the more options button, select copy, and specify the name CENTRALIZED-CONTROL-POLICY-V4 for the new policy.



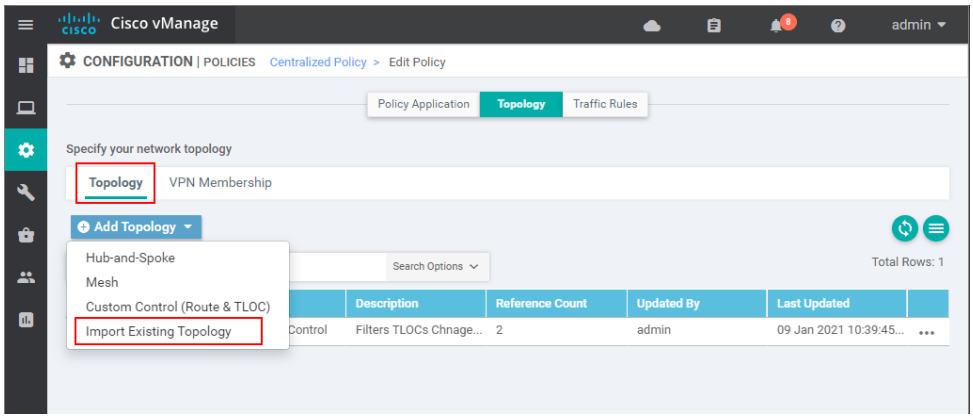
Creating an inbound Centralized Control Policy - step 12

Once it is created, you can see that it is still not Activated. We will need to edit it and point to the Topology definition that we have created.



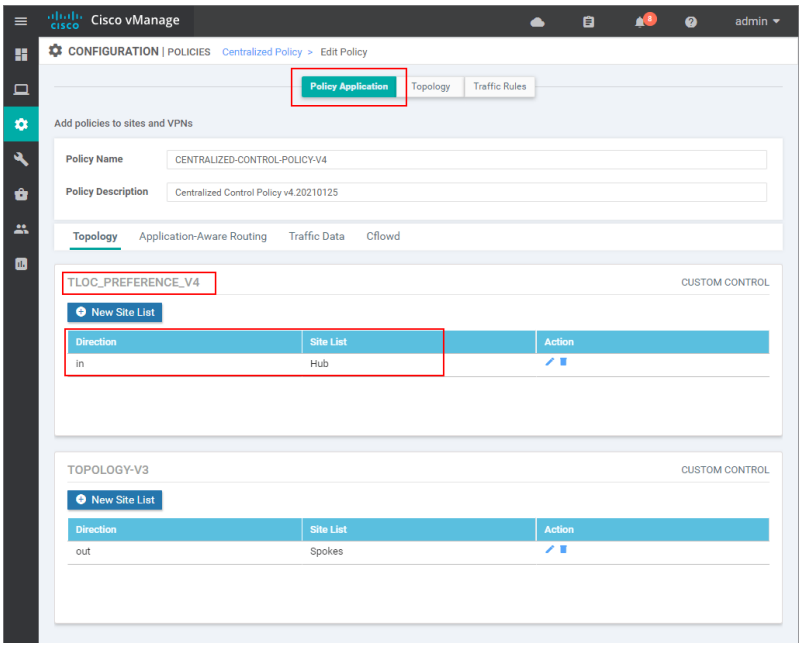
Creating an inbound Centralized Control Policy - step 13

We go to Edit > Topology and then on the Add Topology dropdown we select Import Existing Topology.



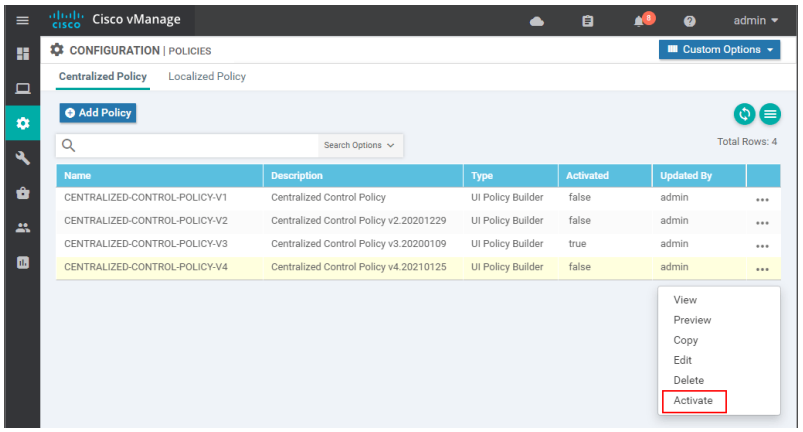
Creating an inbound Centralized Control Policy - step 14

In there, we select the one we created - TLOC_PREFERENCE_V4. After that, we go back to the Policy Application and apply the TLOC_PREFERENCE_V4 to site-list Hub in inbound direction as shown in the screenshot below:



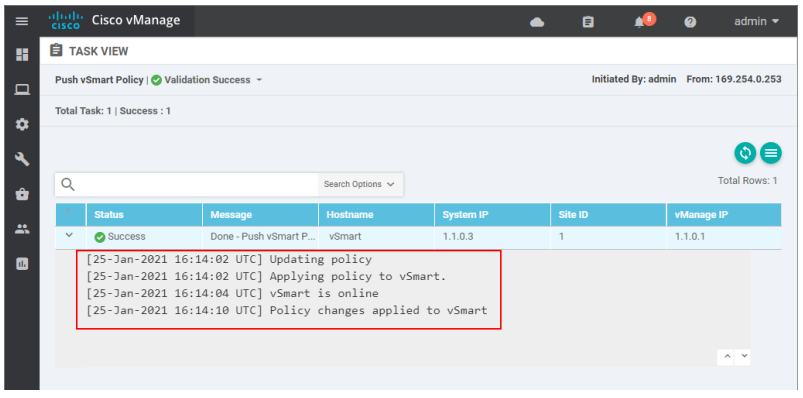
Creating an inbound Centralized Control Policy - step 15

Then we go back to Configuration > Policies and Activate the newest one named v4.



Creating an inbound Centralized Control Policy - step 17

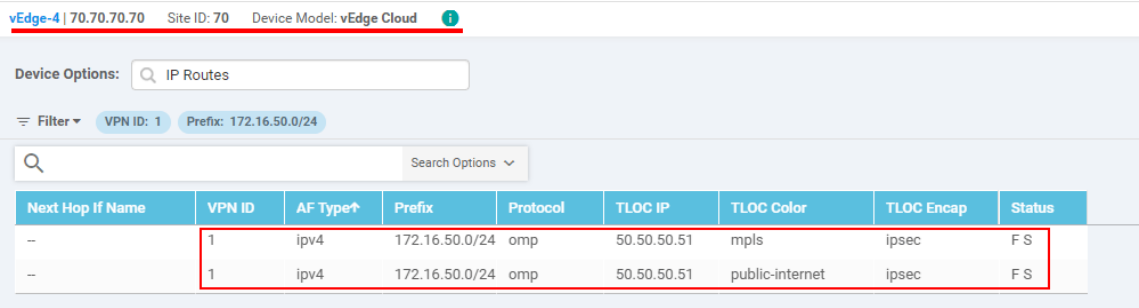
If everything has gone to plan, the activation should be successful.



Creating an inbound Centralized Control Policy - step 17

Verifications

Now if we check the routing information of vEdge-4 about the prefix 172.16.50.0/24, we can see that it prefers the omp routes that have a next-hop TLOC address of vEdge-2 because the vEdge-2's TLOCs have a higher preference that the ones of vEdge-1.



Next Hop If Name	VPN ID	AF Type	Prefix	Protocol	TLOC IP	TLOC Color	TLOC Encap	Status
--	1	ipv4	172.16.50.0/24	omp	50.50.50.51	mpls	ipsec	F S
--	1	ipv4	172.16.50.0/24	omp	50.50.50.51	public-internet	ipsec	F S

vEdge4 OMP route table after the inbound policy has been applied

LAB 5: Traffic Engineering - End-to-End Path Tracking

What is End-to-End Path Tracking?

As we have seen in the previous lessons, Centralized Control Policies allow us to design and configure traffic engineering. To understand what end-to-end path tracking is, let's first look at a simple TE use-case shown in figure 1 below. Suppose that we have a security stack hosted at site-3, and we want to redirect the traffic from Site-1 destined to Site-2 to go through the security stack. To engineer this traffic flow, we need to change the default routing behavior of the SD-WAN fabric, which would be to directly forward the traffic through the tunnel between site-1 and site-2 (T1-T2).

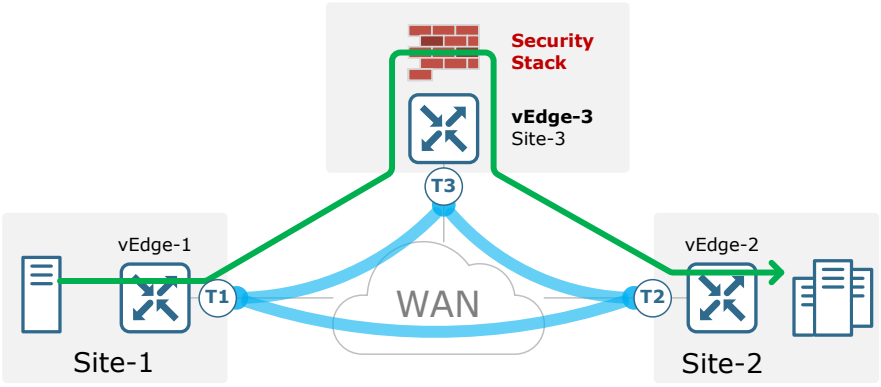


Figure 1. Traffic sourced at Site-1 destined to Site-2 goes through the intermediate router at Site-3

To redirect the traffic from Site-1 destined to Site-2 through Site-3, we need to provision two control policies, one for Site-1, where vEdge-1 is located, and a second one for Site-2, where vEdge-2 is located. The control policy for Site-1 would change the next-hop TLOC for the traffic destined to the vEdge-2 to tloc T3, and the control policy for Site-2 would change the next-hop TLOC for the traffic destined for Site-1 to tloc T3.

This traffic engineering policy would redirect the traffic from Site-1 destined to Site-2 to go through Site-3, regardless of whether the path between Site-3 and Site-2 is actually available. So when tunnel T3-T2 becomes unavailable, vEdge-1 won't know and will still send the traffic to vEdge-3, which will then drop the traffic because there is no path available toward vEdge-2. Figure 2 below illustrates this problem:

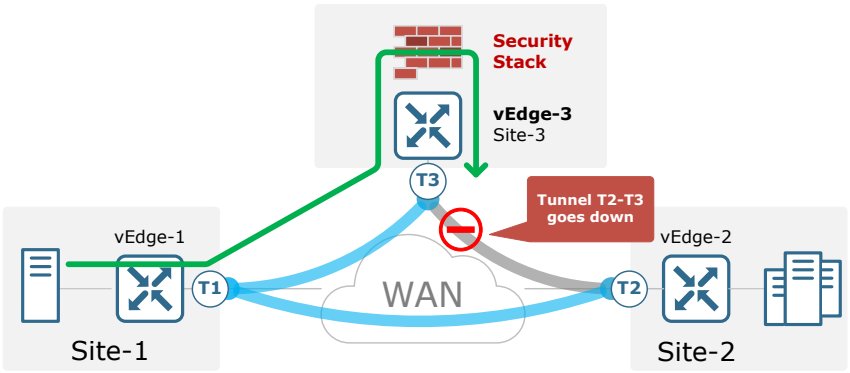


Figure 2. Tunnel T2-T3 goes down

Enabling the End-to-End Path Tracking feature would allow vSmart to monitor the path to the ultimate destination (vEdge-2), and to inform the source router (vEdge-1) when that path between vEdge-3 and vEdge-2 (tunnel T3-T2) is not available. The source vEdge-1 can then remove the path from its route table and route the traffic through the second-best path (tunnel T1-T2).

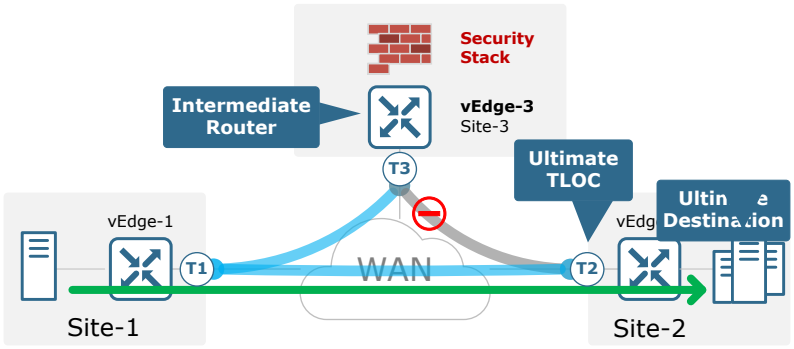


Figure 3. End-to-End Path Tracking

TLOC Action

End-to-end Path Tracking can be achieved by using four different TLOC action options as you can see in the CLI output below:

```
vSmart(config-sequence-1)# action accept set tloc-action ?
Description: Action to be taken with ultimate specified TLOC or service
Possible completions:
backup  ecmp  primary  strict
```

Strict Option (Default option)

In normal circumstances, the communication between vEdge-1 and vEdge-2 goes through vEdge-3 which is an Intermediate Router. If the overlay tunnel between T3 and T2 goes down, vEdge-1 drops the traffic.

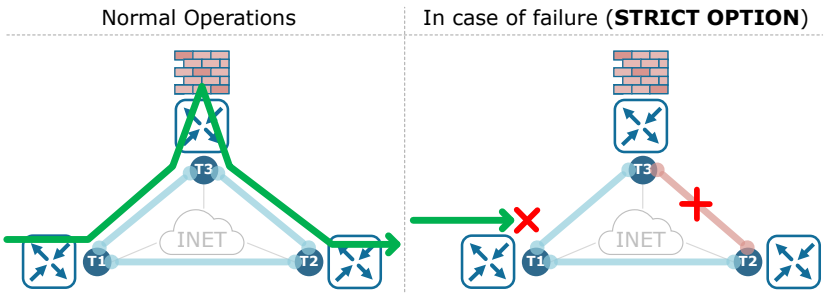


Figure 4. TLOC Action - Strict

This option is useful in use cases where security (or another network service) is more important than availability. If the traffic could not go through the intermediate router and subsequently through the security stack, it'd better get dropped.

Primary Option

In normal circumstances, the communication between vEdge-1 and vEdge-2 goes through vEdge-3 which is an Intermediate Router. If the overlay tunnel between T3 and T2 goes down, vEdge-1 would forward the traffic directly to Site-2 via tunnel T1-T2.

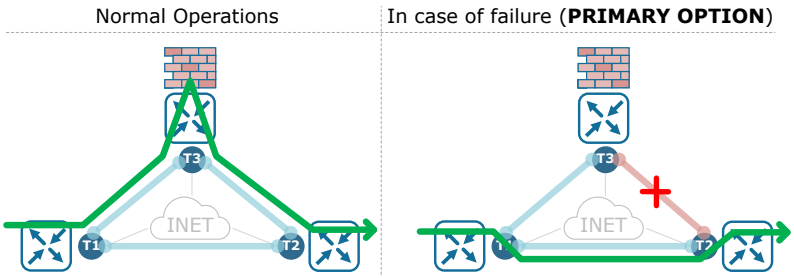


Figure 5. TLOC Action - Primary

This option is useful in use cases where availability is more important than security (or another network service). If the traffic could not go through the intermediate router and subsequently through the security stack, it will be forwarded directly through the T1-T2 tunnel without going through the network service.

Backup Option

In normal circumstances, the communication between vEdge-1 and vEdge-2 would not go through the Intermediate Router. If the overlay tunnel between T1 and T2 goes down, vEdge-1 will forward the traffic through the intermediate router.

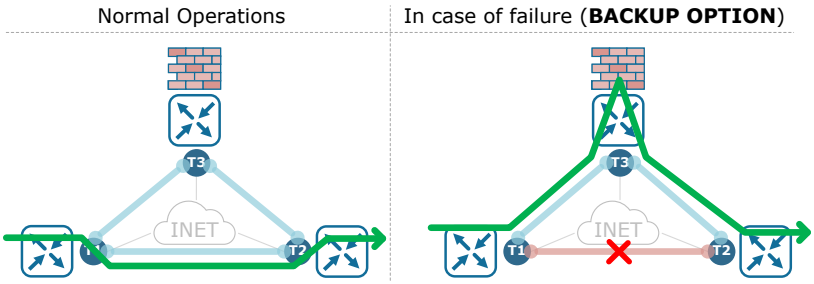


Figure 6. TLOC Action - Backup

ECMP Option

In normal circumstances, the communication between vEdge-1 and vEdge-2 would be load-balanced through the Intermediate Router and through the direct tunnel T1-T2 as well. If the overlay tunnel between T1 and T2 goes down, vEdge-1 will continue forwardingtraffic through the intermediate router.

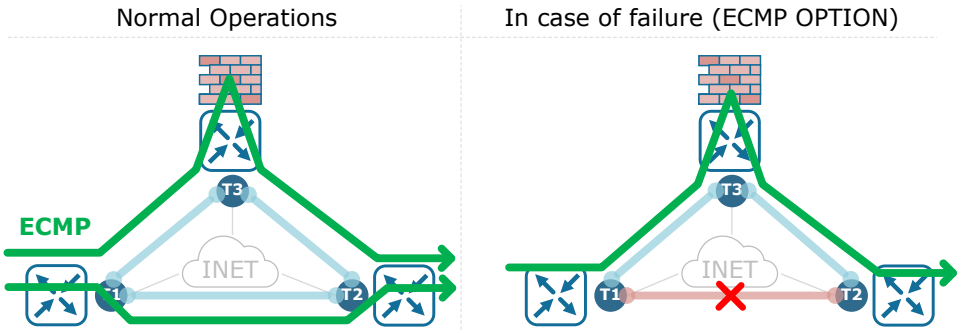


Figure 7. TLOC Action - ECMP

Configuring End-to-End Path Tracking

To demonstrate the End-to-End path tracking feature, we are going to set up a simple topology as shown in figure 8 below:

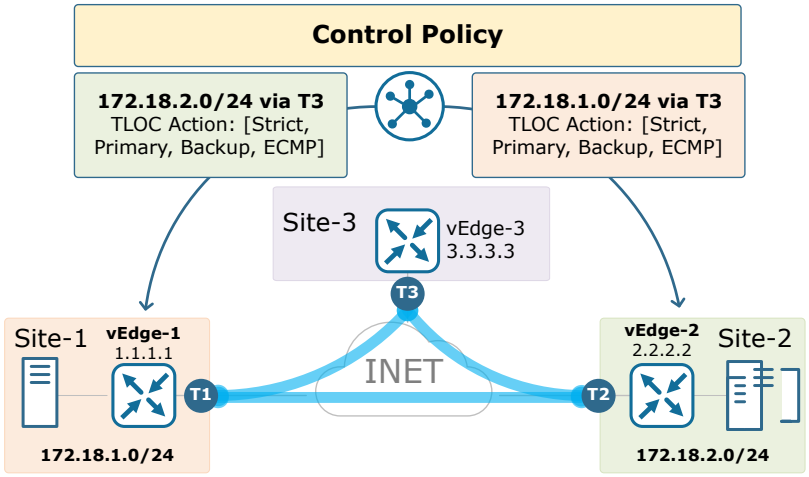


Figure 8. End-to-End Path Tracking Configuration Diagram

For a starting point, let's say that all vEdges are configured as shown on the diagram and there is no policy applied on vSmart at all. Router vEdge-1 will advertise the prefix 172.18.1.0/24 with next-hop T1 and vEdge-2 will advertise the subnet 172.18.2.0/24 with next-hop T2. Therefore, the traffic between 172.18.1.0/24 and 172.18.2.0/24 will go through the direct overlay tunnel T1-T2.

To change that, we are going to configure two control policies on vSmart that will change the next-hop TLOC for both subnets to be the tloc T3 of vEdge-3. This will make vEdge-3 an intermediate router for the traffic between 172.18.1.0/24 and 172.18.2.0/24.

First we will need to configure two site lists for site-1 and site-2 and two prefix lists for subnet 172.16.2.0/24 and 172.16.2.0/24 respectively. This config is highlighted with green in the output below. Once the lists are defined, we will use them in two control policies called OUTBOUND-TO-SITE-1 and OUTBOUND-TO-SITE-2. The first policy will match prefix list PREFIX-1 and will set the next-hop TLOC to tloc T3 (3.3.3.3, mpls, ipsec). We are going to do the same for the other prefix PREFIX-2. In the end, control policy OUTBOUND-TO-SITE-1, as the name implies, will be applied to site-list SITE-1 in an outbound direction. We do the same for control policy OUTBOUND-TO-SITE-2. Control policies config is highlighted with yellow and then the policies are applied with the config highlighted with orange.

Lab 6: VPN Membership Policy - Isolating guest users

The Business Need

It is pretty common that an organization has branches where guest users are permitted to connect to the network and provided Direct Internet Access (DIA). As a rudimentary security measure, the guests can be confined within their own VPN. However, by default in Cisco SD-WAN, any-to-any connectivity within a single VPN is automatically established between all sites through the exchange of OMP routes and TLOCs. In most enterprise VPNs, this automatic any-to-any behavior is desired. However, most organizations would not want to allow guests to communicate with other guests across the overlay fabric as shown in figure 1 below.

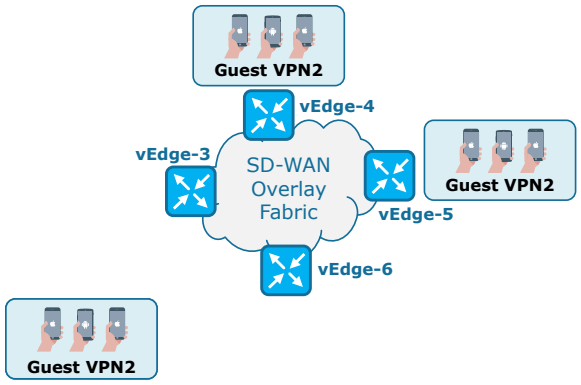


Figure 1. By default, any-to-any connectivity is established within each VPN.

VPN Membership policies are the tool that allows network administrators to prohibit the exchange of control plane information for particular VPNs and subsequently to isolate the users within a VPN from the SD-WAN fabric.

What is a VPN Membership policy?

A VPN membership policy is a special type of centralized control policy that is used to control what VPN routing tables are distributed to which particular vEdge routers. In a default SD-WAN fabric with no VPN membership policy applied, the vSmart controller advertises all OMP routes for all VPNs to all vEdge routers. However, if an organization wants to restrict the participation of specific vEdge routers in particular VPNs, a VPN Membership policy that enforces this restriction is applied to vSmart.

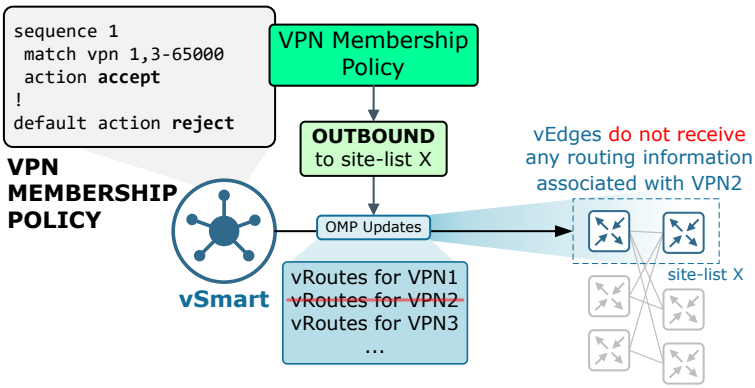
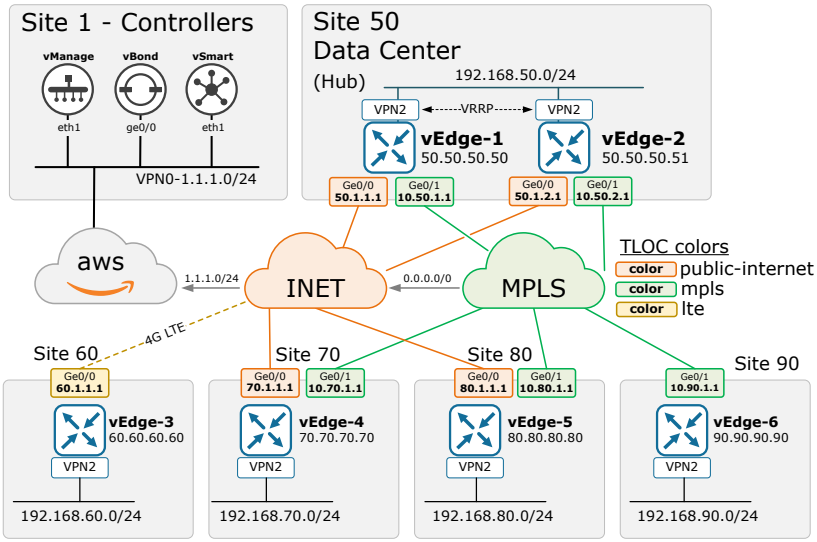


Figure 2. VPN Membership Policy

An important point to notice is that we do not set a direction when applying a VPN membership policy. This type of policy is automatically applied in an outbound direction from the perspective of the vSmart controller and affects the OMP updates sent from vSmart to the vEdge routers.

Configuring VPN Membership policies

For this lab example, we are going to use the same lab topology that we have been using through all lessons for centralized control policies. For a guest segment, we are going to use vpn_id 2.

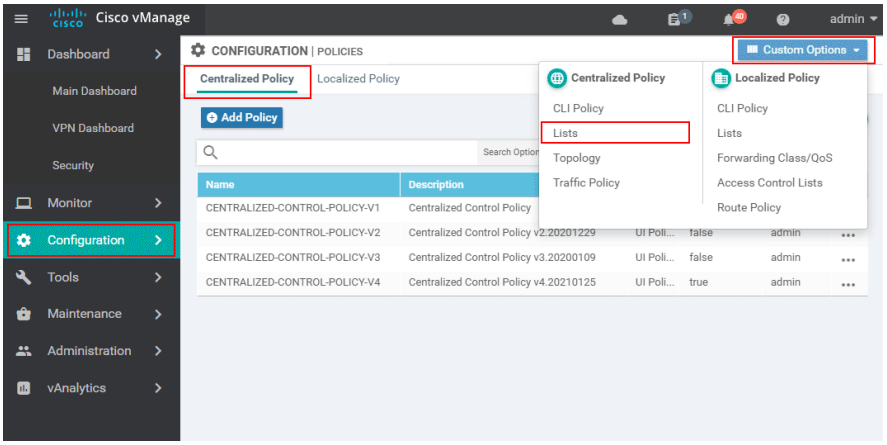


VPN Membership Policy Lab Diagram

As you can see in the lab topology, there is one directly connected prefix in VPN 2 on each site. If we check the routing table on any of the branch vEdge routers, we are going to see that the vSmart controller has advertised all prefixes to all routers and every vEdge knows about all networks in the guest segment.

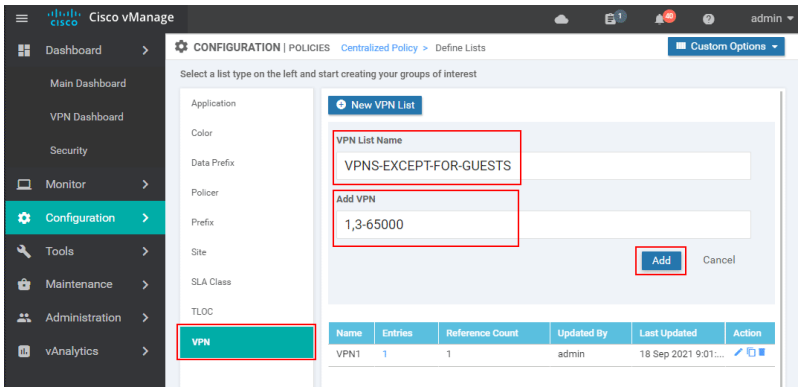
However, in order to isolate the guest users from communicating over the SD-WAN overlay fabric, we are going to configure a VPN Membership policy that will prohibit the vSmart controller from advertising any OMP routing information associated with vpn_id 2 to the branch sites.

The first thing that we have to do is to create a vpn-list that matches the vpn-ids of all VPNs that we want to be permitted to communicate across the overlay fabric. As we have only got one guest segment with id 2, the vpn-list will match vpn_ids 1,3-65000 (practically every single segment except for the guest one). Let's create the list by going to Configuration > Policies > Custom Options > Lists as shown in the screenshot below:



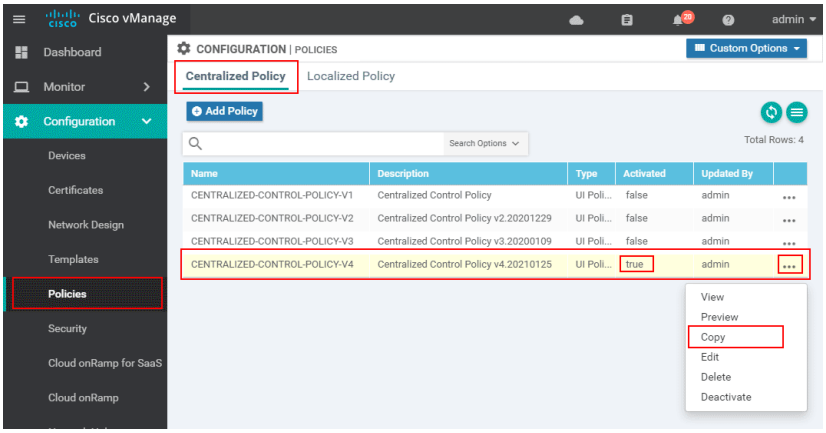
Create a new policy list

Then we go to VPN lists and create a new one named VPNS-EXCEPT-FOR-GUESTS that includes ids 1,3-65000.



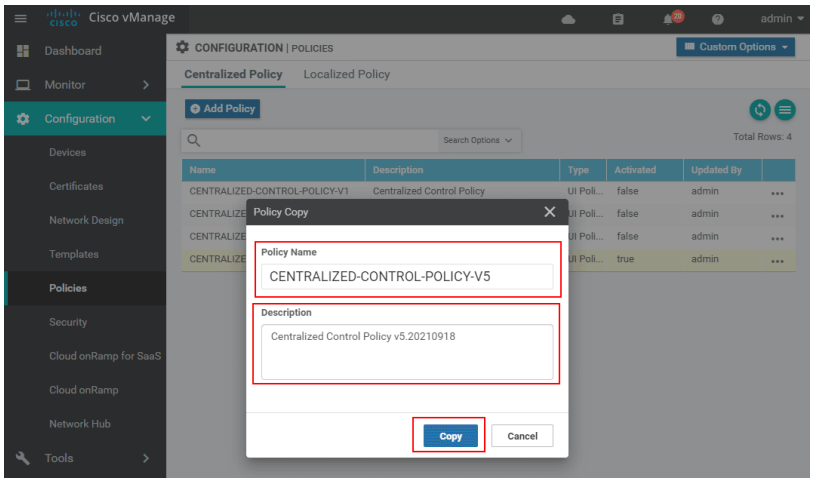
Define a new VPN list

Once the list is configured, we need to copy the latest active control policy that is applied. We go to Configuration > Policies > Centralised Policy and click the more options. In there we select Copy.



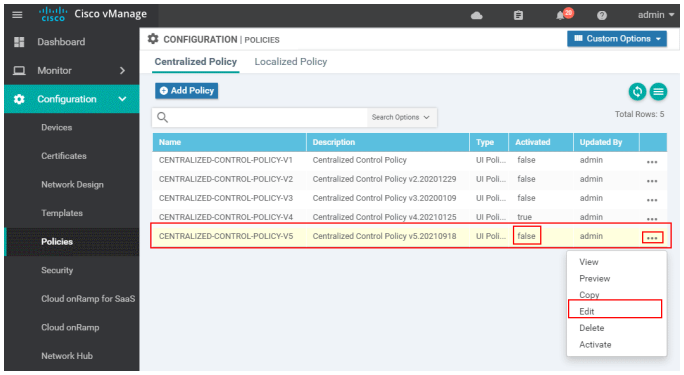
Copy the existing Control Policy

As a general rule of thumb, we included a version number in the name of the new policy - CENTRALIZED-CONTROL-POLICY-V5, and in the description, we write a timestamp in cleartext.



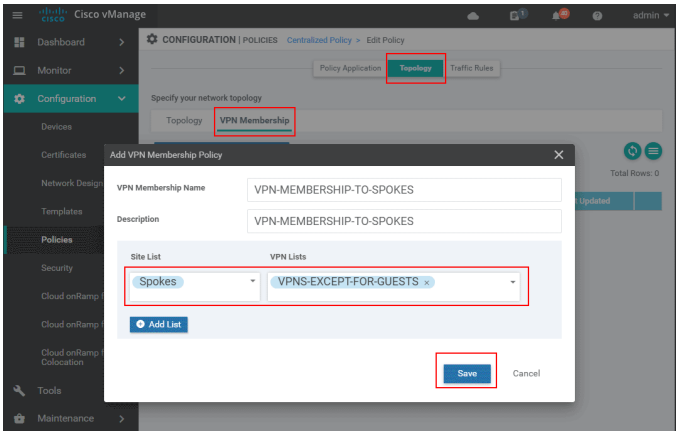
Set up a new name for the control policy

You can see the new policy is created but is not activated yet (The activated value is false). Before we push it to vSmart, we need to edit it and create a new VPN Membership policy that achieves the objectives of this lab example. We go to the more options button and select Edit, as shown in the screenshot below:



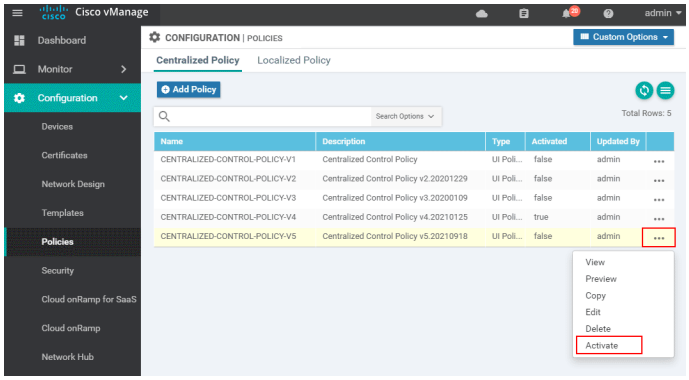
Edit the new policy

Once we enter the Edit Policy menu, we go to the Topology tab and then to the VPN Membership subtab. In there, we specify a name and description of the VPN membership policy. Then we select a site-list that identifies which WAN edge routers will be affected by the membership policy and a vpn-list that identifies which VPN routing tables will be sent to these vEdges. In our example, we are going to apply the policy to site-list SPOKES and will use the vpn-list that we have created earlier in the lesson.



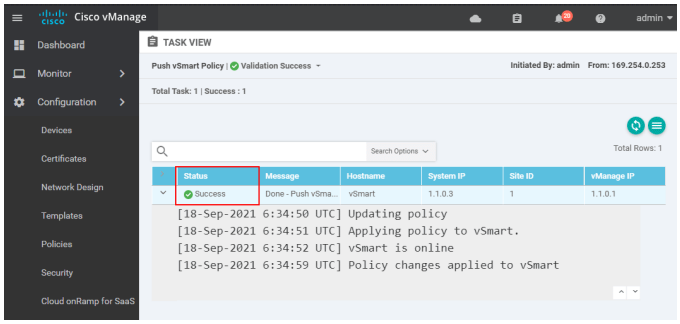
Add a new VPN Membership Policy.

Once the VPN Membership policy is configured, we can go ahead and activate the latest version of the Centralized Control Policy. This is done by going to the more options and selecting Activate as shown below.



Activate the new control policy

Once the policy is activated, vManage pushes it to the vSmart controller using a NETCONF transaction. If the policy is successfully applied, it becomes part of vSmart's running configuration, and vManage lets us know that the activation is successful.



The new policy is successfully activated.

Checking the results

Before we applied the VPN Membership policy, we had checked the VPN-2 routing table of vEdge-4 and saw that the router knew about every network within the guest segment. However, if we check the routing table now, we are going to see that the router has not received any OMP routes for VPN 2 at all. It only knows about its own directly connected subnet at the moment.

9. CENTRILIZED DATA POLICIES

What is a Centralized Data Policy?

A Centralized Data Policy allows network administrators to override the normal forwarding decisions that would occur at specific WAN edge routers and define a different set of actions that would be performed instead. It is provisioned through the vMange GUI and pushed to the vSmart controllers (hence, it is centralized). Data policies are applied to traffic flows throughout VPNs in the overlay network. They can permit and deny traffic based either on a 6-tuple match (src IP, dst IP, src port, dst port, DSCP value, and protocol type) or on VPN membership. An important point to understand is that a centralized data policy acts on an entire VPN and is not interface-specific (localized data policies control the flow of packets in and out of specific interfaces).

By default, no centralized data policy is provisioned in a Cisco SD-WAN solution. Therefore all prefixes within every VPN are reachable from anywhere within the VPN. When we want to restrict access between specific sources and destinations within the VPN, we use a centralized data policy that filters traffic based on a 6-tuple.

A centralized data policy can be applied in three different modes from the perspective of a WAN edge router:

- From-Service (Upstream)
- From-Tunnel (Downstream)
- All (Upstream and Downstream)

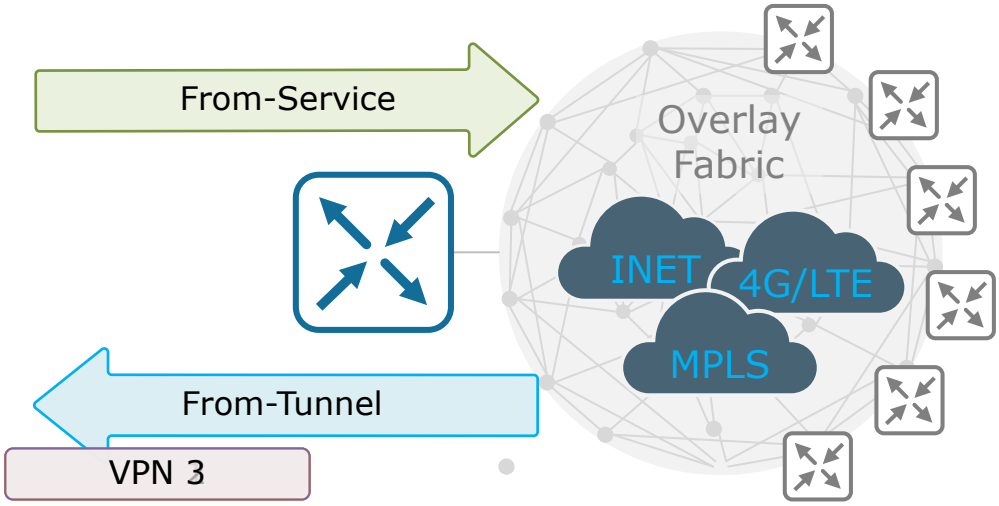


Figure 1. Data Policy Direction

This allows for a different policy to be applied to the same site list but in a different direction.

```
apply-policy site-list <name>
  data-policy <name> all | from-service | from-tunnel
```

Applying a policy

A data policy itself is never pushed to the WAN edge routers in the overlay fabric. vSmart controllers actually push the results of the data policy via OMP and the effects of the policy are reflected on the WAN edge devices.This concept is visualized in figure 2 below:

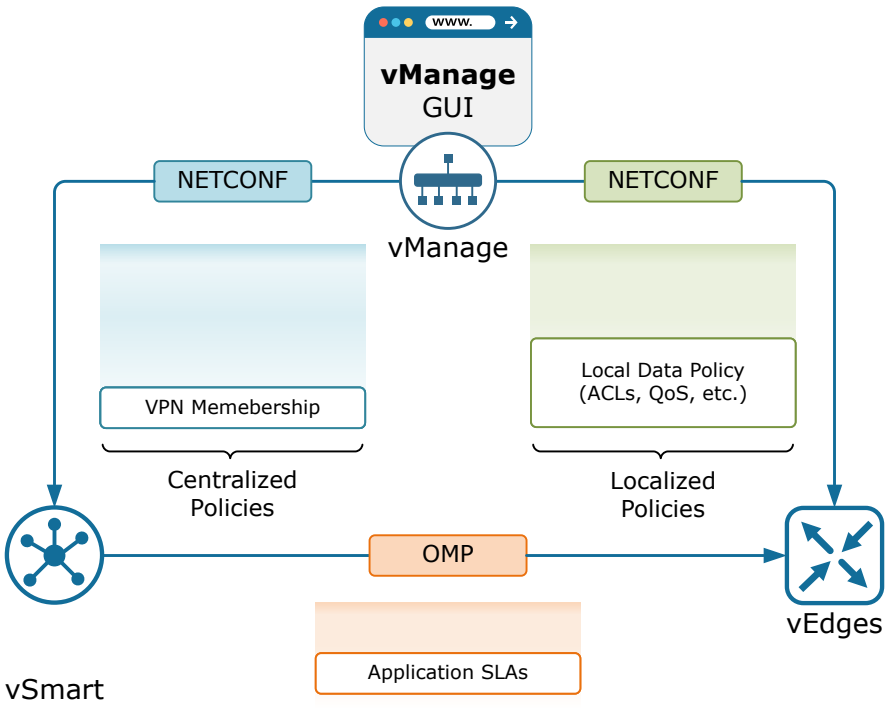


Figure 2. Provisioning a data policy

Data Policy Configuration Components

Centralized data policies are configured through the Cisco vManage policy wizard. It guides you through four sequential screens that define different parts of the policy construct:

- Create Groups of Interest — At this step, we create lists of interesting items that will later be called in the match or action statements in the policy.
- Configure Traffic Rules — At this point, we specify the match and action conditions.
- Apply Policies to Sites and VPNs — At this step, we associate the policy with a site list and VPNs.

The following figure 3 illustrates the components of a centralized data policy:

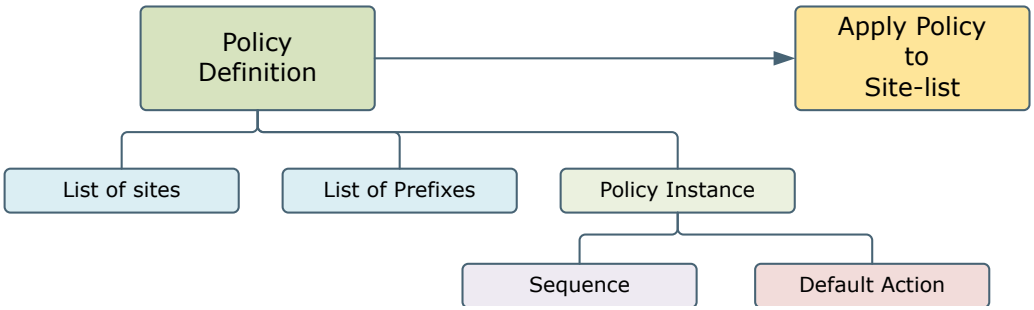


Figure 3. Data Policy Configuration Components

Use cases

These types of policies are typically used to manipulate traffic flows across the overlay fabric. Here are some typical use-cases that we are going to configure and analyze in the next couple of lessons:

- Permit/Deny a set of sources to send traffic to any destination outside the local site. For example, allowing some host to communicate only with other hosts that are within the local site;
- Modify the next-hop of a set of sources when sending traffic to a specific set of destinations outside the local site. For example, traffic steering voice traffic over one path and data traffic over another of specific hosts within the local site;
- Permit/Deny a set of source IP addresses and ports to send traffic to any destination outside the local site;
- Permit/Deny a set of source IP addresses and ports to send traffic to a specific port at a specific destination outside the local site.

10. APPLICATION-AWARE ROUTING POLICIES

Configuring Application-Aware Routing (AAR) Policies

In Traditional WAN networks, routers make forwarding decisions based primarily on link-state and routing metrics. Cisco SD-WAN allows us to configure SLA-based routing that considers performance characteristics such as packet loss, latency, and jitter when making forwarding decisions. This is done using Application-aware Routing policies (App-route). However, AAR policies are one of the hardest topics of Cisco SD-WAN to learn because they represent a technique to route data traffic across the WAN that network engineers have typically not used before. That is why in this lesson we are going to deep-dive into the different App-route policy actions that Cisco SD-WAN offers as of the latest release (20.6.1).

The structure of an App-route Policy

There are four key parts to the Application-aware Routing policy process:

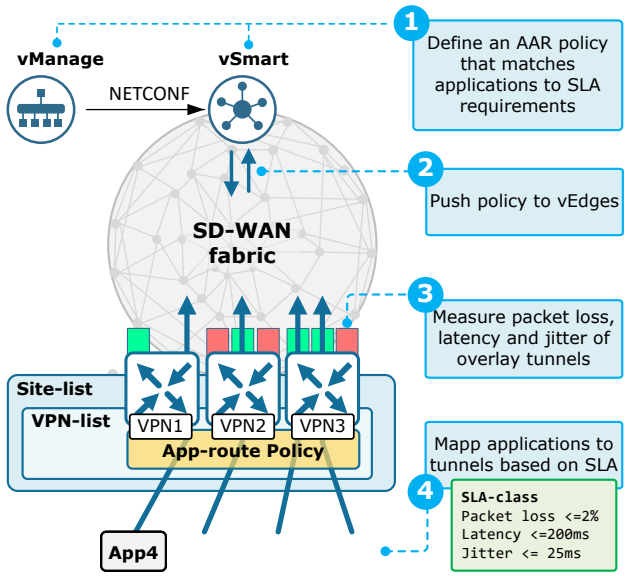


Figure 1. How an Application-aware Routing Policy Works

Step 1: Creating the App-route policy - As with all other centralized policy types, the first thing is always to define the policy via vManage's GUI or for practicing purposes directly through the CLI on vSmart. The basic building blocks of an application-aware routing policy are as follow:

1. VPN-list that specifies which vpn-ids the policy will affect;
2. App-list that matches the applications of interest;
3. App-probe-class that specifies the BFD dscp value to be used when probing the WAN (optional);
4. SLA-class that specifies the maximum packet loss, latency, and jitter;
5. The match-action rules of the App-route policy itself;
6. A site-list that specify which sites will receive the AAR policy;

Step 2: Pushing the policy down to vEdges - Once the App-route policy is defined and activated, vManage pushes it to the vSmart controller. The policy becomes part of vSmart's running-config. vSmart then sends the policy via OMP updates to all vEdges that are matched by the site-ids listed in the applied site-list. The Edge routers that receive the policy execute it in memory, which means that they do not store the app-route policy in their permanent configuration files. Upon reboot, each vEdge needs to first establish the control-plane connection to the SD-WAN controllers and receive the latest applicable version of the current AAR policy.

Step 3: Measuring the performance of overlay tunnels - In Cisco SD-WAN, once a vEdge router establishes an overlay tunnel, it automatically starts a BFD session with the remote peer. This behavior cannot be disabled or changed. The WAN edge router then uses this BFD session to monitor the link-state of the tunnel and also measure performance characteristics such as packet loss, jitter, and latency;

Step 4: Mapping applications to overlay tunnels that meet SLA - The BFD reports the performance metrics of each overlay tunnel to the applied AAR policy. The policy evaluates these metrics against the configured SLA threshold and makes SLA-based forwarding decisions.

Understanding the AAR policy actions

One of the most difficult things when it comes to Application-aware Routing policies is to understand the different actions that a WAN edge router could take in case of an out-of-threshold condition.

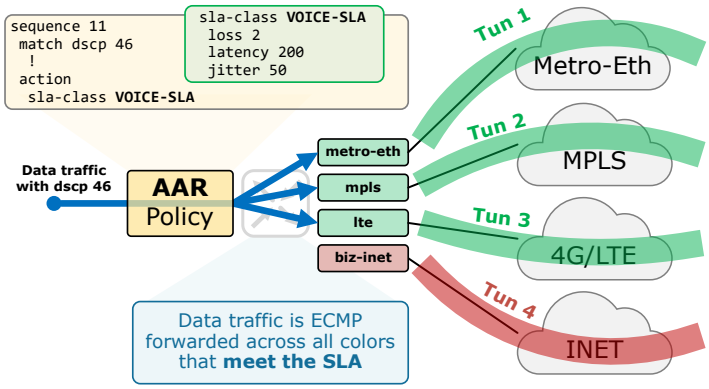


Figure 2. App-route action: SLA-class

The simplest action that we can configure in an AAR policy sequence is an SLA-class with no additional parameters as highlighted in the output below. The data traffic that matches the sequence will be ECMP-forwarded across all overlay tunnels that match the SLA class thresholds.

If no overlay tunnel matches the SLA thresholds, the data traffic is ECMP-forwarded across all available tunnels.

Preferred-Color

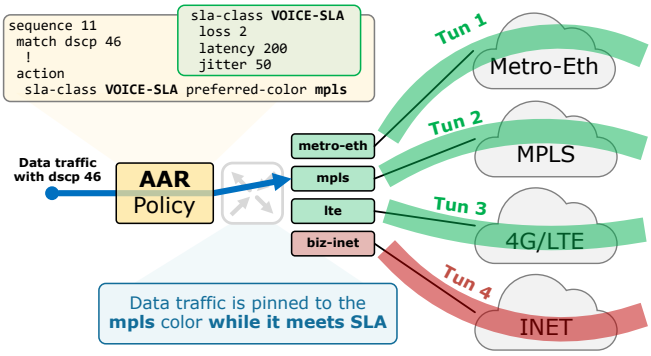


Figure 3. App-route action: Preferred Color

In cases where multiple overlay tunnels meet the SLA thresholds, we may want to specify preferred tunnels. This is done using the preferred-color [color] parameter. When this option is used, the software first tries to forward the data traffic through the preferred color, if it meets the SLA. If the preferred tunnel does not meet the SLA, the data traffic is sent through any overlay tunnel that matches the SLA class.

If no tunnel meets the SLA, data traffic is ECMP-forwarded across any available tunnel. Notice that the preference-color option is a loose matching. This means that the data traffic is always forwarded, no matter whether the preferred-color tunnel meets the SLA or not.

Multiple Preferred-Color

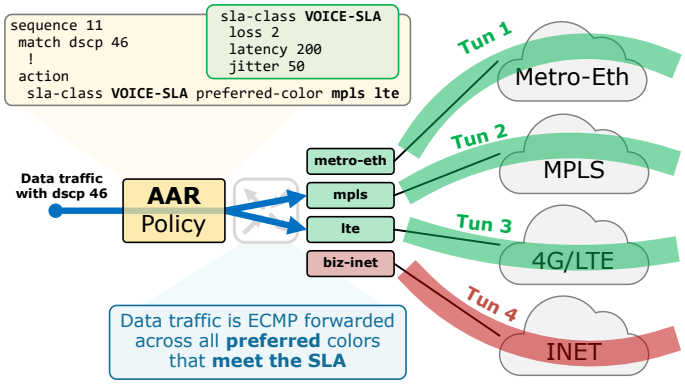


Figure 4. App-route action: Multiple Preferred colors

We can specify more than one preferred color as shown in the output below. When there are multiple preferred colors configured, the software tries to load-balance the data traffic across all preferred colors that meet the SLA.

Again, if no preferred color meets the SLA, the data traffic is ECMP-forwarded across all available tunnels.

Strict

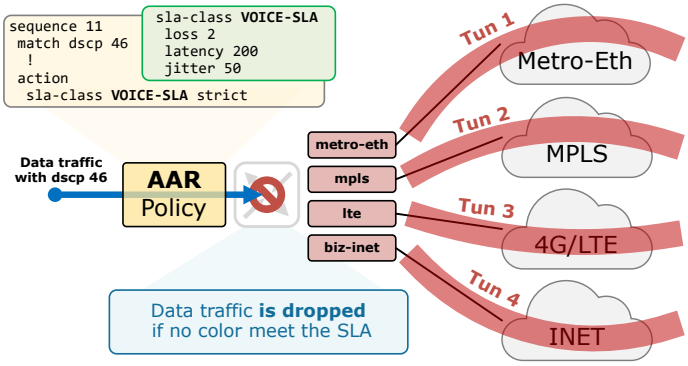


Figure 5. App-route action: Strict

Many network engineers ask themselves the question - Why would anybody need to drop traffic when the latency of the WAN jumps above the threshold? Well, there are many real-time systems where the timely delivery of data is critical. For example, radars and air-traffic surveillance systems make calculations based on the current position of an airplane in space. However, an airplane flies at a speed of 1000km/h. If the exact coordinates of a plane are received with 0.5 sec delay, they are useless, because the plane is not there anymore. Furthermore, when the system receives coordinates of the same plane via multiple radars, out-of-sync data could actually harm more than help. Therefore, for such systems, it is better to not receive the data when the latency of the WAN jumps above the threshold. That's one example of why this option exists. However, 99.9% of the time, it won't be used.

Fallback-to-best-path

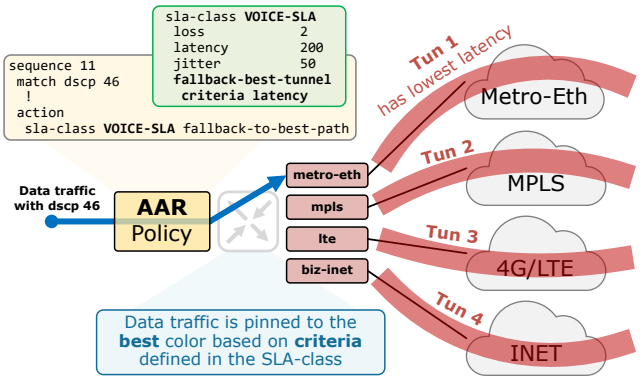


Figure 6. App-route action: Fallback to best color

Backup-sla-preferred-color

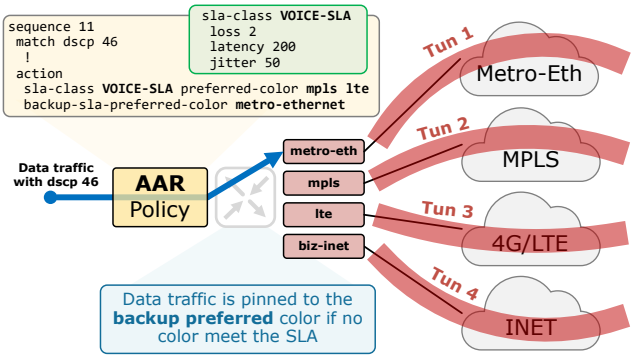


Figure 7. App-route action: Backup SLA preferred color

The backup preferred color is a parameter that tells the WAN edge router where to forward the data traffic when no tunnel meets the SLA thresholds. Similar to the preferred-color action, the backup-SLA-preferred-color is considered to be a loose matching which means that if the configured color is not available, the router will forward the traffic through any available color.

The main difference between preferred-color and backup-sla-preferred-color is that:

preferred-color – matched data traffic is pinned to a transport color (or colors) as long as the overlay tunnels meet the SLA thresholds;

backup-sla-preferred-color – matched data traffic is pinned to a transport color ONLY when no tunnel meets the SLA thresholds.

Notice that in a single match-action rule, we can't configure both the strict and backup-sla-preferred-color actions.

Default Action

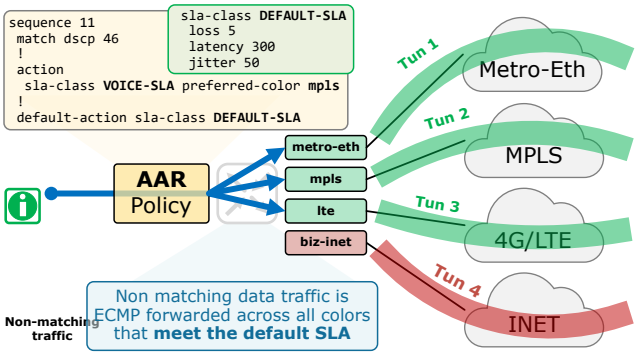


Figure 8. App-route action: Default action

Unlike centralized data policies where the default action is to drop all packets, the default action for application-aware routing policies is to accept and transmit the data traffic based on the VPN routing table, with no SLA-based considerations.

However, Cisco SD-WAN allows us to define a "default" sla-class at the end of an Application-aware Routing Policy as shown in the output above. This feature is useful in cases where we want to implement some baseline SLA requirements. For example, let's say that we do not want to forward any data traffic through a tunnel that experiences latency higher than 300ms. We define a sla-class named DEFAULT-SLA and apply it under the default action for the app-route policy. Then when the data traffic that is not matched at any policy rule hits the default action, the router will first try to forward the traffic through the tunnels that have latency lower than 300ms (tunnels that meet the SLA). If no overlay tunnel meets the SLA class configured in the default action, the WAN edge router will forward the data traffic through any of the available tunnels load-balancing across any equal paths.

Notice that we cannot configure the strict option under the default-action.

Summary of all App-route Actions

The following flow chart illustrates all App-route actions that we have seen so far:

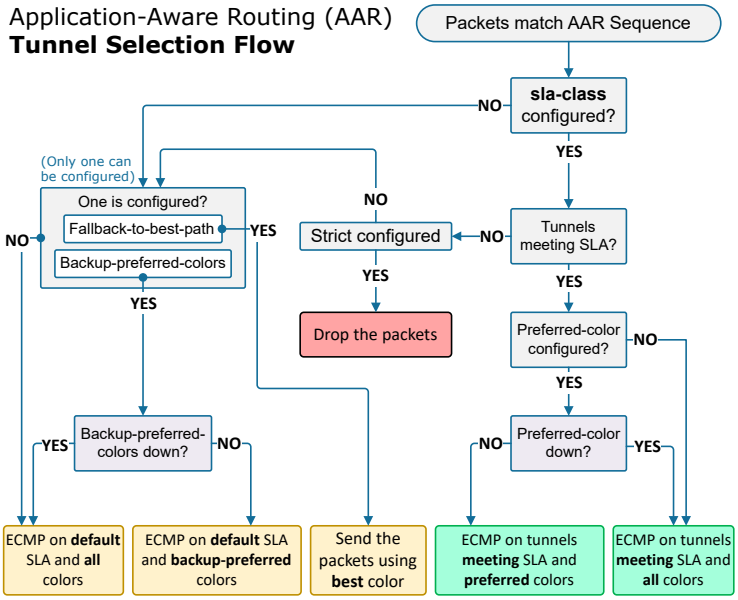


Figure 9. Application-Aware Routing Tunnel Selection Flow Chart

Notice that each app-route action leads to the data traffic being forwarded. The only exception is the strict keyword which explicitly drops the traffic if no overlay tunnel meets the SLA requirements.

Applying the AAR Policy

Application-aware routing policies are applied without specifying a direction as shown in the output below. That's because they always affect the data traffic that comes to a WAN edge router from the service side.

Notice that application-aware routing policies accept non-matching traffic by default. That is why they are considered to be positive policies while all other SD-WAN policies are considered negative ones.

11. CLOUD ONRAMP

Cloud onRamp for SaaS

The Business Need

Before we begin, let's first define what is a Software-as-a-Service (SaaS) application from a network perspective? Well, a SaaS application is just an enterprise-grade business application that is typically used over the Internet. The most popular SaaS applications at the moment are Microsoft Office 365, Google Workplace, and Salesforce. From a network standpoint, there is not much difference between a SaaS application and a regular Internet website. However, there is a huge difference from a business perspective! That is why SaaS applications are special and need to be treated differently because the business relies on these software-as-a-service apps, and when they are not working, the business is impacted.

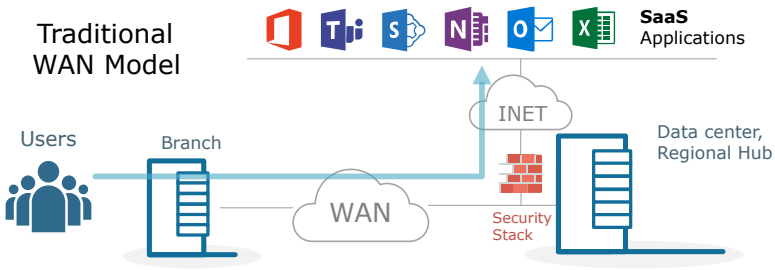


Figure 1. The Traditional WAN model and SaaS

So the business needs to make sure that these SaaS applications are available 24/7. But how do we achieve that with the traditional WAN approach of backhauling traffic over private WAN circuits to the data center via a hub-and-spoke architecture? This model is expensive, introduces unnecessary latency, creates a bandwidth bottleneck at the datacenter WAN links, and a performance bottleneck at the centralized security stack. This imposes major problems related to single-point-of-failures (the data center, the security stack), complexity, and user experience. The visibility into the application performance characteristics between the end-user and the Internet is also very limited.

Another aspect of the problem is the traditional network protocol stack. Even if we allow direct Internet access at branches, there is a very limited set of traditional protocols that can track the performance of a service on the Internet and route around a failure or performance degradation. But at the same time, Internet links do not have guaranteed quality, they can degrade at any given moment. Also, Internet Service providers (ISP) can suffer from outages, congestions, ongoing DDoS attacks, prefix black holes, and other conditions that can affect the reachability of Internet services. Security is another major factor that makes the tasks even harder. Opening the branches directly to Internet can expose them to cybersecurity vulnerabilities, would allow users to access unauthorized storage locations on the Internet, and would expose the company to sensitive data infiltration.

What is Cloud onRamp for SaaS?

Cloud onRamp for SaaS (formerly known as CloudExpress) is a set of Cisco SD-WAN capabilities designed to address the challenges that Software-as-a-Service applications impose on the wide-area network. In a nutshell, Cloud onRamp for SaaS lets us specify SaaS applications and DIA(Direct Internet access) interfaces and then allows only these pre-defined apps to break out to the Internet through the specified local DIA circuits while at the same time determines the best performing path for each SaaS app. Additionally, the solution continuously monitors each available path for each SaaS application, and in case of a problem with one path, it dynamically moves the SaaS traffic to an alternative one.

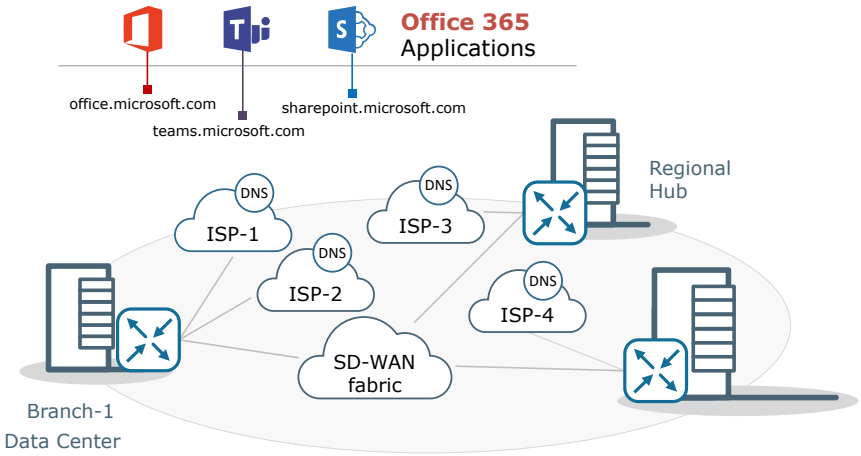


Figure 2. Cloud onRamp for SaaS Topology

There are a few common architectures that we are going to go through in this lesson using the topology shown in figure 2:

- Scenario 1: Accessing SaaS applications through DIA Links at branches
- Scenario 2: Using the DIA link of a Gateway Site for redundancy
- Scenario 3: Direct Internet Access through Colos or CNFs
- Scenario 4: Direct Internet Access through Secure Web Gateways (SWGs)

Scenario 1: Accessing SaaS applications through DIA Links at branches

Organizations that use multiple inexpensive Internet links at remote branches can enable Cloud onRamp on the WAN edge router to permit traffic from selected SaaS applications to break out directly to the Internet. It is important to note that, only traffic from these SaaS applications will be allowed to use the local Intertnet links, while all other user flows will follow the regular overlay routing paths.

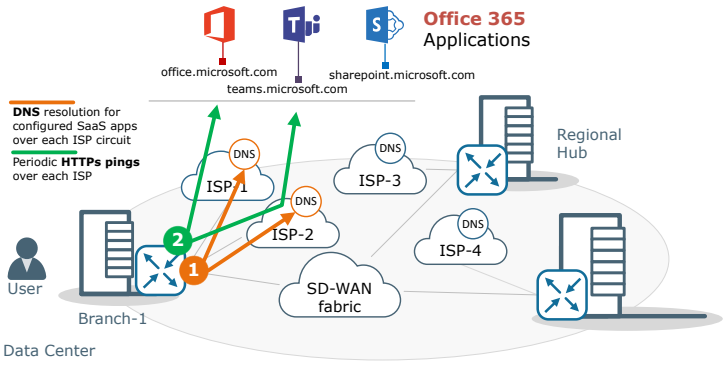


Figure 3. Cloud onRamp for SaaS Quality Probing

When we specify a Software-as-a-Service App for Branch-1, the vEdge on-site performs the following steps visualized in figure 3:

1. The WAN edge router at Branch-1 performs DNS resolution for the configured SaaS applications separately over each ISP circuit. This implies that there must be a DNS server address in VPN 0 for each different Internet Service Provider. Note that most popular SaaS apps have their own worldwide networks and will resolve with different IP addresses in different regions/sub-regions of the world.
2. The WAN edge router at Branch-1 initiates periodic HTTPs pings to each configured Software-as-a-Service app. A Quality of Experience score (1-10) is then calculated based on packet loss and latency values reported by the HTTPs pings. This is done separately over each ISP circuit. The vEdge router then selects the path with the highest score as the best-performing path.

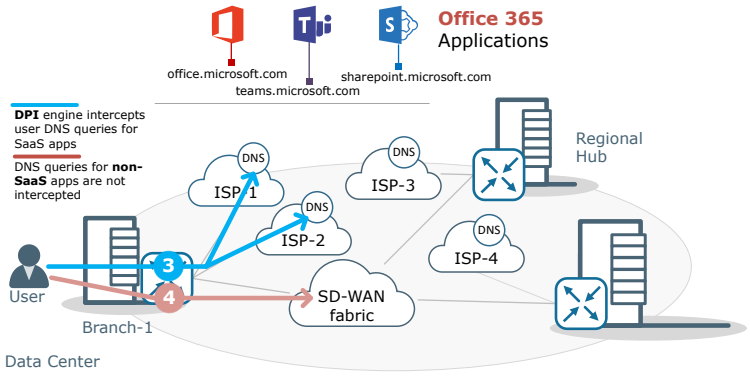


Figure 4. Cloud onRamp for SaaS Host DNS resolution

When a user onsite connects for the first time to one of the Software-as-a-Service apps, the user's device initiates a DNS query for the apps' URL, and the following chain of events visualized in figure 4 happen:

3. The WAN edge router's DPI engine intercepts the user's DNS query. If the host DNS query is for the Cloud onRamp SaaS application, the vEdge router forwards it over the best performing circuit to the DNS server that is defined for this ISP.
4. DNS queries for non-Cloud onRamp applications are forwarded according to the routing table towards the SD-WAN overlay fabric

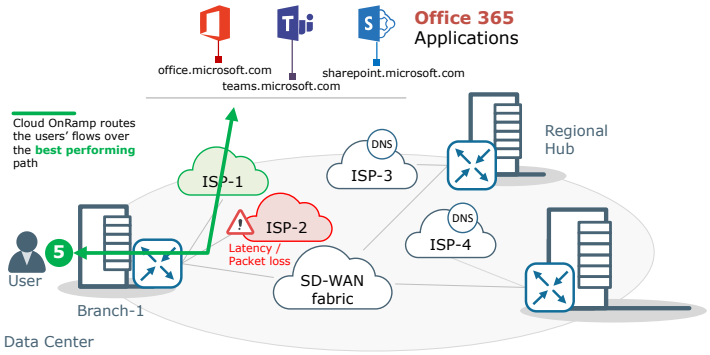


Figure 5. Cloud onRamp for SaaS Traffic Steering

5. When a user initiates a connection to the app, the WAN edge's DPI engine identifies that this flow is part of a Cloud onRamp for SaaS application and overrides the routing decision for it. The flow is rerouted through the best-performing Internet circuit that is configured for this Software-as-a-Service service.

It is very important to note that all other users' flows that are not part of the Cloud onRamp will be routed using the traditional Cisco SD-WAN routing over the overlay fabric.

Scenario 2: Using the DIA link of a Gateway Site

In many production deployments, remote sites only have one Internet link which can be used for Direct Internet Access (DIA). In this scenario, the branch site can use a gateway site that has DIA links to the Internet for redundancy in case of its own Internet link degrades.

The Cisco SD-WAN can select the best connection for each application through the gateway site. If the remote site connects use more than one gateway site, SD-WAN ensures that SaaS traffic uses the optimal path for each app, even through different gateway sites.

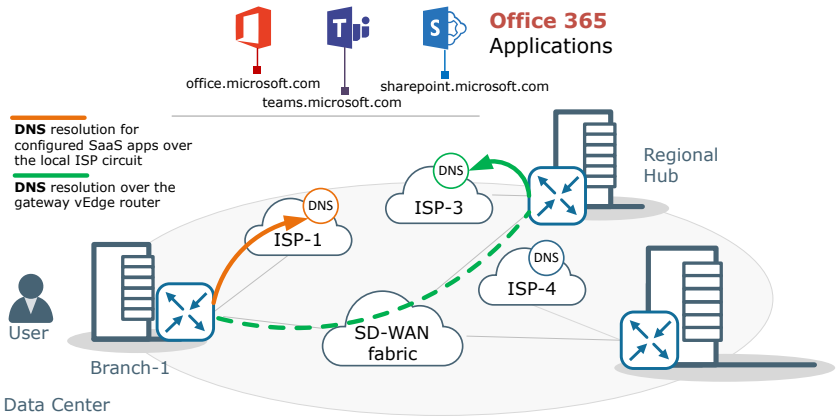


Figure 6. DNS Resolution via a Gateway Site

The process is similar to the previous scenario with the only difference being the way the QoE score of the path via the gateway site is calculated.

1. The WAN edge router at Branch-1 and at the gateway site (the Regional Hub) perform DNS resolution for the configured SaaS application as is visualized in figure 6.

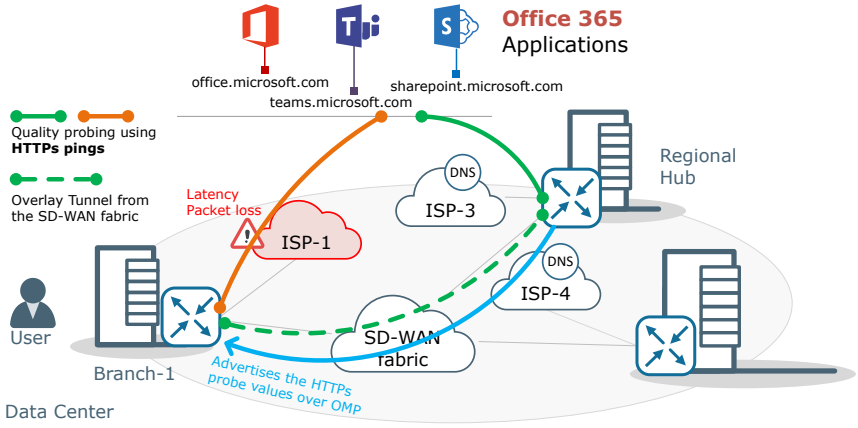


Figure 7. Quality Probing via a Gateway Site

2. Both vEdge routers then initiate periodic HTTP probes toward the configured app.
3. The vEdge router at Branch-1 determines the best performing path toward the SaaS application based on the QoE score.
   - Compares between the local Internet link's QoE score vs the composite metric of HTTP pings from the gateway vEdge (advertised via OMP, the blue line) + overlay tunnel health to gateway vEdge (the green dashed line)
   - The overlay tunnel health is determined using BFD

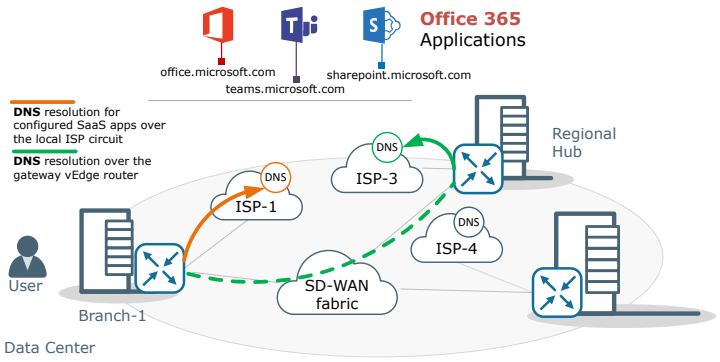


Figure 8. Traffic Steering via a Gateway Site

4. When a user onsite connects for the first time to one of the Software-as-a-Service apps, the user's device initiates a DNS query for the apps' URL, and the following chain of events happen:
5. The WAN edge router's DPI engine intercepts the DNS query
   - If the local DIA link is the best path, the router forwards the DNS query over the local Internet link
   - If the Gateway vEdge router (at the Regional Hub) is the best path, Branch-1's vEdge router forwards DNS query to the gateway vEdge router, which in turn forwards it to the DNS server defined locally over its local DIA circuit.
6. When a user initiates a connection to the app, the WAN edge's DPI engine identifies that this flow is part of a Cloud onRamp for SaaS application and overrides the routing decision for it. The flow is rerouted through the best-performing path.

Scenario 3: Direct Internet Access through Colos or CNFs

Some organizations do not want to allow direct Internet access at each and every remote branch and instead opt to use regional hubs to serve Internet traffic. These regional hubs can be hosted in 3rd party colocation facilities (Colos) or Carrier-Neutral Facilities (CNFs), and they can provide security capabilities with Next-Generation Firewall (NGFW) or Unified Threat Management (UTM).

In such deployments, Cloud onRamp can be deployed in a gateway mode, and it helps ensure that the optimal regional gateway is dynamically chosen for the traffic for each SaaS application.

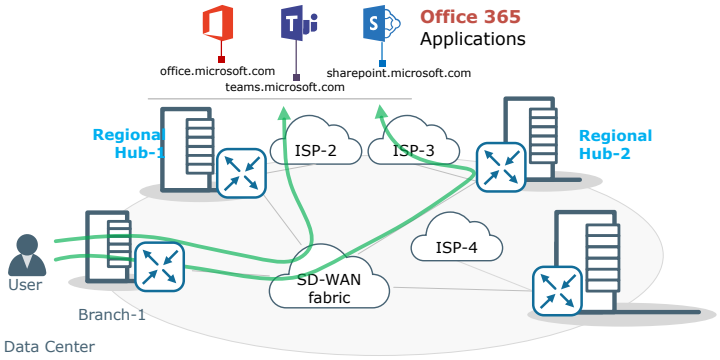


Figure 9. Direct Internet Access through Regional Hubs

Scenario 4: Direct Internet Access through Secure Web Gateways (SWGs)

In some deployments, enterprises connect remote branches to the SD-WAN fabric using inexpensive broadband Internet circuits, and they choose to enforce their IT security policies through a Secure Web Gateway (SWG) or Cloud Access Security Broker (CASB) point of presence. In such scenarios, Cloud onRamp for SaaS can be set up to dynamically choose the optimal path from among the multiple paths to the SWG (Figure 10).

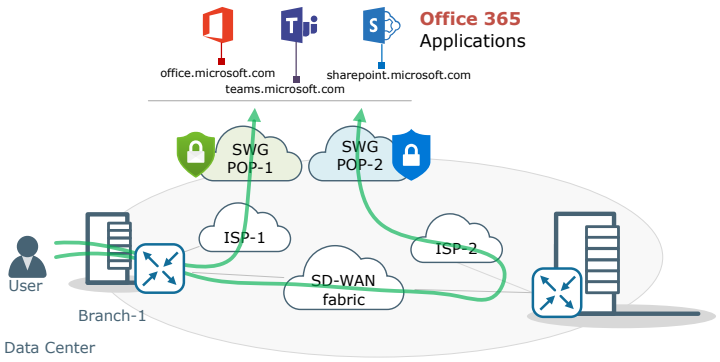


Figure 10. Direct Internet Access through Secure Web Gateways

Cloud onRamp for IaaS

Before we begin, let's first define what is Infrastructure as a Service (IaaS). IaaS is a virtualized computing infrastructure, provisioned and managed over the Internet that can be used to host and deliver enterprise applications.

The Business Need

Infrastructure as a Service (IaaS) has many benefits over the on-premise data center infrastructure. To list a few:

- It can scale up and down as the business demand changes;
- It drastically reduces the time to market;
- It decreases capital expenses and reduces ongoing costs;
- It has better security than on-prem;
- And many more.

The most popular IaaS providers are Amazon Web Service (AWS), Microsoft Azure, and Google Cloud (GCP). Extending the enterprise network to a public cloud provider can be a challenging task though. Each cloud provider has different connectivity and provisioning models.

What is Cloud onRamp for IaaS?

Cloud onRamp for IaaS is a set of capabilities that extend the Cisco SD-WAN overlay fabric to a public cloud instance. This allows remote branches, campuses, and data centers within the SD-WAN overlay fabric to leverage features such as Application-Aware Routing (AAR) to choose the best path to reach the applications hosted in private VPCs within a public cloud provider such as AWS, Azure, or GCP. Cloud onRamp for IaaS is designed to automate the connectivity to IaaS workloads and most importantly to provide visibility into the cloud.

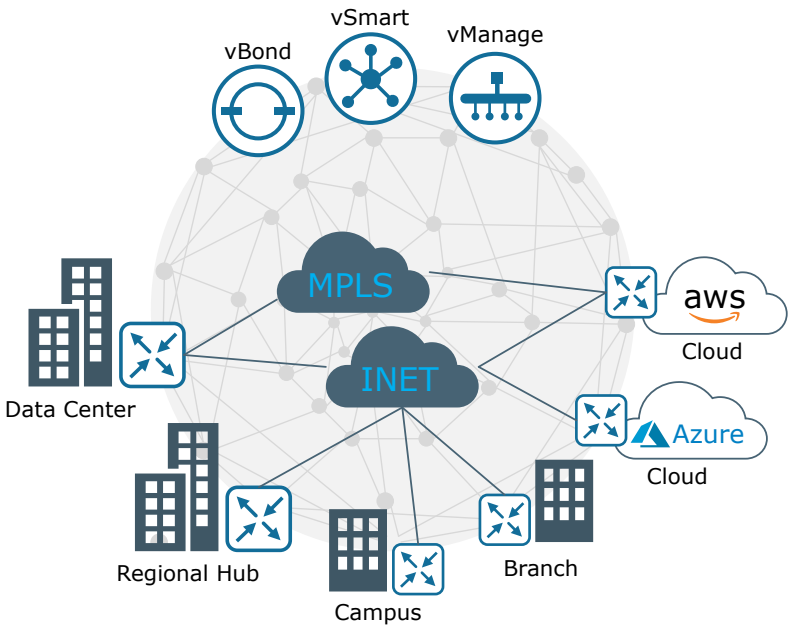


Figure 1. Cloud onRamp for IaaS

The connection between the SD-WAN overlay fabric and a public-cloud application is provided by a pair of redundant virtual WAN edge routers as is visualized in figure 2. These virtual SD-WAN devices (vEdge Cloud or Cisco CSR1000V) act as a transit between the overlay fabric and the applications hosted in the cloud. They provide device level and path resiliency to the connectivity to the public cloud.

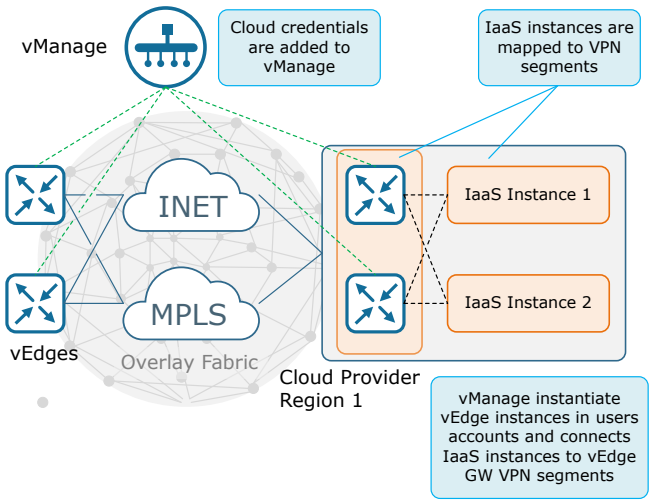


Figure 2. How does Cloud onRamp for IaaS works

A high-level overview of the steps required to deploy a Cloud onRamp for IaaS are:

Identify a pair of unused vEdge Cloud routers in Cisco vManage.

Attach a basic device template to both devices.

Enter the cloud provider's credentials (AWS or Azure API access and secret keys).

Add the transit Virtual Private Cloud (VPC) or transit Virtual Network (VNet) configuration depending on the cloud provider.

Discover and map host VPCs or host VNets to the transit VPC/VNET.

The infrastructure on AWS and Microsoft Azure can be seamlessly integrated into the overlay fabric. Once integrated into the fabric, the cloud vEdges can be managed using the same templates, security, and other Cisco SD-WAN policies which are used on-premises and on other clouds.

Cloud Terminology

Before we see how Cloud OnRamp for IaaS is implemented in AWS and Azure, let's introduce some cloud terminology that will come in handy later on.

Description	AWS	Azure	GCP
Customer Hierarchy	Organization > Accounts > Users > API/Secret Keys	Tenant > Subscriptions > Users > Client IDs/Secrets	
Geography	Regions > Availability Zones > VPCs	Locations > Availability Sets > VNETs	Regions > Zones >
VMs	EC2 instances	Azure Virtual Machines	Compute Engine
Virtual Networks Private	AWS Virtual Private Cloud (VPC)	Virtual Network (VNET)	Google Virtual Private Cloud (VPC)
Dedicated Connection	AWS Direct Connect	Azure ExpressRoute	Google Cloud Interconnect
Internet Gateway	IGW	Internet Gateway	
IPsec VPN Gateway	VGW	Azure VPN Gateway	Cloud VPN
Security	Security Groups / ACLs	Network Security Groups (NSG)	Compute Engine Firewall Rules

Public Cloud Terminology

Cloud onRamp for IaaS with AWS

Amazon Web Services (AWS) is the biggest Infrastructure-as-a-Service provider as of present. A virtual on-demand network on AWS is called a virtual private cloud (VPC). A VPC is logically isolated from other virtual networks (VPC) within the AWS infrastructure. The cloud provider allows traffic to flow between different VPCs within a region or between regions through VPC peering connections. However, AWS does not allow traffic to transit through a host VPC. This means that traffic must either originate or terminate within a virtual provide cloud (VPC) but not pass through it. If we consider this at scale, as the number of VPC instances increases, the amount of VPC peerings between the instances would increases dramatically if full-mesh connectivity between VPCs is a requirement.

Cloud onRamp for IaaS is designed to solve these scaling issues by implementing a networking construct called a Transit VPC. A Transit VPC can connect multiple VPCs that might be geographically disparate or running in separate AWS accounts.

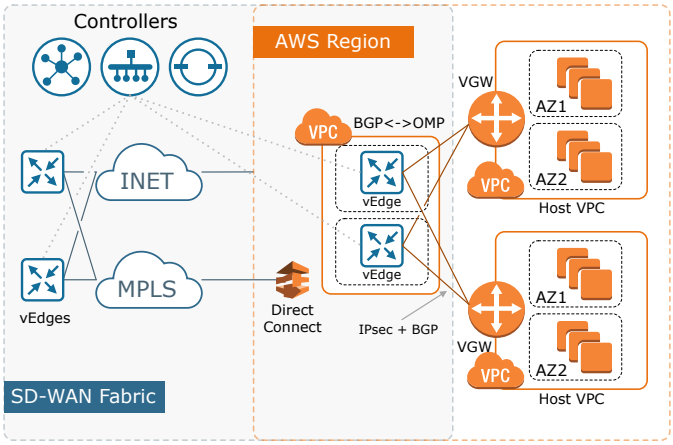


Figure 3. Cloud onRamp for IaaS with AWS

Within the VPC dedicated to function as a transit point, a pair of Cisco WAN Edge routers are deployed to route the traffic between the host VPCs. Each vEdge is instantiated within a different availability region within the transit VPC for resilience in case of failure and is automatically provisioned with the following:

- A transport VPN 0, also available via an AWS Elastic IP address
- A management VPN 512, available via an AWS Elastic IP address (public IP address)
- One or more service VPNs which have a range from 0 - 65528

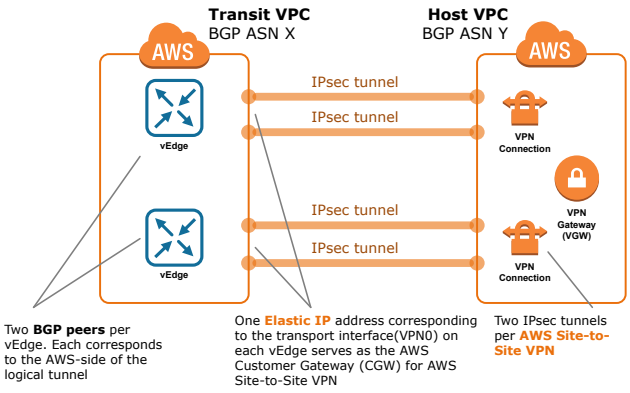


Figure 4. VPN connections to Host VPCs

The transit VPC also provides the entry point from AWS into the Cisco SD-WAN overlay fabric. The AWS VPN gateway at each host VPC establishes redundant site-to-site VPN connections to each vEdge cloud router within the transit VPC, through the service VPN side of the Cisco vEdge cloud routers.

When a host VPC is mapped to the transit VPC, Cisco Cloud onRamp automatically creates a redundant pair of AWS site-to-site VPN connections at the host VPC using the AWS APIs. Each AWS IPsec VPN connection is then mapped to one of the two Cisco vEdge routers within the transit VPC. From an AWS perspective, each Cisco WAN Edge Cloud router within the transit VPC functions as a customer gateway. Each AWS site-to-site VPN connection consists of a pair of IPsec tunnels established to the same customer gateway. Therefore, a total of two IPsec tunnels is established from each host VPC to the transit VPC as is shown in figure 4.

Cloud onRamp for IaaS with Azure

At a high level, the process of deploying Cloud onRamp for IaaS with Azure is pretty similar to the process with AWS.

The entire solution is completely automated – the end-user simply needs to enter his Azure subscription ID, along with the tenant ID, application ID and secret key in the related vManage section, discover Azure hosted virtual networks and workloads, and define two routers for interconnection. Cisco Cloud onRamp for IaaS brings up a fully deployed Azure hosted transit VNet containing a pair of Cisco WAN Edge routers, extends the fabric of the Cisco SD-WAN overlay network into the public cloud via the transit VNet, and allows Cisco SD-WAN branches to connect directly to public-cloud application providers.

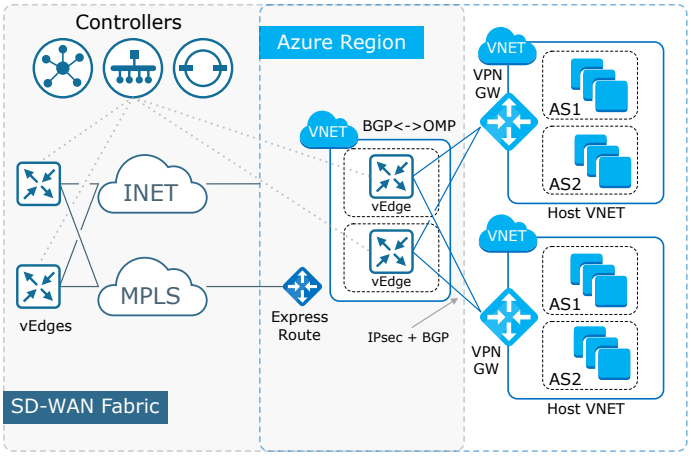


Figure 5. Cloud onRamp for IaaS with Azure

Using Cisco SD-WAN Cloud onRamp for IaaS, you can automatically spin up virtual WAN Edge router instances via Cisco vManage in a specific region of the public cloud. These virtual instances become part of the SD-WAN overlay and establish data plane connectivity to the WAN Edge routers located in the branch and/or the datacenter. As a result, secure end-to-end connectivity is established between the workloads in the cloud, physical branches, and data centers.

Cloud onRamp with AWS - Design Options

In this lesson, we are going to go through a high-level overview of the most common Cisco SD-WAN Cloud onRamp designs when it comes to integration with Amazon Web Services (AWS). The focus of each design is how to deploy secure network connectivity from remote locations within the SD-WAN fabric to one or more Amazon Web Services (AWS) virtual private clouds (VPCs) using the Cisco SD-WAN Cloud onRamp for Infrastructure as a Service (IaaS) feature.

A VPC is an on-demand virtual network, logically isolated from other virtual networks within an IaaS public cloud.

Design Option 1 - Using a Transit VPC

This is the most common design option that Cisco SD-WAN support. A pair of WAN edge routers are deployed within a transit VPC that has the single purpose of transporting traffic between other Host-VPCs. These host VPCs are connected to the transit VPC using AWS Site-to-Site VPN connections as is visualized in figure 1 below:

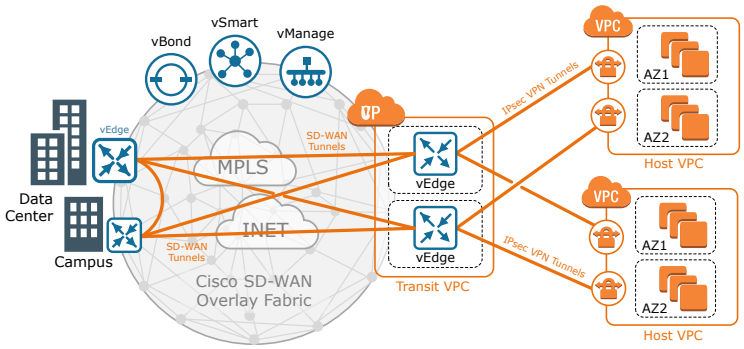


Figure 1. Cloud onRamp with AWS using a Transit VPC

As with all network designs, this one has its own advantages and disadvantages.

Pros:

- **Extended fabric -** The main benefit of this design is that it extends the Cisco SD-WAN overlay fabric into the AWS cloud. This allows all remote locations that are part of the SD-WAN fabric to use advanced features such as Application-Aware Routing (AAR) to choose the best path to reach application hosted in the cloud provider. This is visualized in figure 2 where a client in a branch site communicates with an application hosted in a private VPC in AWS. The Cisco SD-WAN overlay fabric chooses ISP-1 over ISP-2 as the best performing transport to AWS based on the AppQoE score calculated by AAR.
- **Fully Automated -** This design is fully automated and managed through the vManage GUI.

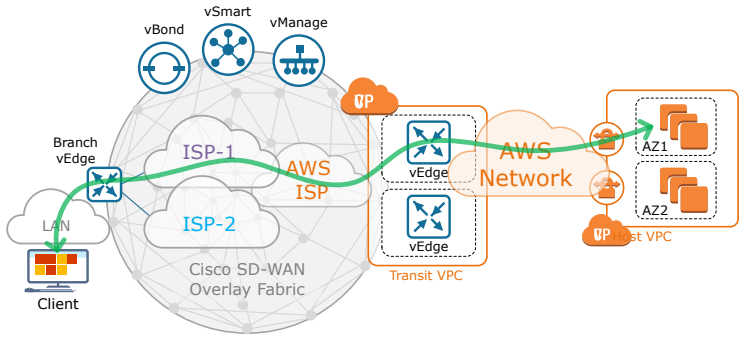


Figure 2. Application-aware Routing chooses the best path to AWS

Cons:

- **Scale -** The major drawback of this design option is that each host VPC has to establish an AWS Site-to-Site VPN tunnel to each vEdge in the transit VPC. Therefore, each host VPC that the organization has in AWS would add four additional IPsec tunnels since each AWS site-to-site VPN consists of two IPsec tunnels and there are two vEdges in the transit VPC. Hence, as the number of Host VPCs increases, the number of IPsec tunnels increases x4 and at some point, the limit of supported IPsec tunnels would be reached at the WAN edge devices.

Design Option 2 - Using a Transit VPC as a Cloud Gateway (CGW)

This design option again extends the overlay fabric into the cloud through a transit VPC. However, host VPCs are now connected to an AWS Transit Gateway (TGW) instead of directly to the Transit VPC.

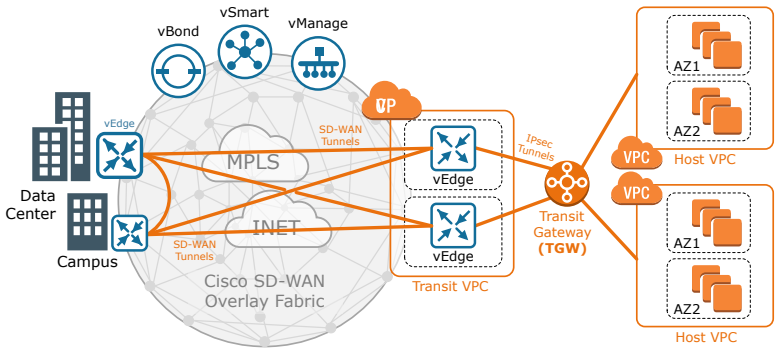


Figure 3. Cloud onRamp with AWS using a Cloud Gateway

The AWS TGW connects to the transit VPC through VPN attachments. This design is referred in the Cisco documentations as the SD-WAN Cloud Gateway (CGW).

Pros:

- **Extended fabric -** This design option also extends the Cisco SD-WAN overlay fabric into the AWS cloud. This allows all remote locations that are part of the SD-WAN fabric to use advanced features such as Application-Aware Routing (AAR) to choose the best path to reach application hosted in the cloud provider.
- **Scale -** This design option is more scalable than the above one because each host VPC is connected to the Transit Gateway (TGW) via a VPC attachment instead of a VPN attachment (AWS site-to-site VPN per VPC).

Cons:

- **Not fully automated yet -** Implementing Cloud onRmp Cloud Gateway design is not fully automated in the present vManage version 20.3.1. Onboarding existing AWS Transit Gateway with connected host VPCs requires manual setup and extensive AWS knowledge.

Design Option 3 - Using a Transit VPC as a Cloud Gateway with TGW Connect

Cisco has recently announced that the new version of vManage will support automation of the entire orchestration of the Transit Gateway deployment and VPC networking, hence reducing the manual operations that I have classified as a downside of the above design option.

The new design will support the AWS TGW Connect attachments. These connect attachments are in essence GRE tunnels running BGP on top. They support 4 times higher bandwidth than the regular IPsec tunnels and eliminate the costs of maintaining many IPsec tunnels.

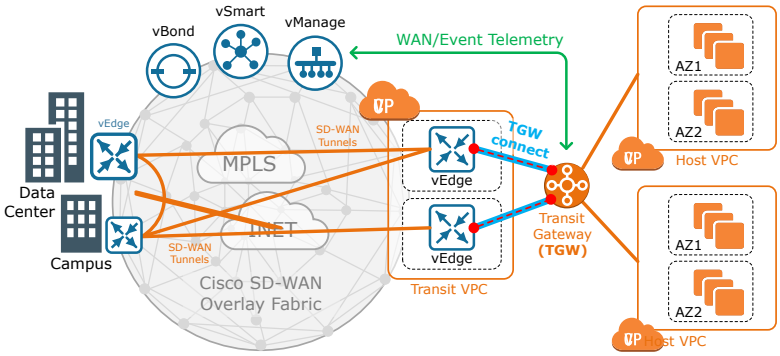


Figure 4. Cloud onRamp with AWS using a Cloud Gateway with TGW Connect

Pros:

- Higher bandwidth with GRE tunnels - the GRE tunnels offer 4 times more bandwidth than the IPsec tunnels
- Private IP addresses - Removing the IPsec tunnels removes the need for public IP addresses. Organizations can now deploy this design using private IP space which greatly reduces the cyberattack surface.
- Increased route limit - Currently the BGP route limit is 100. The new architecture promises to increase to significantly increase this limit.
- Increased visibility into the cloud - Integration with Transit Gateway Network Manager will allow for telemetry data exchange between vManage and TGW, which will subsequently increase the level of visibility within the cloud.

Cons:

- Still not available as of present-day - vManage version 20.3.1

Design Option 4 - Using a Transit Gateway (TGW)

Another design option is to use a Transit Gateway (TGW) that aggregates all host VPCs, and then also connect the vEdges at remote locations directly to the AWS Transit Gateway as is visualized in figure 5.

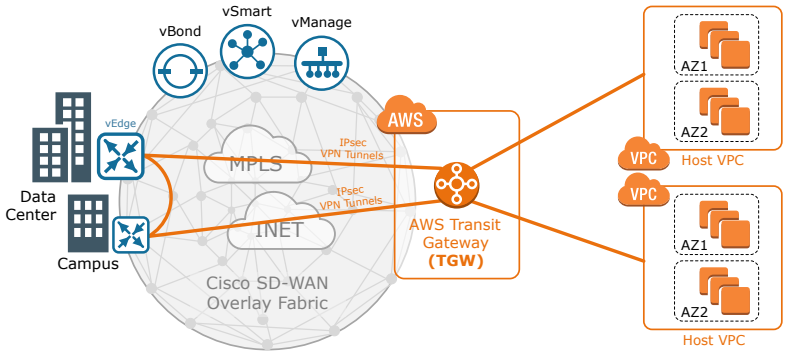


Figure 5. Cloud onRamp with AWS using a Transit Gateway

Pros:

- Cost - This design does not need to use a transit VPC. Therefore, all expenses incurred running AWS EC2 instances to support the vEdge routers would not apply.
- Bandwidth - Since host VPCs are connected with a VPC connection through the AWS TWG, there is typically more bandwidth available than using Virtual Private Gateways (VGWs) and Site-to-Site VPN connections

Cons:

- No fabric in the cloud - This design option does not extend the Cisco SD-WAN overlay fabric into the AWS cloud. Therefore, all benefits of using Application-Aware Routing (AAR) are not available.
- No automation - Static configuration of public IP addresses is required for each AWS Site-to-Site VPN connection at the AWS TGW.
- Cost - Charges for each AWS Site-to-Site VPN Connection from vEdge routers at remote locations to the TGW still apply. Depending on the number of remote locations that need to have a tunnel to AWS, the overall cost may get higher than the other design options.

NETWORKERS HOME
Ignite Your Genius

Head Office:

L-149, 1st, 2nd and 3rd Floor, Eshwari Mansion, 5th Main Road, Sector-6 HSR Layout,
Bengaluru, Karnataka 560102, India
Mobile No: **+91-9611027980** | **+91-9354284954**, Email: **info@networkershome.com**